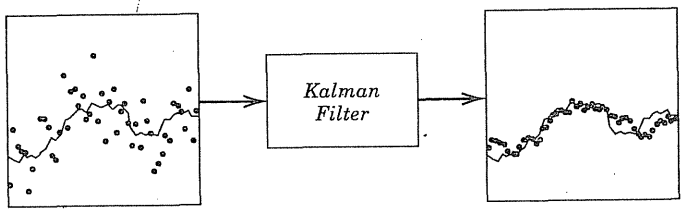


(See also p. 224 for more info on this example.)

". . . the Kalman filter represents the most widely applied and demonstrably useful result to emerge from the state variable approach of *modern control theory.*"

Harold W. Sorenson
Kalman Filtering: Theory and Application,
IEEE Press, 1985

THIRD EDITION

Introduction to Random Signals and Applied Kalman Filtering

WITH MATLAB EXERCISES AND SOLUTIONS

Robert Grover Brown
Electrical Engineering Department
Iowa State University

Patrick Y. C. Hwang
Rockwell International Corporation

JOHN WILEY & SONS
New York • Chichester • Brisbane • Toronto • Singapore • Weinheim

ACQUISITIONS EDITOR Charity Robey
MARKETING MANAGER Jay Kirsch
PRODUCTION EDITOR Ken Santor
MANUFACTURING MANAGER Mark Cirillo
ILLUSTRATION COORDINATOR Gene Aiello
COVER DESIGNER Carol C. Grobe

This book was set in 10/12 Times Roman by Pro-Image Corporation, and printed and bound by Hamilton Printing Company. The cover was printed by Lehigh Press, Inc.

Recognizing the importance of preserving what has been written, it is a policy of John Wiley & Sons, Inc. to have books of enduring value published in the United States printed on acid-free paper, and we exert our best efforts to that end.

The paper on this book was manufactured by a mill whose forest management programs include sustained yield harvesting of its timberlands. Sustained yield harvesting principles ensure that the number of trees cut each year does not exceed the amount of new growth.

Library of Congress Cataloging-in-Publication Data

Brown, Robert Grover.
Introduction to random signals and applied Kalman filtering : with MATLAB exercises and solutions / Robert Grover Brown, Patrick Y.C. Hwang. — 3rd ed.
p. cm.
Includes index.
ISBN 0-471-12839-2 (cloth : alk. paper)
1. Signal processing—Data processing. 2. Random noise theory. 3. Kalman filtering—Data processing. 4. MATLAB. I. Hwang, Patrick Y. C. II. Title.
TK5102.9.B75 1997
621.382'2—dc20 96-25880
CIP

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

Preface to the Third Edition

This text is a third edition of *Introduction to Random Signals and Applied Kalman Filtering*. At the time we prepared the second edition, there was no comprehensive PC mathematics software available to engineering students at an affordable price. All this changed with the publication of *The Student Edition of MATLAB®* (Prentice-Hall, 1992).* This was followed in 1995 with an upgraded version under the same title (but referred to as Version 4). This has made it feasible for an instructor to expect every student in class to have available, as a minimum, the student edition of MATLAB as an aid in solving engineering problems. This is especially true in upper-level courses where the student would very likely have gained some familiarity with MATLAB from earlier courses. We are not providing the MATLAB software package with this text. It is assumed that the student already has MATLAB available (or perhaps some other suitable mathematics software) and knows how to use it.

The problems at the end of each chapter that are flagged with a small computer icon are "computer" exercises. They cannot be worked out by paper-and-pencil methods with a reasonable amount of effort. They were written with MATLAB in mind, but they can, of course, be worked out with other suitable software. We encourage all those who are interested in learning about Kalman filtering to complete a generous number of these exercises. The discrete Kalman filter is a numerical procedure, so considerable insight into the filter's behavior is gained in doing so. Just *looking* at the equations is not enough for most of us. Programming the equations and analyzing the results of specific examples is the best way to get the insight that is so essential in engineering work.

MATLAB is an especially good software package for working out Kalman filtering problems because of its convenience in handling matrix operations. A diskette containing MATLAB M-files giving solutions for most of the computer problems and examples is included inside the back cover of the text. The solution

* MATLAB is a registered trademark of The Math Works, Inc., 24 Prime Park Way, Natick, MA 01760-1500, Ph: (508) 647-7000

M-files were written for tutorial purposes, and many comment statements are interspersed with the executable ones. They were written to be understandable, not for efficiency. (See Appendix C for more on the software.)

The previous edition of this book included a Kalman filtering software package called "Kalm 'N Smooth." This menu-oriented software is not as versatile as MATLAB, but it is still good effective software for solving certain types of Kalman filtering problems. Thus, it is also included in the diskette for those who wish to use it.

The main thrust of this text is applied Kalman filtering. Our intent, just as in the earlier editions, has been to present the subject matter at an introductory level. We have found from teaching both university-credit and continuing-education courses that the main impediment to learning about Kalman filtering is not the mathematics. Rather, it is the background material in random process theory and linear systems analysis that usually causes the difficulty. Chapters 1 through 3 are intended to provide a minimal background in random process theory and the response of the linear systems to random inputs. Knowledge of this material is essential to the subject matter in the remaining chapters. The necessary prerequisite material on linear systems analysis can be found in most any junior- or senior-level engineering text on linear systems analysis or linear control systems. Chapter 4 is on Wiener filtering. Those who are primarily interested in Kalman filtering may wish to skip this chapter, except for Section 4.7 on the discrete Wiener filter. This has relevance to the discrete Kalman filter. Chapters 5 through 11 deal with various facets of Kalman filtering with emphasis on applications throughout.

The authors wish to express special thanks to Dr. Larry Levy for his many helpful comments during the course of preparation of this third edition. We also wish to thank Lova Brown (Mrs. Robert Grover Brown) for her patience and help in preparing the final manuscript.

Robert Grover Brown
Patrick Y. C. Hwang

Contents

1

Probability and Random Variables: A Review

1.1 RANDOM SIGNALS

Nearly everyone has some notion of random or noiselike signals. One has only to tune an ordinary AM radio away from a station, turn up the volume, and the result is static, or noise. If one were to look at a strip-chart recording of such a signal, it would appear to wander on aimlessly with no apparent order in its amplitude pattern, as shown in Fig. 1.1. Signals of this type cannot be described with explicit mathematical functions such as sine waves, step functions, and the like. Their description must be put in probabilistic terms. Early investigators recognized that random signals could be described loosely in terms of their spectral content, but a rigorous mathematical description of such signals was not formulated until the 1940s, most notably with the work of Wiener and Rice (1, 2).

Noise is usually unwanted. The additive noise in the radio signal disturbs our enjoyment of the music or interferes with our understanding of the spoken word; noise in an electronic navigation system induces position errors that can be disastrous in critical situations; noise in a digital data transmission system can cause bit errors with obvious undesirable consequences; and on and on. Any noise that corrupts the desired signal is bad; it is just a question of how bad! Even after designers have done their best to eliminate all the obvious noise-producing mechanisms, there always seems to be some noise left over that must be suppressed with more subtle means, such as filtering. To do so effectively, one must understand noise in quantitative terms.

Probability plays a key role in the description of noiselike signals. Our treatment of this subject must necessarily be brief and directed toward the specific needs of subsequent chapters. The scope is thus limited in this regard. We make no apology for this, because many fine books have been written on probability in the broader sense. Our main objective here is the study of random signals and optimal filtering, and we wish to move on to this area as quickly as possible. First, though, we must at least review the bare essentials of probability with special emphasis on random variables.

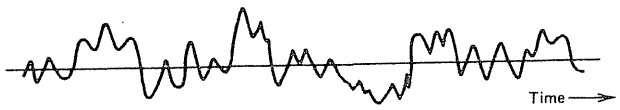


Figure 1.1 Typical noise waveform.

1.2 INTUITIVE NOTION OF PROBABILITY

Most engineering and science students have had some acquaintance with the intuitive concepts of probability. Typically, with the intuitive approach we first consider all possible outcomes of a chance experiment as being equally likely, and then the probability of a particular event, say, event A, is defined as

$$P(A) = \frac{\text{Possible outcomes favoring event } A}{\text{Total possible outcomes}} \tag{1.2.1}$$

where we read $P(A)$ as "probability of event A." This concept is then expanded to include the relative-frequency-of-occurrence or statistical viewpoint of probability. With the relative-frequency concept, we imagine a large number of trials of some chance experiment and then define probability as the relative frequency of occurrence of the event in question. Considerations such as what is meant by "large" and the existence of limits are normally avoided in elementary treatments. This is for good reason. The idea of limit in a probabilistic sense is subtle.

Although the older intuitive notions of probability have limitations, they still play an important role in probability theory. The ratio-of-possible-events concept is a useful problem-solving tool in many instances. The relative-frequency concept is especially helpful in visualizing the statistical significance of the results of probability calculations. That is, it provides the necessary tie between the theory and the physical situation. Two examples that illustrate the usefulness of these intuitive notions of probability should now prove useful.

EXAMPLE 1.1

In straight poker, each player is dealt 5 cards face down from a deck of 52 playing cards. We pose two questions:

(a) What is the probability of being dealt four of a kind, that is, four aces, four kings, and so forth?

(b) What is the probability of being dealt a straight flush, that is, a continuous sequence of five cards in any suit?

Solution to Question (a) This problem is relatively complicated if you think in terms of the sequence of chance events that can take place when the cards are dealt one at a time. Yet the problem is relatively easy when viewed in terms of the ratio of favorable to total number of outcomes. These are easily counted in this case. There are only 48 possible hands containing 4 aces; another 48 containing 4 kings; etc. Thus, there are $13 \cdot 48$ possible four-of-a-kind hands. The total number of possible poker hands of any kind is obtained from the combination formula for "52 things taken 5 at a time" (3). This is given by the binomial coefficient

$$\binom{52}{5} = \frac{52!}{5!(52-5)!} = \frac{52 \cdot 51 \cdot 50 \cdot 49 \cdot 48}{5 \cdot 4 \cdot 3 \cdot 2 \cdot 1} = 2{,}598{,}960 \tag{1.2.2}$$

Therefore, the probability of being dealt four of a kind is

$$P(\text{four of a kind}) = \frac{13 \cdot 48}{2{,}598{,}960} = \frac{624}{2{,}598{,}960} \approx .00024 \tag{1.2.3}$$

Solution to Question (b) Again, the direct itemization of favorable events is the simplest approach. The possible sequences in each of four suits are: AKQJ10, KQJ109, . . . , 5432A. (*Note:* We allow the ace to be counted either high or low.) Thus, there are 10 possible straight flushes in each suit (including the royal flush of the suit) giving a total of 40 possible straight flushes. The probability of a straight flush is, then,

$$P(\text{Straight flush}) = \frac{40}{2{,}598{,}960} \approx .000015 \tag{1.2.4}$$

We note in passing that in poker a straight flush wins over four of a kind; and, rightly so, since it is the rarer of the two hands. ▩

EXAMPLE 1.2

Craps is a popular gambling game played in casinos throughout the world (4). The player rolls two dice and plays against the house (i.e., the casino). If the first roll is 7 or 11, the player wins immediately; if it is 2, 3, or 12, the player loses immediately. If the first roll results in 4, 5, 6, 8, 9, or 10, the player continues to roll until either the same number appears, which constitutes a win, or a 7 appears, which results in the player losing. What is the player's probability of winning when throwing the dice?

This example was chosen to illustrate the shortcoming of the direct count-the-outcomes approach. In this case, one cannot enumerate all the possible outcomes. For example, if the player's first roll is a 4, the play continues until another outcome is reached. Presumably, the rolling could continue on ad infinitum without a 4 or 7 appearing, which is what is required to terminate the game. Thus, the direct enumeration approach fails in this situation. On the other hand, the relative-frequency-occurrence approach works quite well. Table 1.1 shows the relative frequency of occurrence of the various numbers on the first roll. The numbers in the column labeled "probability" were obtained by enumerating the 36 possible outcomes and allotting $\frac{1}{36}$ for each outcome that yields a sum corresponding to the number in the first column. For example, a 4 may be obtained with the combinations (1, 3), (2, 2), or (1, 3). For the cases where the game continues after the first throw, the subsequent probabilities were ob-

Table 1.1 Probabilities in Craps

Number of First Throw	Probability	Result of First Throw	Subsequent Probabilities and Results	Relative Frequency of Winning with Various First Throws
2	$\frac{1}{36}$	Lose		0
3	$\frac{2}{36}$	Lose		0
4	$\frac{3}{36}$	Continue	$P(4 \text{ before } 7) = \frac{1}{3}$ (win) $P(7 \text{ before } 4) = \frac{2}{3}$ (lose)	$\frac{3}{36} \cdot \frac{1}{3}$
5	$\frac{4}{36}$	Continue	$P(5 \text{ before } 7) = \frac{2}{5}$ (win) $P(7 \text{ before } 5) = \frac{3}{5}$ (lose)	$\frac{4}{36} \cdot \frac{2}{5}$
6	$\frac{5}{36}$	Continue	$P(6 \text{ before } 7) = \frac{5}{11}$ (win) $P(7 \text{ before } 6) = \frac{6}{11}$ (lose)	$\frac{5}{36} \cdot \frac{5}{11}$
7	$\frac{6}{36}$	Win		$\frac{6}{36}$
8	$\frac{5}{36}$	Continue	$P(8 \text{ before } 7) = \frac{5}{11}$ (win) $P(7 \text{ before } 8) = \frac{6}{11}$ (lose)	$\frac{5}{36} \cdot \frac{5}{11}$
9	$\frac{4}{36}$	Continue	$P(9 \text{ before } 7) = \frac{2}{5}$ (win) $P(7 \text{ before } 9) = \frac{3}{5}$ (lose)	$\frac{4}{36} \cdot \frac{2}{5}$
10	$\frac{3}{36}$	Continue	$P(10 \text{ before } 7) = \frac{1}{3}$ (win) $P(7 \text{ before } 10) = \frac{2}{3}$ (lose)	$\frac{3}{36} \cdot \frac{1}{3}$
11	$\frac{2}{36}$	Win		$\frac{2}{36}$
12	$\frac{1}{36}$	Lose		0

Total probability of winning = $\frac{244}{495} \approx .4929$

tained simply by observing the *relative* frequency of occurrence of the numbers involved. For example, a 7 is twice as likely as a 4. Thus, the relative frequency of rolling a 7 before a 4 should be twice that of "4 before 7," and the respective probabilities are $\frac{2}{3}$ and $\frac{1}{3}$. The total probability of winning with a 4 on the first throw was reasoned as follows. A 4 appears on the first roll only $\frac{3}{36}$ of the time; and, of this fraction, only $\frac{1}{3}$ of the time will this result in an ultimate win. Thus, the relative frequency of winning via this route is the product of $\frac{3}{36} \cdot \frac{1}{3}$. Admittedly, this line of reasoning is quite intuitive, but that is the very nature of the relative-frequency-of-occurrence approach to probability.

For the benefit of those who like to gamble, it should be noted that craps is a very close game. The edge in favor of the house is only about $1\frac{1}{2}$ percent. (Also see Problem 1.7.) ■

1.3 AXIOMATIC PROBABILITY

It should be apparent that the intuitive concepts of probability have their limitations. The ratio-of-outcomes approach requires the equal-likelihood assumption for all outcomes. This may fit many situations, but often we wish to consider "unfair" chance situations as well as "fair" ones. Also, as demonstrated in Example 1.2, there are many problems for which all possible outcomes simply cannot be enumerated. The relative-frequency approach is intuitive by its very nature. Intuition should never be ignored; but, on the other hand, it can lead one astray in complex situations. For these reasons, the axiomatic formulation of probability theory is now almost universally favored among both applied and theoretical scholars in this area. As we would expect, axiomatic probability is compatible with the older, more heuristic probability theory.

Axiomatic probability begins with the concept of a *sample space.* We first imagine a conceptual chance experiment. The sample space is the set of all possible *outcomes* of this experiment. The individual outcomes are called *elements* or *points* in the sample space. We denote the sample space as S and its set of elements as $\{s_1, s_2, s_3, \ldots\}$. The number of points in the sample space may be finite, countably infinite, or simply infinite, depending on the experiment under consideration. A few examples of sample spaces should be helpful at this point.

EXAMPLE 1.3

The Experiment Make a single draw from a deck of 52 playing cards. Since there are 52 possible outcomes, the sample space contains 52 discrete points. If we wished, we could enumerate them as Ace of Clubs, King of Clubs, Queen of Clubs, and so forth. Note that the points of the sample space in this case are "things," not numbers. ■

EXAMPLE 1.4

The Experiment Two fair dice are thrown and the number of dots on the top of each is observed. There are 36 discrete outcomes that can be enumerated as (1, 1), (1, 2), (1, 3), . . . , (6, 5), (6, 6). The first number in parentheses identifies the number of dots on die 1 and the second is the number on die 2. Thus, 36 distinct 2-tuples describe the possible outcomes, and our sample space contains 36 points or elements. Note that the points in this sample space retain the identity of each individual die and the number of dots shown on its top face. ■

EXAMPLE 1.5

The Experiment Two fair dice are thrown and the sum of the number of dots is observed. In this experiment, we do not wish to retain the identity of the numbers on each die; only the sum is of interest. Therefore, it would be perfectly proper to say the possible outcomes of the experiment are {2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}. Thus, the sample space would contain 11 discrete elements. From this and the preceding example, it can be seen that we have some discretion in how we define the sample space corresponding to a certain experiment. It depends to some extent on what we wish to observe. If certain details of the experiment are not of interest, they often may be suppressed with some resultant simplification. However, once we agree on what items are to be grouped together and called *outcomes,* the sample space must include all the defined outcomes; and, similarly, the result of an experiment must always yield one of the defined outcomes. ■

EXAMPLE 1.6

The Experiment A dart is thrown at a target and the location of the hit is observed. In this experiment we imagine that the random mechanisms affecting the throw are such that we get a continuous spread of data centered around the bull's-eye when the experiment is repeated over and over. In this case, even if we bound the hit locations within a certain region determined by reasonableness, we still cannot enumerate all possible hit locations. Thus, we have an infinite number of points in our sample space in this example. Even though we cannot enumerate the points one by one, they are, of course, identifiable in terms of either rectangular or polar coordinates. ■

It should be noted that elements of a sample space must always be *mutually exclusive* or *disjoint.* On a given trial, the occurrence of one excludes the occurrence of another. There is no overlap of points in a sample space.

In axiomatic probability, the term event has special meaning and should not be used interchangeably with outcome. An *event* is a special subset of the sample space S. We usually wish to consider various events defined on a sample space, and they will be denoted with uppercase letters such as $A, B, C, \ldots$, or perhaps $A_1, A_2, \ldots$, etc. Also, we will have occasion to consider the set of operations of union, intersection, and complement of our defined events. Thus, we must be careful in our definition of events to make the set sufficiently complete such that these set operations also yield properly defined events. In discrete problems, this can always be done by defining the set of events under consideration to be all possible subsets of the sample space S. We will tacitly assume that the null set is a subset of every set, and that every set is a subset of itself.

One other comment about events is in order before proceeding to the basic axioms of probability. The event A is said to occur if *any* point in A occurs.

The three axioms of probability may now be stated. Let S be the sample space and A be any event defined on the sample space. The first two axioms are

$$\textit{Axiom 1:} \quad P(A) \geq 0 \tag{1.3.1}$$

$$\textit{Axiom 2:} \quad P(S) = 1 \tag{1.3.2}$$

Now, let $A_1, A_2, A_3, \ldots$ be mutually exclusive (disjoint) events defined on S. The sequence may be finite or countably infinite. The third axiom is then

$$\textit{Axiom 3:} \quad P(A_1 \cup A_2 \cup A_3 \cup \ldots) = P(A_1) + P(A_2) + P(A_3) + \cdots \tag{1.3.3}$$

Axiom 1 simply says that the probability of an event cannot be negative. This certainly conforms to the relative-frequency-of-occurrence concept of probability. Axiom 2 says that the event S, which includes all possible outcomes, must have a probability of unity. It is sometimes called the certain event. The first two axioms are obviously necessary if axiomatic probability is to be compatible with the older relative-frequency probability theory. The third axiom is not quite so obvious, perhaps, and it simply must be assumed. In words, it says that when we have nonoverlapping (disjoint) events, the probability of the union of these events is the sum of the probabilities of the individual events. If this were not so, one could easily think of counterexamples that would not be compatible with the relative-frequency concept. This would be most undesirable.

We now recapitulate. There are three essential ingredients in the formal approach to probability. First, a sample space must be defined that includes all possible outcomes of our conceptual experiment. We have some discretion in what we call outcomes, but caution is in order here. The outcomes must be disjoint and all-inclusive such that $P(S) = 1$. Second, we must carefully define a set of events on the sample space, and the set must be closed such that the operations of union, intersection, and complement also yield events in the set. Finally, we must assign probabilities to all events in accordance with the basic axioms of probability. In physical problems, this assignment is chosen to be compatible with what we feel to be reasonable in terms of relative frequency of occurrence of the events. If the sample space S contains a finite number of elements, the probability assignment is usually made directly on the elements of S. They are, of course, elementary events themselves. This, along with Axiom 3, then indirectly assigns a probability to all other events defined on the sample space. However, if the sample space consists of an infinite "smear" of points, the probability assignment must be made on events and not on points in the sample space. This will be illustrated later in Example 1.8.

Once we have specified the sample space, the set of events, and the probabilities associated with the events, we have what is known as a *probability space.* This provides the theoretical structure for the formal solution of a wide variety of probability problems.

EXAMPLE 1.7

Consider a single throw of two dice, and let us say we are only interested in the sum of the dots that appear on the top faces. This chance situation fits many games that are played with dice. In this case, we will define our sample space to be

$$S = \{2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$$

and it is seen to contain 11 discrete points. Next, we define the set of possible events to be all subsets of S, including the null set and S itself. Note that the elements of S are elementary events, and they are disjoint, as they should be. Also, $P(S) = 1$. Finally, we need to assign probabilities to the events. This could be done arbitrarily (within the constraints imposed by the axioms of probability), but in this case we want the results of our formal analysis to coincide with the relative-frequency approach. Therefore, we will assign probabilities to the elements in accordance with Table 1.2, which, in turn, indirectly specifies probabilities for all other events defined on S. We now have a properly defined probability space, and we can pose a variety of questions relative to the single throw of two dice.

Suppose we ask: What is the probability of throwing either a 7 or an 11? From Axiom 3, and noting that "7 or 11" is the equivalent of saying "7 $\cup$ 11," we have

$$P(7 \text{ or } 11) = \frac{6}{36} + \frac{2}{36} = \frac{2}{9} \tag{1.3.4}$$

Next, suppose we ask: What is the probability of not throwing 2, 3, or 12? This calls for the complement of event "2 or 3 or 12," which is the set {4, 5, 6, 7, 8, 9, 10, 11}. Recall that we say the event occurs if *any* element in the set occurs. Therefore, again using Axiom 3, we have

$$P(\text{Not throwing 2, 3, or 12}) = \frac{3 + 4 + 5 + 6 + 5 + 4 + 3 + 2}{36}$$

$$= \frac{8}{9} \tag{1.3.5}$$

Table 1.2 Probabilities for Two-Dice Example

Sum of Two Dice	Assigned Probability
2	$\frac{1}{36}$
3	$\frac{2}{36}$
4	$\frac{3}{36}$
5	$\frac{4}{36}$
6	$\frac{5}{36}$
7	$\frac{6}{36}$
8	$\frac{5}{36}$
9	$\frac{4}{36}$
10	$\frac{3}{36}$
11	$\frac{2}{36}$
12	$\frac{1}{36}$

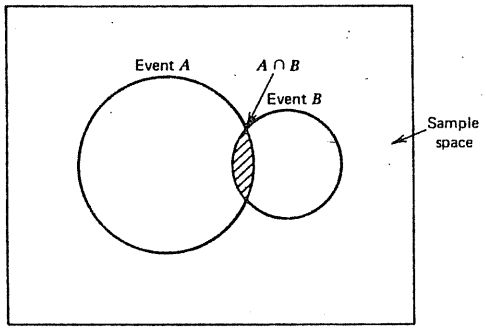


Figure 1.2 Venn diagram for two events A and B.

Suppose we now pose the further question: What is the probability that two "4's" are thrown? In our definition of the sample space, we suppressed the identity of the individual dice, so this simply is not a proper question for the probability space, as defined. This example will be continued, but first we digress for a moment to consider intersection of events. ▩

In addition to the set operations of union and complementation, the operation of intersection is also useful in probability theory. The intersection of two events A and B is the event containing points that are common to both A and B. This is illustrated in Fig. 1.2 with what is sometimes called a Venn diagram. The points lying within the heavy contour comprise the union of A and B, denoted as $A \cup B$ or "A or B." The points within the shaded region are the event "A intersection B," which is denoted $A \cap B$, or sometimes just "A and B."* The following relationship should be apparent just from the geometry of the Venn diagram:

$$P(A \cup B) = P(A) + P(B) - P(A \cap B) \tag{1.3.6}$$

The subtractive term in Eq. (1.3.6) is present because the probabilities in the overlapping region have been counted twice in the summation of $P(A)$ and $P(B)$.

The probability $P(A \cap B)$ is known as the *joint* probability of A and B and will be discussed further in Section 1.4. We digress for the moment, though, and look at two examples.

EXAMPLE 1.7 (continued)

We return to the two-dice example. Suppose we define event A as throwing a 4, 5, 6, or 7. Event B will be defined as throwing a 7, 8, 9, 10, or 11. We now pose two questions:

* In many references, the notation for "A union B" is "$A + B$," and "A intersection B" is shortened to just "AB." We will be proceeding to the study of random variables shortly, and then the chance occurrences will be related to real numbers, not "things." Thus, in order to avoid confusion, we will stay with the more cumbersome notation of $\cup$ and $\cap$ for set operations, and reserve $X + Y$ and XY to mean the usual arithmetic operations on real variables.

(a) What is the probability of event "A and B" (i.e., $A \cap B$)?
(b) What is the probability of event "A or B" (i.e., $A \cup B$)?

The answer to (a) is found by looking for the elements of the sample space that are common to both events A and B. Since the number 7 is the only common element,

$$P(A \cap B) = P(7) = \frac{1}{6}$$

The answer to (b) can be found either by itemizing the elements in $A \cup B$ or from Eq. (1.3.6). Using Eq. (1.3.6) leads to

$$\begin{aligned} P(A \cup B) &= P(A) + P(B) - P(A \cap B) \\ &= \frac{18}{36} + \frac{20}{36} - \frac{6}{36} \\ &= \frac{8}{9} \end{aligned}$$

This is easily verified using the direct itemization method. ■

EXAMPLE 1.8

Let us reconsider the dart-throwing experiment. We might consider a simple game where we draw a circle of radius R on the wall. If the player's dart lands on or within the circle, the player wins; if the dart lands outside R, the player loses. We pose the simple question: What is the probability that the player wins on the single throw of a dart?

In this example, the sample space is all the points on the wall. (We assume that the player can at least hit the wall.) Thus, there are an infinite number of points in our sample space S.

In this game, we define only two events on the sample space—event W, player wins; and event L, player loses. This is a legitimate set of events because the set operations of union, intersection, and complement yield defined events of the set.

The assignment of probabilities in this case must be made directly on the two events rather than on the sample space, because the probability of hitting *exactly* any particular point on the wall is zero and this tells us nothing about events. To add a note of realism, we might speculate that we are devising a simple gambling game for the house, one that the patrons will enjoy and that will also be profitable to the house. Our observations indicate that the typical player throws about 10 percent of his or her darts within a circle of radius R_1. In addition, the hits are more or less uniformly spaced within the circle. It would then be appropriate to assign probabilities to the two events as

$$\begin{aligned} P(\text{Hit lies on or within } R_1) &= .1 \\ P(\text{Hit lies outside } R_1) &= .9 \end{aligned}$$

If the establishment were to offer 5 to 1 odds, this game might well please the players and at the same time produce revenue for the house.

Before we leave this example, it should be mentioned that the radius R may be treated as a parameter, and hence we have the structure for looking at a whole family of problems, not just a single one. To add variety to the game, the proprietor might occasionally wish to decrease the diameter of the circle and increase the odds. Clearly, if the hits within the circle are nearly uniformly distributed, reducing the area by a factor of 2 will reduce the relative frequency of occurrence by a similar factor. Yet the corresponding reduction in radius is only a factor of $\sqrt{2}$, which should make the game "look" attractive. The appropriate probability assignments to this case would be .05 for "win" and .95 for "lose." Presumably, the house could offer 10 to 1 odds and still net the same average return as received on the 5-to-1 game. ■

1.4 JOINT AND CONDITIONAL PROBABILITY

In complex situations it is often desirable to arrange the elements of the sample space in arrays. This provides an orderly way of grouping related points in the space and is especially useful when considering successive trials of similar experiments. Consider the following example.

EXAMPLE 1.9

The Experiment Draw a card from a deck of 52 playing cards. Then replace the card, reshuffle the deck, and draw a card a second time. This is known as sampling with replacement. The sample space for this experiment, when viewed as a whole, consists of all possible pairs of cards for the two draws. This amounts to $(52)^2$ or 2704 elements—clearly an unwieldy number of elements to keep track of without some systematic way of ordering things. Yet, in spite of its size, the sample space is quite manageable when viewed as an array as shown in Table 1.3. This is a two-dimensional array, but it should be obvious that the

Table 1.3 Joint Probabilities for Two Draws with Replacement

First Draw \ Second Draw	Ace of Spades	King of Spades	⋯	Deuce of Clubs
Ace of spades	$\frac{1}{2704}$	$\frac{1}{2704}$		$\frac{1}{2704}$
King of spades	$\frac{1}{2704}$	$\frac{1}{2704}$		$\frac{1}{2704}$
⋮	⋯	⋯		⋯
(52 elements)	(Total of 2704 entries)			
Deuce of clubs	$\frac{1}{2704}$	$\frac{1}{2704}$		$\frac{1}{2704}$

concept is easily extended to higher-order situations—conceptually at least. Had we specified three draws rather than two, we would have a three-dimensional array, and so forth. Thus, the idea of an n-dimensional array associated with n trials of an experiment is an important concept. Note that summing out the numbers along any particular row yields the probability of drawing that particular card on the first draw irrespective of the result of the second draw. This leads to the idea of marginal probability, which will now be considered in a more general setting. ■

Let A and B loosely refer to "first" and "second" chance experiments. Time is not of the essence, so the experiments may be either successive or simultaneous as the situation dictates. Let $A_1, A_2, \ldots, A_m$ denote disjoint events associated with experiment A and, similarly, $B_1, B_2, \ldots, B_n$ denote disjoint events for experiment B. This leads to the joint-probability array shown in Table 1.4. Note first that n does not have to equal m, and therefore the array is not necessarily square. Also, the so-called *marginal probabilities* are shown in Table 1.4 to the right and bottom of the joint-probability array. These are the result of summing out rows or columns, as the case may be, and the term marginal arose because these probabilities are often written in the margins outside the $m \times n$ array. Clearly, summing out horizontally yields the probability of a particular event in the experiment A, irrespective of results of experiment B. Similarly, summing columns yields $P(B_1)$, $P(P_2)$, and so forth. Also, we tacitly assume that events $A_1, A_2, \ldots, A_m$ and $B_1, B_2, \ldots, B_n$ are all-inclusive as well as disjoint, so that the sum of the marginal probabilities, either vertically or horizontally, is unity.

Table 1.4 also contains information about the relative frequency of occurrence of various events in one set, given a particular event in the other set. For example, look at column 2, which lists $P(A_1 \cap B_2)$, $P(A_2 \cap B_2)$, $\ldots$, $P(A_m \cap B_2)$. Since no other entries in the table involve B_2, this list of numbers gives the relative distribution of events $A_1, A_2, \ldots, A_m$, given B_2 has occurred. However, the set of numbers appearing in column 2 is not a legitimate probability distribution because the sum is $P(B_2)$, not unity. So, imagine "renormalizing" all the entries in the column by dividing by $P(B_2)$. The new set of numbers is then $P(A_1 \cap B_2)/P(B_2)$, $P(A_2 \cap B_2)/P(B_2)$, $\ldots$, $P(A_m \cap B_2)/P(B_2)$, the sum is unity, and the relative distribution corresponds to the relative frequency of occurrence of $A_1, A_2, \ldots, A_m$, given B_2. This heuristic reasoning leads us to the formal definition of conditional probability.

Table 1.4 Array of Joint and Marginal Probabilities

A \ B	Event B_1	Event B_2	$\cdots$	Event B_n	Marginal Probabilities
Event A_1	$P(A_1 \cap B_1)$	$P(A_1 \cap B_2)$	$\cdots$	$P(A_1 \cap B_n)$	$P(A_1)$
Event A_2	$P(A_2 \cap B_1)$	$P(A_2 \cap B_2)$	$\cdots$	$P(A_2 \cap B_n)$	$P(A_2)$
$\vdots$	$\vdots$	$\vdots$		$\vdots$	$\vdots$
Event A_m	$P(A_m \cap B_1)$	$P(A_m \cap B_2)$	$\cdots$	$P(A_m \cap B_n)$	$P(A_m)$
Marginal Probabilities	$P(B_1)$	$P(B_2)$	$\cdots$	$P(B_n)$	Sum = 1

The *conditional probability* of A_i given B_j is defined as

$$P(A_i|B_j) = \frac{P(A_i \cap B_j)}{P(B_j)} \tag{1.4.1}$$

Similarly, the conditional probability of B_j given A_i is

$$P(B_j|A_i) = \frac{P(B_j \cap A_i)}{P(A_i)} \tag{1.4.2}$$

It is tacitly assumed in the above equations that $P(B_j)$ and $P(A_i)$ are not zero. Otherwise, conditional probability is not defined. It should also be emphasized that conditional probability is a *defined* concept and is not derived from other concepts. The discussion leading up to Eqs. (1.4.1) and (1.4.2) was presented to give an intuitive rationale for the definition and was not intended as a proof.

A useful relationship is obtained when Eqs. (1.4.1) and (1.4.2) are combined. Each equation may be solved for the probability of A_i intersection B_j and the results equated. This leads to *Bayes rule* (or *Bayes theorem*):

$$P(A_i|B_j) = \frac{P(B_j|A_i)P(A_i)}{P(B_j)} \tag{1.4.3}$$

This relationship is useful in reversing the conditioning of events. Note that the joint probability array $P(A_i \cap B_j)$ contains all the necessary information for computing *all* marginal and conditional probabilities. Conversely, if you know $P(A_i|B_j)$ and $P(B_j)$ [or $P(B_j|A_i)$ and $P(A_i)$], there is sufficient information to find $P(A_i \cap B_j)$.

EXAMPLE 1.10

For variety, we now consider an urn problem. The urn contains two red and two black balls. Two balls are drawn sequentially from the urn without replacement.

(a) What is the array of joint probabilities for the first and second draws?
(b) What is the conditional probability that the second draw is red, given the first draw is red?

To obtain the joint probability table, we first define a sample space consisting of all possible outcomes, including the identity of the individual balls. The four balls will be referred to as Red 1, Red 2, Black 1, and Black 2. The joint probability array for the first and second draws is given in Table 1.5. Note the effect of the "without replacement" statement. It gives rise to zeros along the major diagonal, because drawing Red 1 on the first draw precludes Red 1 being drawn again on the second draw, and so forth. In effect, there are really only 12 nontrivial outcomes for this experiment, and we assume they are all equally likely.

In the original problem, there was no mention of retaining the individuality of the two red and two black balls—only "red" and "black" were specified. Therefore, we can consolidate outcomes in accordance with the partitioning

Table 1.5 Joint Probability Table for Four-Ball Urn Example

First Draw \ Second Draw	Red 1	Red 2	Black 1	Black 2
Red 1	0	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{1}{12}$
Red 2	$\frac{1}{12}$	0	$\frac{1}{12}$	$\frac{1}{12}$
Black 1	$\frac{1}{12}$	$\frac{1}{12}$	0	$\frac{1}{12}$
Black 2	$\frac{1}{12}$	$\frac{1}{12}$	$\frac{1}{12}$	0

shown by dashed lines in Table 1.5. This leads to the two-by-two array shown in Table 1.6. This, then, is the answer to part (a).

For the conditional probability we will use the basic definition given by Eq. (1.4.2). Writing this out explicitly for the conditional situation posed in question (b), we have

$$P(\text{Second draw red}|\text{First draw red}) = \frac{P(\text{First draw red and second draw red})}{P(\text{First draw red})} \tag{1.4.4}$$

The numerator of Eq. (1.4.4) is the upper-left entry in Table 1.6. The denominator is the marginal probability obtained by summing the elements of the first row in Table 1.6. This yields

$$P(\text{Second draw red}|\text{First draw red}) = \frac{\frac{1}{6}}{\frac{1}{2}} = \frac{1}{3} \tag{1.4.5}$$

This is the solution to part (b). Note this checks with the result one would obtain by considering the three balls that remain in the urn after a red one is withdrawn. ■

Table 1.6 Joint Probability Table Reduced to Two-by-Two Array

First Draw \ Second Draw	Red	Black
Red	$\frac{1}{6}$	$\frac{1}{3}$
Black	$\frac{1}{3}$	$\frac{1}{6}$

1.5 INDEPENDENCE

In qualitative terms, two events are said to be independent if the occurrence of one does not affect the likelihood of the other. If we toss two coins simultaneously, we would not expect the outcome of one of the coins to affect the other. Similarly, if we draw a card from a deck of 52 playing cards, then replace it, reshuffle, and draw a second time, we would not expect the second outcome to be affected by the first. However, if we draw the second card without replacing the first, it is a much different matter. For example, the probability of drawing an ace on the second draw with replacement is $\frac{4}{52}$. However, if we draw an ace on the first draw and do not replace it, the probability of getting an ace on the second draw is only $\frac{3}{51}$. In the "without replacement" experiment, the outcome of the first draw certainly affects the chances on the second draw, so the two events are not independent.

Formally, events A and B are said to be *independent* if

$$P(A \cap B) = P(A)P(B) \tag{1.5.1}$$

Also, it should be evident from Eq. (1.5.1) and the defining equations for conditional probability, Eqs. (1.4.1) and (1.4.2), that if A and B are independent

$$\left.\begin{aligned} P(A|B) &= P(A) \\ P(B|A) &= P(B) \end{aligned}\right\} \quad \begin{matrix} \text{For } A \text{ and } B \\ \text{independent only} \end{matrix} \tag{1.5.2}$$

We might also note that the defining equation for independence, Eq. (1.5.1), usually provides the simplest test for independence. This is illustrated with two examples.

EXAMPLE 1.11

The joint probability array for the simultaneous toss of two coins is given in Table 1.7. The marginal probabilities are also shown in the "margins" with their significance stated in words in parentheses. Note that each of the four joint probabilities of the array may be written as the product of their respective marginal probabilities. Thus, all events are independent in this case. ■

EXAMPLE 1.12

Let us reconsider the urn experiment described in Example 1.10, Section 1.4. Recall that the two balls were withdrawn sequentially *without* replacement, and that this led to the joint probability array shown in Table 1.8. The marginal probabilities are also included, just as in the previous example. However, in this case none of the joint probabilities can be written as the product of the respective marginal probabilities. Thus, all event pairs are dependent. ■

To recapitulate, we say events A and B are independent if their joint probability can be written as the product of the individual total probabilities, $P(A)$ and $P(B)$. Otherwise, they are said to be dependent.

Table 1.7 Joint and Marginal Probabilities for Toss of Two Coins

First Coin \ Second Coin	Heads	Tails	
Heads	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$ (Prob. first coin is heads)
Tails	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{2}$ (Prob. first coin is tails)
	$\frac{1}{2}$ (Prob. second coin is heads)	$\frac{1}{2}$ (Prob. second coin is tails)	

1.6 RANDOM VARIABLES

In the study of noiselike signals, we are nearly always dealing with physical quantities such as voltage, torque, distance, and so forth, which can be measured in physical units. In these cases, the chance occurrences are related to real numbers, not just "things" like heads or tails, black balls or red balls, and the like. This brings us to the notion of a random variable. Let us say we have a conceptual experiment for which we have defined a suitable sample space, an appropriate set of events, and a probability assignment for the set of events. A *random variable* is simply a function that maps every point in the same space (things) on to the real line (numbers). A simple example of this mapping is shown in Fig. 1.3. Note that each face of the die is embossed with a pattern of dots, not a number. The assignment of numbers is our own doing and could be almost anything. In Fig. 1.3 we just happened to choose the most common numerical assignment, namely, the sum of the number of dots, but this was not necessary.

Presumably, in our probability space, probabilities have been assigned to the events in the sample space in accordance with the basic axioms of probability. Associated with each event in the original sample space (things) there will be a corresponding event in the random-variable space (numbers). These

Table 1.8 Joint and Marginal Probabilities for Urn Example

First Draw \ Second Draw	Red	Black	
Red	$\frac{1}{6}$	$\frac{1}{3}$	$\frac{1}{2}$
Black	$\frac{1}{3}$	$\frac{1}{6}$	$\frac{1}{2}$
	$\frac{1}{2}$	$\frac{1}{2}$	

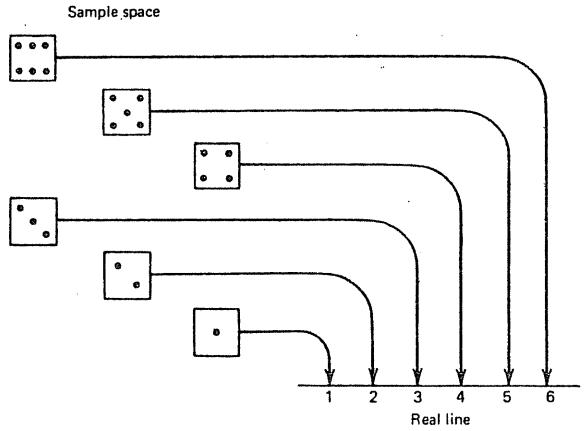


Figure 1.3 Mapping for the throw of one die.

will be called *equivalent events,* and it is only natural that we should assign probabilities to the random-variable events in the same manner as for the original sample-space events. Stated formally, we have

$$P(\text{equivalent event on the real line}) = P(\text{corresponding event in the original sample space}) \quad (1.6.1)$$

Two more examples will illustrate the concept of a random variable further.

EXAMPLE 1.13

The mapping that defines a random variable must fit the chance situation at hand if it is to be useful. Sometimes this leads to unusual but perfectly legitimate functional relationships. For example, in the game of pitch, a portion of the scoring is done by summing the card values of the cards each player takes in tricks during the course of play. The card values, by arbitrary rules of the game, are as follows:

Card of Any Suit	Card Value
2 through 9	0
10	10
Jack	1
Queen	2
King	3
Ace	4

Thus, in exploring your chances relative to this aspect of the game, it would be appropriate to map the 52 points in the sample space (for a single card) into

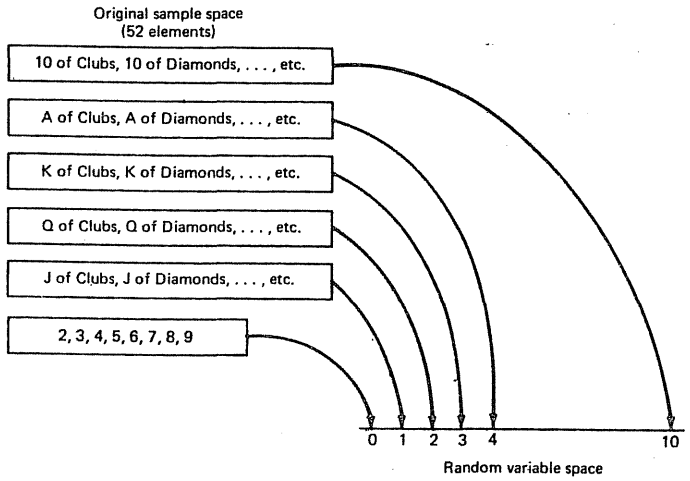


Figure 1.4 Mapping for pitch example.

real numbers in accordance with the above table. This is also shown in Fig. 1.4. Note that multiple points in the sample space map into the *same* number of the real line. This is perfectly legitimate. The mapping must not be ambiguous in going from the sample space to the real line; however, it may be ambiguous going the other way. That is, the mapping need not be one-to-one. ■

EXAMPLE 1.14

In many games, the player spins a pointer that is mounted on a circular card of some sort and is free to spin about its center. This is depicted in Fig. 1.5 and the circular card is intentionally shown without any markings along its edge. Suppose we define the outcome of an experiment as the location on the periphery of the card at which the pointer stops. The sample space then consists of an infinite number of points along a circle. For analysis purposes, we might wish to identify each point in the sample space in terms of an angular coordinate measured in radians. The functional mapping that maps all points on a circle to

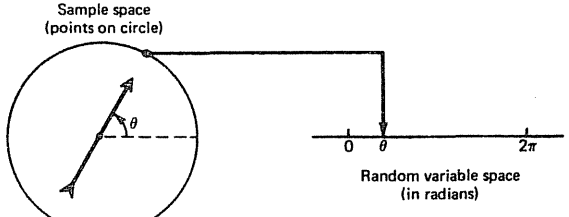


Figure 1.5 Mapping for spin-the-pointer example.

corresponding points on the real line between 0 and 2π would then define an appropriate random variable. ■

1.7 PROBABILITY DISTRIBUTION AND DENSITY FUNCTIONS

When the sample space consists of a finite number of elements, the probability assignment can be made directly on the sample-space elements in accordance with what we feel to be reasonable in terms of likelihood of occurrence. This then defines probabilities for all events defined on the sample space. These probabilities, in turn, transfer directly to equivalent events in the random-variable space. The allowable realizations (i.e., real numbers) in the random-variable space are elementary equivalent events themselves, so the result is a probability associated with each allowable realization in the random-variable space. The sum of these probabilities must be unity, just as in the original sample space, but the distribution need not be the same. A continuation of Example 1.13, Section 1.6, will illustrate this.

EXAMPLE 1.15

The mapping from the sample space to the real line for the pitch card game was shown in Fig. 1.4. Let us assign equal probabilities for all elements in the original sample space. The probabilities for the allowable realizations in the random-variable space would then be:

Random Variable Realization	Probability
0	$\frac{32}{52}$
1	$\frac{4}{52}$
2	$\frac{4}{52}$
3	$\frac{4}{52}$
4	$\frac{4}{52}$
10	$\frac{4}{52}$

Note that the probabilities are not distributed uniformly in the random-variable space. The end result of the mapping is a set of real numbers representing the possible realizations of the random variable and a corresponding set of probabilities that sum to unity. Once this correspondence has been established, the original sample space is usually ignored. ■

The random variable of Example 1.15 is an example of a *discrete* random variable in that its allowable realizations are discrete (i.e., countable) rather than continuous. The associated discrete set of probabilities is sometimes referred to as the *probability mass distribution* or simply *probability distribution.*

We also have occasion to work with continuous random variables. As a matter of fact, the usual electronic noise that is encountered in a wide variety

of applications is of this type, that is, the voltage (or current) may assume a continuous range of values. The corresponding sample space then also contains an infinite number of points, so we cannot assign probabilities directly on the points of the sample space; this must be done on the defined events. We will continue the spin-the-pointer example of Section 1.6 (Example 1.14) to illustrate how this is done.

Let X denote a continuous random variable corresponding to the angular position of the pointer after it stops. Presumably, this could be any angle between 0 and 2 π radians; therefore, the probability of any particular position is infinitesimal. Thus, we assign a probability to the event that the pointer stops within a certain continuous range of values, say, between 0 and θ radians. If all positions are equally likely, it is reasonable to assign probabilities as follows (within the admissible range of θ):

$$P(0 \leq X \leq \theta) = \begin{cases} \dfrac{1}{2\pi}\theta, & 0 \leq \theta \leq 2\pi \end{cases} \tag{1.7.1}$$

Note that the probability assignment is a function of the parameter θ and the function is sketched in Fig. 1.6. The linear portion of the function between 0 and 2π is due to the "equally likely" assumption.

The function sketched in Fig. 1.6 is known as a *cumulative distribution function* (or just probability distribution function), and it simply describes the probability assignment as it reflects onto equivalent events in the random variable space. Specifically, the probability distribution function associated with the random variable X is defined as

$$F_X(\theta) = P(X \leq \theta) \tag{1.7.2}$$

where θ is a parameter representing a realization of X.

Just as in the discrete case, once this distribution is established in the random-variable space, the original sample space is usually ignored. It should be clear from the definition that a probability distribution function always has the following properties:

1. $F_X(\theta) \to 0, \quad$ as $\theta \to -\infty$.

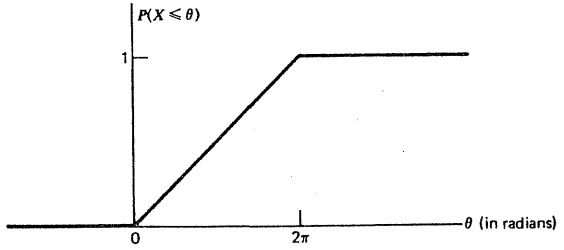


Figure 1.6 Probability distribution function for pointer example.

2. $F_X(\theta) \to 1, \quad$ as $\theta \to \infty$.
3. $F_X(\theta)$ is a nondecreasing function of θ.

The information contained in the distribution function (e.g., Fig. 1.6) may also be presented in derivative form. Specifically, let $f_X(\theta)$ be defined as

$$f_X(\theta) = \frac{d}{d\theta} F_X(\theta) \tag{1.7.3}$$

The function $f_X(\theta)$ is known as the *probability density function* associated with the random variable X. The density function for the pointer example is shown in Fig. 1.7. From properties 1, 2, and 3 just cited for the distribution function, it should be apparent that the density function has the following properties:

1. $f_X(\theta)$ is a nonnegative function.
2. $\int_{-\infty}^{\infty} f_X(\theta)\, d\theta = 1$.

It should also be apparent from elementary calculus that the shaded area shown in Fig. 1.7 represents the probability that X lies between θ_1 and θ_2. If θ_1 and θ_2 are separated by an infinitesimal amount, $\Delta\theta$, the area is approximately $f_X(\theta_1)\,\Delta\theta$, and thus we have the term probability *density*.

The probability density and distribution functions are alternative ways of describing the relative distribution of the random variable. Both functions are useful in random-variable analysis, and you should always keep in mind the derivative/integral relationship between the two. As a matter of notation, we will normally use an uppercase symbol for the distribution function and the corresponding lowercase symbol for the density function. The subscript in each case indicates the random variable being considered. The argument of the function is a dummy variable and may be almost anything.

1.8 EXPECTATION, AVERAGES, AND CHARACTERISTIC FUNCTION

The idea of averaging is so commonplace that it may not seem worthy of elaboration. Yet there are subtleties, especially as averaging relates to probability. Thus we need to formalize the notion of average.

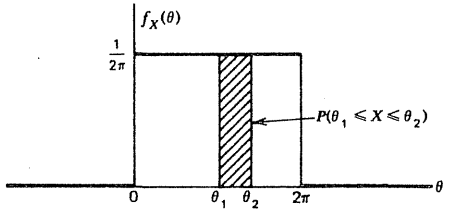


Figure 1.7 Probability density function for pointer example.

Perhaps the first thing to note is that we always average over numbers and not "things." There is no such thing as the average of apples and oranges. When we compute a student's average grades, we do not average over A, B, C, and so on; instead, we average over numerical equivalents that have been arbitrarily assigned to each grade. Also, the quantities being averaged may or may not be governed by chance. In either case, random or deterministic, the average is just the sum of the numbers divided by the number of quantities being averaged. In the random case, the *sample average* or *sample mean* of a random variable X is defined as

$$\overline{X} = \frac{X_1 + X_2 + \cdots + X_N}{N} \tag{1.8.1}$$

where the bar over X indicates average, and $X_1, X_2, \ldots,$ are sample realizations obtained from repeated trials of the chance situation under consideration. We will use the terms *average* and *mean* interchangeably, and the adjective *sample* serves as a reminder that we are averaging over a finite number of trials as in Eq. (1.8.1).

In the study of random variables we also like to consider the conceptual average that would occur for an infinite number of trials. This idea is basic to the relative-frequency concept of probability. This hypothetical average is called *expected value* and is aptly named; it simply refers to what one would "expect" in the typical statistical situation. Beginning with discrete probability, imagine a random variable whose n possible realizations are $x_1, x_2, \ldots, x_n$. The corresponding probabilities are $p_1, p_2, \ldots, p_n$. If we have N trials, where N is large, we would expect approximately p_1N x_1's, p_2N x_2's, etc. Thus, the sample average would be

$$\overline{X} \approx \frac{(p_1N)x_1 + (p_2N)x_2 + \cdots + (p_nN)x_n}{N} \tag{1.8.2}$$

This suggests the following definition for expected value for the discrete probability case:

$$\text{Expected value of } X = E(X) = \sum_{i=1}^{n} p_i x_i \tag{1.8.3}$$

where n is the number of allowable values of the random variable X.

Similarly, for a continuous random variable X, we have

$$\text{Expected value of } X = E(X) = \int_{-\infty}^{\infty} x f_X(x)\, dx \tag{1.8.4}$$

It should be mentioned that Eqs. (1.8.3) and (1.8.4) are definitions, and the arguments leading up to these definitions were presented to give a sensible rationale for the definitions, and not as a proof. We can use these same arguments for defining the expectation of a function of X, as well as for X. Thus, we have the following:

Discrete case:

$$E(g(X)) = \sum_{i}^{n} p_i g(x_i) \tag{1.8.5}$$

Continuous case:

$$E(g(X)) = \int_{-\infty}^{\infty} g(x) f_X(x)\, dx \tag{1.8.6}$$

As an example of the use of Eq. (1.8.6), let the function $g(X)$ be X^k. Equation (1.8.6) [or its discrete counterpart Eq. (1.8.5)] then provides an expression for the kth *moment* of X, that is,

$$E(X^k) = \int_{-\infty}^{\infty} x^k f_X(x)\, dx \tag{1.8.7}$$

The second moment of X is of special interest, and it is given by

$$E(X^2) = \int_{-\infty}^{\infty} x^2 f_X(x)\, dx \tag{1.8.8}$$

The first moment is, of course, just the expectation of X, which is also known as the *mean* or *average value* of X. Note that when the term *sample* is omitted, we tacitly assume that we are referring to the hypothetical infinite-sample average.

We also have occasion to look at the second moment of X "about the mean." This quantity is called the *variance* of X and is defined as

$$\text{Variance of } X = E[(X - E(X))^2] \tag{1.8.9}$$

In a qualitative sense, the variance of X is a measure of the dispersion of X about its mean. Of course, if the mean is zero, the variance is identical to the second moment.

The expression for variance given by Eq. (1.8.9) can be reduced to a more convenient computational form by expanding the quantity within the brackets and then noting that the expectation of the sum is the sum of the expectations. This leads to

$$\begin{aligned} \text{Var } X &= E[X^2 - 2X \cdot E(X) + (E(X))^2] \\ &= E(X^2) - (E(X))^2 \end{aligned} \tag{1.8.10}$$

The square root of the variance is also of interest, and it has been given the name *standard deviation,* that is,

$$\text{Standard deviation of } X = \sqrt{\text{variance of } X} \tag{1.8.11}$$

EXAMPLE 1.16

Let X be uniformly distributed in the interval $(0, 2\pi)$. This leads to the probability density function (see Example 1.14).

$$f_X(x) = \begin{cases} \dfrac{1}{2\pi}, & 0 \le x < 2\pi \\ 0, & \text{elsewhere} \end{cases}$$

Find the mean, variance, and standard deviation of X.

The mean is just the expectation of X and is given by Eq. (1.8.4).

$$\text{Mean of } X = E(X) = \int_0^{2\pi} x \cdot \frac{1}{2\pi}\, dx$$

$$= \frac{1}{2\pi} \cdot \frac{x^2}{2}\bigg|_0^{2\pi} = \pi \tag{1.8.12}$$

Now that we have computed the mean, we are in a position to find the variance from Eq. (1.8.10).

$$\text{Var } X = \int_0^{2\pi} x^2 \frac{1}{2\pi}\, dx - \pi^2$$

$$= \frac{4}{3}\pi^2 - \pi^2$$

$$= \frac{1}{3}\pi^2 \tag{1.8.13}$$

The standard deviation is now just the square root of the variance:

$$\text{Standard deviation of } X = \sqrt{\text{Var } X}$$

$$= \sqrt{\frac{1}{3}\pi^2} = \frac{1}{\sqrt{3}}\pi \tag{1.8.14}$$

■

The *characteristic function* associated with the random variable X is defined as

$$\psi_X(\omega) = \int_{-\infty}^{\infty} f_X(x) e^{j\omega x}\, dx \tag{1.8.15}$$

It can be seen that $\psi_X(\omega)$ is just the Fourier transform of the probability density function with a reversal of sign on ω. Thus, the theorems (and tables) of Fourier transform theory can be used to advantage in evaluating characteristic functions and their inverses.

The characteristic function is especially useful in evaluating the moments of X. This can be demonstrated as follows. The moments of X may be written as

$$E(X) = \int_{-\infty}^{\infty} x f_X(x)\, dx \tag{1.8.16}$$

$$E(X^2) = \int_{-\infty}^{\infty} x^2 f_X(x)\, dx$$

$$\vdots$$

$$\text{etc.} \tag{1.8.17}$$

Now consider the derivatives of $\psi_x(\omega)$ evaluated at $\omega = 0$.

$$\left[\frac{d\psi_X}{d\omega}\right]_{\omega=0} = \left[\int_{-\infty}^{\infty} jx f_X(x) e^{j\omega x}\, dx\right]_{\omega=0} = \int_{-\infty}^{\infty} jx f_X(x)\, dx \tag{1.8.18}$$

$$\left[\frac{d^2\psi_X}{d\omega^2}\right]_{\omega=0} = \left[\int_{-\infty}^{\infty} (jx)^2 f_X(x) e^{j\omega x}\, dx\right]_{\omega=0} = \int_{-\infty}^{\infty} j^2 x^2 f_X(x)\, dx$$

$$\vdots$$

$$\text{etc.} \tag{1.8.19}$$

It can be seen that

$$E(X) = \frac{1}{j}\left[\frac{d\psi_X}{d\omega}\right]_{\omega=0} \tag{1.8.20}$$

$$E(X^2) = \frac{1}{j^2}\left[\frac{d^2\psi_X}{d\omega^2}\right]_{\omega=0}$$

$$\vdots$$

$$\text{etc.} \tag{1.8.21}$$

Thus, with the help of a table of Fourier transforms, you can often evaluate the moments without performing the integrations indicated in their definitions. [See Problems 1.21(b) and 2.30 for applications of the characteristic function.]

1.9 NORMAL OR GAUSSIAN RANDOM VARIABLES

The random variable X is called *normal* or *Gaussian* if its probability density function is

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2\sigma^2}(x - m_X)^2\right] \tag{1.9.1}$$

Note that this density function contains two parameters m_X and σ^2. These are the random variable's mean and variance. That is, for the f_X specified by Eq. (1.9.1),

$$\int_{-\infty}^{\infty} x f_X(x)\, dx = m_X \tag{1.9.2}$$

and

$$\int_{-\infty}^{\infty} (x - m_X)^2 f_X(x)\, dx = \sigma^2 \tag{1.9.3}$$

Note that the normal density function is completely specified by assigning numerical values to the mean and variance. Thus, a shorthand notation has come into common usage to designate a normal random variable. When we write

$$X \sim N(m_X, \sigma^2) \tag{1.9.4}$$

we mean X is normal with its mean given by the first argument in parentheses and its variance by the second argument. Also, as a matter of terminology, the terms normal and Gaussian are used interchangeably in describing normal random variables, and we will make no distinction between the two.

The normal density and distribution functions are sketched in Figs. 1.8*a* and 1.8*b*. Note that the density function is symmetric and peaks at its mean. Qualitatively, then, the mean is seen to be the most likely value, with values on either side of the mean gradually becoming less and less likely as the distance from the mean becomes larger. Since many natural random phenomena seem to exhibit this central-tendency property, at least approximately, the normal distribution is encountered frequently in applied probability. Recall that the variance is a measure of dispersion about the mean. Thus, small σ corresponds to a sharp-peaked density curve, whereas large σ will yield a curve with a flat peak.

The normal distribution function is, of course, the integral of the density function:

$$F_X(x) = \int_{-\infty}^{x} f_X(u)\, du = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{1}{2\sigma^2}(u - m_X)^2\right] du \tag{1.9.5}$$

Unfortunately, this integral cannot be represented in closed form in terms of elementary functions. Thus, its value must be obtained from tables or by numerical integration. A brief tabulation for zero mean and unity variance is given in Table 1.9. A quick glance at the table will show that the distribution is very close to unity for values of the argument greater than 4.0 (i.e., 4σ). In our table, which was taken from Feller (5), the tabulation quits at about this point. In some applications, though, the difference between $F_X(x)$ and unity [i.e., the area under the "tail" of $f_X(x)$] is very much of interest, even though it is quite small. Tables

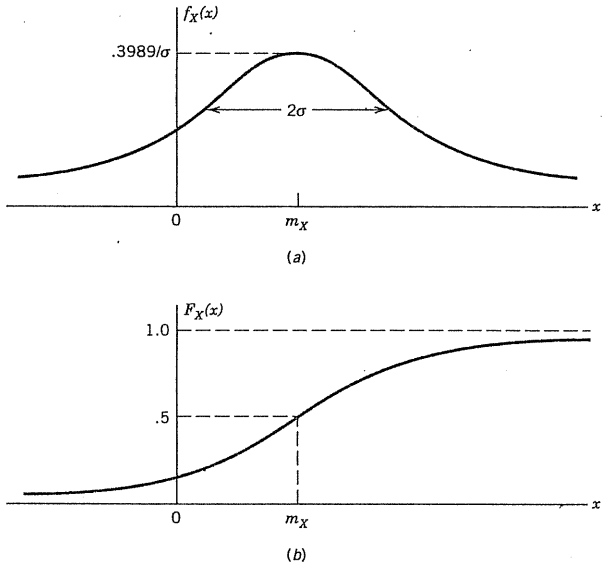


Figure 1.8 (*a*) Normal density function. (*b*) Normal distribution function.

such as the one given here are not of much use in such cases, because the range of x is limited and the resolution is poor. Fortunately, it is relatively easy to integrate the normal density function numerically using software such as MATLAB's quad8 or quad. In evaluating the tail of the density function, it is better to integrate $f_X(u)$ from x to some large finite upper limit (representing ∞) than to evaluate $1 - F_X(x)$ literally. Some trial-and-error experimentation is usually necessary in determining a suitable upper limit that will yield the desired accuracy and, at the same time, will allow reasonably fast convergence of the numerical integration. This is manageable, though, and there are approximate formulas to help in determining an appropriate upper limit in the integration (see Problem 1.40*).

The numerical integration of f_X is also recommended for evaluating the probability that X lies between any finite limits, as well as in the "tail of the distribution" problem. The accuracy and resolution obtained from numerical integration are nearly always better than those obtained from tables. For example, say we want the probability that $2.5 < X < 3.0$. From Table 1.9 we get

$$P[2.5 < X < 3.0] \approx .998650 - .993790 = .004860$$

Using MATLAB's quad8 and quad, we get

$$P[2.5 < X < 3.0] \approx .00485977$$

where the decimal number has been truncated at the point where quad8 and quad begin to disagree on the result.

Table 1.9 The Normal Density and Distribution Functions for Zero Mean and Unity Variance

$$f_X(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2}, \qquad F_X(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-u^2/2}\, du$$

x	$f_X(x)$	$F_X(x)$	x	$f_X(x)$	$F_X(x)$
.0	.398 942	.500 000	2.3	.028 327	.989 276
.1	.396 952	.539 828	2.4	.022 395	.991 802
.2	.391 043	.579 260	2.5	.017 528	.993 790
.3	.381 388	.617 911	2.6	.013 583	.995 339
.4	.368 270	.655 422	2.7	.010 421	.996 533
.5	.352 065	.691 462	2.8	.007 915	.997 445
.6	.333 225	.725 747	2.9	.005 953	.998 134
.7	.312 254	.758 036	3.0	.004 432	.998 650
.8	.289 692	.788 145	3.1	.003 267	.999 032
.9	.266 085	.815 940	3.2	.002 384	.999 313
1.0	.241 971	.841 345	3.3	.001 723	.999 517
1.1	.217 852	.864 334	3.4	.001 232	.999 663
1.2	.194 186	.884 930	3.5	.000 873	.999 767
1.3	.171 369	.903 200	3.6	.000 612	.999 841
1.4	.149 727	.919 243	3.7	.000 425	.999 892
1.5	.129 581	.933 193	3.8	.000 292	.999 928
1.6	.110 921	.945 201	3.9	.000 199	.999 952
1.7	.094 049	.955 435	4.0	.000 134	.999 968
1.8	.078 950	.964 070	4.1	.000 089	.999 979
1.9	.065 616	.971 283	4.2	.000 059	.999 987
2.0	.053 991	.977 250	4.3	.000 039	.999 991
2.1	.043 984	.982 136	4.4	.000 025	.999 995
2.2	.035 475	.986 097	4.5	.000 016	.999 997

In spite of the previous remarks, tables of probabilities, both normal and otherwise, can be useful for quick and rough calculations. A word of caution is in order, though, relative to normal distribution tables. They come in a variety of forms. For example, some tables give the one-sided area under the normal density curve from 0 to X, rather than from $-\infty$ to X as we have done in Table 1.9. Other tables do something similar by tabulating a function known as the error function (6), which is normalized differently than the usual distribution function. Thus a word of warning is in order. Be wary of using unfamiliar tables!

1.10 IMPULSIVE PROBABILITY DENSITY FUNCTIONS

In the case of the normal distribution and many others, the probability associated with the random variable X is smoothly distributed over the real line from $-\infty$ to ∞. The corresponding probability density function is then continuous, and the probability that any particular value of X, say, x_0, is realized is zero. This situation is common in physical problems, but we also have occasion to consider cases where the random variable has a mixture of discrete and smooth distribution. Rectification or any sort of hard limiting of noise leads to this situation, and an example will illustrate how this affects the probability density and distribution functions.

EXAMPLE 1.17

Consider a simple half-wave rectifier driven by noise as shown in Fig. 1.9. For our purpose here, it will suffice to assume that the amplitude of the noise is normally distributed with zero mean. That is, if we were to sample the input at any particular time t_1, the resultant sample is a random variable, say, X, whose distribution is $N(0, \sigma_x^2)$. The corresponding output sample is, of course, a different random variable; it will be denoted as Y.

Because of our assumption of normality, X will have probability density and distribution functions as shown in Fig. 1.8, but with $m_X = 0$. The sample space in this case may be thought of as all points on the real line, and the function defining the random variable X is just a one-to-one mapping of the sample space into X. Not so with Y though; all positive points in the sample space map one-to-one, but the negative points all map into zero because of the diode! This means that a total probability of $\frac{1}{2}$ in the sample space gets squeezed into a single point, zero, in the Y space. The effect of this on the density and distribution functions for Y is shown in Figs. 1.10*a* and 1.10*b*. Note that in order to have the area under the density function be $\frac{1}{2}$ at $y = 0$, we must have a Dirac delta (impulse) function at the origin. This, in turn, gives rise to a jump or discontinuity in the corresponding distribution function. It should be mentioned that at the point of discontinuity, the value of the distribution function is $\frac{1}{2}$. That is, the distribution function is continuous from the right and not from the left. This is due to the "*equal to* or less than . . ." statement in the definition of the probability distribution function (see Section 1.8). ■

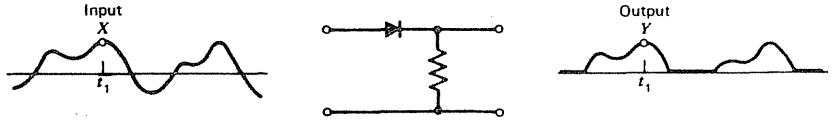


Figure 1.9 Half-wave rectifier driven by noise.

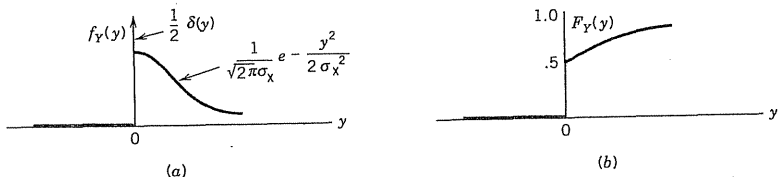


Figure 1.10 Output density and distribution functions for Example 1.17. (*a*) Probability density function for *Y*. (*b*) Probability distribution function for *Y*.

1.11 MULTIPLE RANDOM VARIABLES

In subsequent chapters, we will frequently have occasion to deal with multiple random variables and their mutual relationships. Multiple random variables are often referred to as *multivariate* random variables, with *bivariate* being the special case of two variables. The various probabilistic relationships will be illustrated here for the bivariate case. The extension to three or more random variables is straightforward, so it will not be discussed specifically.

Let us first consider two *discrete* random variables X and Y. By discrete we mean that X and Y may take on discrete values x_i and y_j, where i and j are certain allowed integers. We will define the *joint probability distribution* (or *joint probability mass distribution*) as

$$p_{XY}(x_i, y_j) = P(X = x_i \text{ and } Y = y_j) \tag{1.11.1}$$

Note that this is not a cumulative distribution and, as an extra reminder, we will denote the discrete distribution with p rather than F. Just as in the case of joint events A and B (Section 1.4), the joint distribution of X and Y can be thought of as a two-dimensional array of probabilities, with each element in the array representing the probability of occurrence of a particular combination of X and Y. The sum of the numbers in the array obviously must be unity. Also, summing horizontally or vertically yields the marginal probabilities, just as in the "events" case. An example will illustrate these concepts.

EXAMPLE 1.18

A sack contains 2 pennies, 1 nickel, and 1 dime. A coin is withdrawn at random and then replaced; then a second coin is withdrawn. The face value of the first draw (i.e., possible values are 1, 5, or 10) will be called random variable X, and the value of the second coin will be Y. We will assume that the outcome of the first draw does not affect the second in any way. The sample space in this case consists of 16 elements, and it, along with mapping into the bivariate random variable (X,Y), is shown in Fig. 1.11. The two pennies are distinguished as Pen 1 and Pen 2. Note that the 16 elements of the sample space map into 9 2-tuples in the random-variable space. We have shown the 2-tuples in Fig. 1.11 as two numbers separated by commas. We could, of course, have shown them as two-

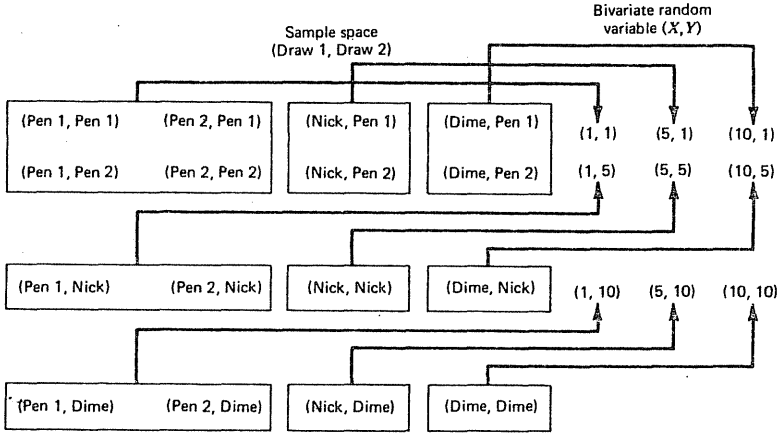


Figure 1.11 Sample space and mapping for Example 1.18.

element column vectors; it's purely a matter of notation. If we assume that all elementary events in the sample space are equally likely, each should be assigned a probability of $\frac{1}{16}$. The joint probability distribution for X and Y is then as shown in Table 1.10. ▩

Just as in Section 1.4, we can write certain relationships among the joint, marginal, and conditional probabilities for random variables X and Y.

Marginal (unconditional) probability:

$$p_X(x_i) = \sum_j p_{XY}(x_i, y_j) \tag{1.11.2}$$

$$p_Y(y_j) = \sum_i p_{XY}(x_i, y_j) \tag{1.11.3}$$

Table 1.10 Joint and Marginal Probabilities for X and Y (X is First Draw and Y is Second Draw)

X \ Y	1	5	10	Marginal Probability $p(x_i)$
1	$\frac{1}{4}$	$\frac{1}{8}$	$\frac{1}{8}$	$\frac{1}{2}$
5	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{4}$
10	$\frac{1}{8}$	$\frac{1}{16}$	$\frac{1}{16}$	$\frac{1}{4}$
Marginal probability $p(y_j)$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	Sum is unity when summed either horizontally or vertically

Conditional probability:

$$p_{X|Y} = \frac{p_{XY}}{p_Y} \tag{1.11.4}$$

$$p_{Y|X} = \frac{p_{XY}}{p_X} \tag{1.11.5}$$

Bayes rule:

$$p_{X|Y} = \frac{p_{Y|X}p_X}{p_Y} \tag{1.11.6}$$

The arguments in Eqs. (1.11.4) to (1.11.6) were omitted because the permissible x_i and y_j are obvious.

The discrete random variables X and Y are defined to be *statistically independent* if

$$p_{XY}(x_i, y_j) = p_X(x_i)p_Y(y_j) \tag{1.11.7}$$

for all allowable x_i and y_j. As an example, the random variables in Example 1.18 are statistically independent because all nine probabilities in Table 1.10 satisfy the product relationship given by Eq. (1.11.7).

Let us now turn our attention to continuous random variables. Just as in the single-variable case, the relative distribution in the multivariate case must be described in terms of either a cumulative distribution function or a density function. Let X and Y be continuous random variables. The *joint cumulative distribution function* is defined* as

$$F_{XY}(x, y) = P(X \le x \text{ and } Y \le y) \tag{1.11.8}$$

Clearly, F_{XY} has the following properties:

$$\text{(a) } F_{XY}(-\infty, -\infty) = 0 \tag{1.11.9}$$

$$\text{(b) } F_{XY}(\infty, \infty) = 1 \tag{1.11.10}$$

$$\text{(c) } F_{XY} \text{ is nondecreasing in } x \text{ and } y \tag{1.11.11}$$

The joint density function of continuous random variables X and Y is given by

$$f_{XY}(x, y) = \frac{\partial^2 F_{XY}(x, y)}{\partial x\, \partial y} \tag{1.11.12}$$

* The term "distribution" is a bit overworked in probability theory. The cumulative distribution of continuous random variables is much different than the probability distribution associated with discrete random variables. Thus we will append the word "cumulative" here to avoid confusion.

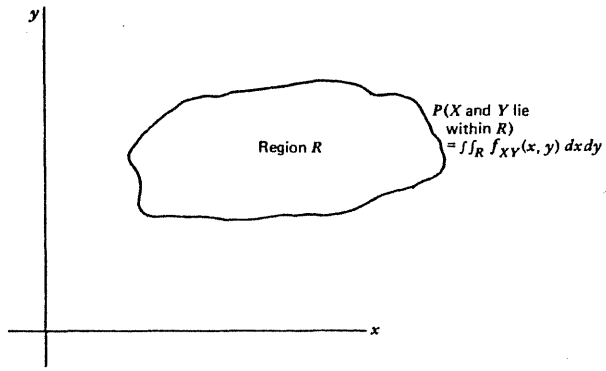


Figure 1.12 Region R in xy plane.

Note that there is an integral/derivative relationship between the cumulative distribution and density functions, just as in the single-variate case. Thus, to find the probability that a joint realization of X and Y will lie within a certain region R in the xy plane, we use the double integral formula

$$P(X \text{ and } Y \text{ lie within } R) = \iint_R f_{XY}(x, y)\, dx\, dy \tag{1.11.13}$$

This is shown geometrically in Fig. 1.12. Of course, if the region R is a differential rectangle, as shown in Fig. 1.13, the probability of X and Y lying within the rectangle is

$$P(x_0 \le X \le x_0 + dx \text{ and } y_0 \le Y \le y_0 + dy) = f_{XY}(x_0, y_0)\, dx\, dy \tag{1.11.14}$$

Thus it should be apparent that f_{XY} has the usual meaning of density in a two-dimensional sense.

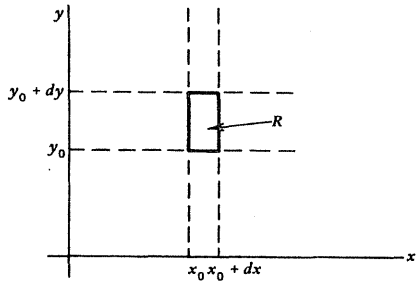


Figure 1.13 Differential area R in xy plane.

The *marginal* or *unconditional densities* are obtained in a manner similar to the discrete case, except that summation is replaced with integration. Thus, we have

$$f_X(x) = \int_{-\infty}^{\infty} f_{XY}(x, y)\, dy \tag{1.11.15}$$

and

$$f_Y(y) = \int_{-\infty}^{\infty} f_{XY}(x, y)\, dx \tag{1.11.16}$$

Equations (1.11.4) and (1.11.5) for discrete conditional probabilities can be applied to density functions for differential regions. This results in similar equations for conditional densities. The differential regions shown in Fig. 1.13 will be used in the following development. From the basic definition of conditional probability, we have

$$P(X \text{ is in strip } dx | Y \text{ is in strip } dy) = \frac{f_{XY}(x_0, y_0)\, dx\, dy}{f_Y(y_0)\, dy} \tag{1.11.17}$$

Canceling the dy's and noting the statement "Y is in strip dy" is approximately the same as saying "Y is equal to y_0" lead to

$$P(x_0 \le X \le x_0 + dx | Y = y_0) = \left[\frac{f_{XY}(x_0, y_0)}{f_Y(y_0)}\right] dx \tag{1.11.18}$$

Note that the quantity in brackets in Eq. (1.11.18) has all the earmarks of a density function, and its probabilistic interpretation is on the left side of the equation. Thus, we define conditional density as

$$f_{X|Y}(x) = \frac{f_{XY}(x, y)}{f_Y(y)} \tag{1.11.19}$$

The functional dependence on the second argument y has been intentionally omitted in $f_{X|Y}$. This is to emphasize that $f_{X|Y}$ is a density function on x, not y [see Eq. (1.11.18)]. Of course, y does appear in $f_{X|Y}$ as a parameter, which may be thought of as a given, deterministic quantity. It is not the primary argument of $f_{X|Y}$ though.

The analogous definition of $f_{Y|X}$ is

$$f_{Y|X}(y) = \frac{f_{XY}(x, y)}{f_X(x)} \tag{1.11.20}$$

Once Eqs. (1.11.19) and (1.11.20) have been established, Bayes rule follows directly.

Bayes rule:

$$f_{X|Y}(x) = \frac{f_{Y|X}(y) f_X(x)}{f_Y(y)} \tag{1.11.21}$$

Statistical independence for continuous random variables X and Y is defined in the same manner as for discrete variables; that is, X and Y are statistically independent if

$$f_{XY}(x, y) = f_X(x) f_Y(y) \tag{1.11.22}$$

An example is now in order.

EXAMPLE 1.19

Consider a dart-throwing game in which the target is a conventional xy coordinate system. The player aims each throw at the origin according to his or her best ability. Since there are many vagaries affecting each throw, we can expect a scatter in the hit pattern. Also, after some practice, the scatter should be unbiased, left-to-right and up-and-down. Let the coordinate of a hit be a bivariate random variable (X, Y). In this example we would not expect the x coordinate to affect y in any way; therefore statistical independence of X and Y is a reasonable assumption. Also, because of the central tendency in X and Y, the assumption of normal distribution would appear to be reasonable. Thus we assume the following unconditional densities:

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2} \tag{1.11.23}$$

$$f_Y(y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-y^2/2\sigma^2} \tag{1.11.24}$$

The joint density is then given by

$$f_{XY}(x, y) = \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2} \cdot \frac{1}{\sqrt{2\pi}\sigma} e^{-y^2/2\sigma^2} = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{1.11.25}$$

Equation (1.11.25) is the special case of a bivariate normal density function in which X and Y are independent, unbiased, and have equal variances. This is an important density function and it is sketched in Fig. 1.14. It is often described as a smooth "hill-shaped" function and, for the special case here, the hill is symmetric in every respect. Thus the equal-height contour shown in Fig. 1.14 is an exact circle. ■

The joint density function for a discrete bivariate random variable (X, Y) can be represented with impulsive-type functions, just as in the single-variate case. However, in the bivariate case, remember that volume (not area) represents

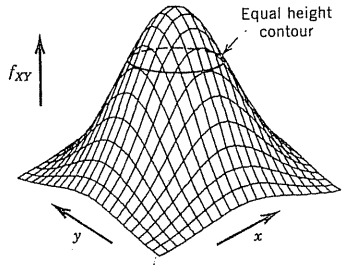


Figure 1.14 Bivariate normal density function.

probability. Thus volume-type impulse functions are appropriate. When X and Y are independent, the appropriate two-dimensional impulse function is obtained as a simple product of Dirac delta functions. For example, a unit volume impulse at the origin is given by $\delta(x)\delta(y)$. However, the situation is more complicated when X and Y are not independent, and we simply have to imagine appropriate volume-type impulsive functions. We will consider examples of this in subsequent chapters. The impulsive density function representation for the coin-drawing example, Example 1.18, is shown in Fig. 1.15.

1.12 CORRELATION, COVARIANCE, AND ORTHOGONALITY

The expectation of the product of two random variables X and Y is of special interest. In general, it is given by

$$E(XY) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} xyf_{XY}(x, y)\,dx\,dy \tag{1.12.1}$$

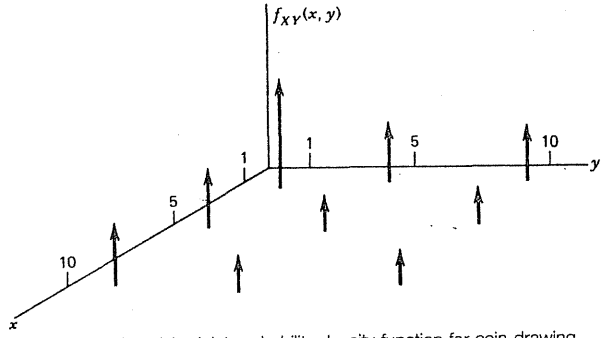


Figure 1.15 Impulsive joint probability density function for coin-drawing example, Example 1.18.

There is a special simplification of Eq. (1.12.1) that occurs when X and Y are independent. In this case, f_{XY} may be factored (see Eq. 1.11.22). Equation (1.12.1) then reduces to

$$E(XY) = \int_{-\infty}^{\infty} xf_X(x)\,dx \int_{-\infty}^{\infty} yf_Y(y)\,dy = E(X)E(Y) \tag{1.12.2}$$

If X and Y possess the property of Eq. (1.12.2), that is, the expectation of the product is the product of the individual expectations, they are said to be *uncorrelated.* Obviously, if X and Y are independent, they are also uncorrelated. However, the converse is not true, except in a few special cases (see Section 1.16).

As a matter of terminology, if

$$E(XY) = 0 \tag{1.12.3}$$

X and Y are said to be *orthogonal.*

The *covariance* of X and Y is also of special interest, and it is defined as

$$\text{Cov of } X \text{ and } Y = E[(X - m_X)(Y - m_Y)] \tag{1.12.4}$$

With the definition of Eq. (1.12.4) we can now define the *correlation coefficient* for two random variables as

$$\text{Correlation coefficient} = \rho = \frac{\text{Cov of } X \text{ and } Y}{\sqrt{\text{Var } X}\sqrt{\text{Var } Y}} = \frac{E[(X - m_X)(Y - m_Y)]}{\sqrt{\text{Var } X}\sqrt{\text{Var } Y}} \tag{1.12.5}$$

The correlation coefficient is a normalized measure of the degree of correlation between two random variables, and the normalization is such that ρ always lies within the range $-1 \le \rho \le 1$. This will be demonstrated (not proved) by looking at three special cases:

1. $Y = X$ (maximum positive correlation):
   When $Y = X$, Eq. (1.12.5) becomes

$$\rho = \frac{E[(X - m_X)(X - m_X)]}{\sqrt{E(X - m_X)^2}\sqrt{E(X - m_X)^2}} = 1$$

2. $Y = -X$ (maximum negative correlation):
   When $Y = -X$, Eq. (1.12.5) becomes

$$\rho = \frac{E[(X - m_X)(-X + m_X)]}{\sqrt{E(X - m_X)^2}\sqrt{E(-X + m_X)^2}} = -1$$

3. X and Y uncorrelated, that is, $E(XY) = E(X)E(Y)$:

Expanding the numerator of Eq. (1.12.5) yields

$$\begin{aligned} &E(XY - m_X Y - m_Y X + m_X m_Y) \\ &\quad = E(XY) - m_X E(Y) - m_Y E(X) + m_X m_Y \\ &\quad = m_X m_Y - m_X m_Y - m_Y m_X + m_X m_Y = 0 \end{aligned}$$

Thus, $\rho = 0$.

We have now examined the extremes of positive and negative correlation and zero correlation; there can be all shades of gray in between. [For further details, see Papoulis (7).]

1.13 SUM OF INDEPENDENT RANDOM VARIABLES AND TENDENCY TOWARD NORMAL DISTRIBUTION

Since we frequently need to consider additive combinations of independent random variables, this will now be examined in some detail. Let X and Y be independent random variables with probability density functions $f_X(x)$ and $f_Y(y)$. Define another random variable Z as the sum of X and Y:

$$Z = X + Y \tag{1.13.1}$$

Given the density functions of X and Y, we wish to find the corresponding density of Z.

Let z be a particular realization of the random variable Z, and think of z as being fixed. Now consider all possible realizations of X and Y that yield z. Clearly, they satisfy the equation

$$x + y = z \tag{1.13.2}$$

and the locus of points in the x, y plane is just a straight line, as shown in Fig. 1.16.

Next, consider an incremental perturbation of z to $z + dz$, and again consider the locus of realizations of X and Y that will yield $z + dz$. This locus is also a straight line, and it is shown as the upper line in Fig. 1.16. It should be apparent that all x and y lying within the differential strip between the two lines map into points between z and $z + dz$ in the z space. Therefore,

$$\begin{aligned} P(z \le Z \le z + dz) &= P(x \text{ and } y \text{ lie in differential strip}) \\ &= \iint_{\substack{\text{Diff.} \\ \text{strip}}} f_X(x) f_Y(y)\, dx\, dy \end{aligned} \tag{1.13.3}$$

But within the differential strip, y is constrained to x according to

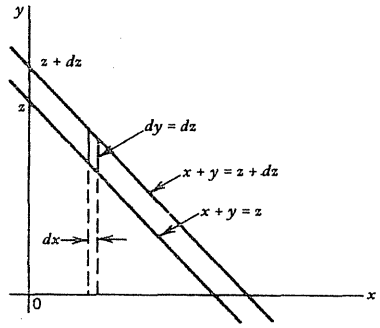


Figure 1.16 Differential strip used for deriving $f_z(z)$.

$$y = z - x \tag{1.13.4}$$

Also, since the strip is only of differential width, the double integral of Eq. (1.13.3) reduces to a single integral. Choosing x as the variable of integration and noting that $dy = dz$ lead to

$$P(z \le Z \le z + dz) = \left[\int_{-\infty}^{\infty} f_X(x) f_Y(z - x)\, dx \right] dz \tag{1.13.5}$$

It is now apparent from Eq. (1.13.5) that the quantity within the brackets is the desired probability density function for Z. Thus,

$$f_Z(z) = \int_{-\infty}^{\infty} f_X(x) f_Y(z - x)\, dx \tag{1.13.6}$$

It is of interest to note that the integral on the right side of Eq. (1.13.6) is a convolution integral. Thus, from Fourier transform theory, we can then write

$$\mathfrak{F}[f_Z] = \mathfrak{F}[f_X] \cdot \mathfrak{F}[f_Y] \tag{1.13.7}$$

where $\mathfrak{F}[\cdot]$ denotes "Fourier transform of $[\cdot]$." We now have two ways of evaluating the density of Z: (1) We can evaluate the convolution integral directly, or (2) we can first transform f_X and f_Y, then form the product of the transforms, and finally invert the product to get f_Z. Examples that illustrate each of these methods follow.

EXAMPLE 1.20

Let X and Y be independent random variables with identical rectangular density functions as shown in Fig. 1.17*a*. We wish to find the density function for their sum, which we will call Z.

Note first that the density shown in Fig. 1.17*a* has even symmetry. Thus $f_Y(z - x) = f_Y(x - z)$. The convolution integral expression of Eq. (1.13.6) is

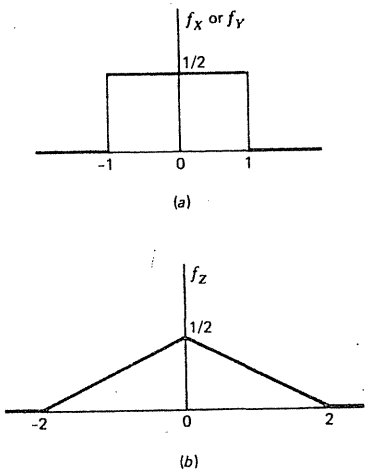


Figure 1.17 Probability density functions for X and Y and their sum. (*a*) Density function for both X and Y. (*b*) Density function for Z, where $Z = X + Y$.

then the integral of a rectangular pulse multiplied by a similar pulse shifted to the right amount z. When $z > 2$ or $z < 2$, there is no overlap in the pulses so their product is zero. When $-2 \leq z \leq 0$, there is a nontrivial overlap that increases linearly beginning at $z = -2$ and reaching a maximum at $z = 0$. The convolution integral then increases accordingly as shown in Fig. 1.17*b*. A similar argument may be used to show that $f_Z(z)$ decreases linearly in the interval where $0 \leq z \leq 2$. This leads to the triangular density function of Fig. 1.17*b*.

We can go one step further now and look at the density corresponding to the sum of three random variables. Let W be defined as

$$W = X + Y + V \tag{1.13.8}$$

where X, Y, and V are mutually independent and have identical rectangular densities as shown in Fig. 1.17*a*. We have already worked out the density for $X + Y$, so the density of W is the convolution of the two functions shown in Figs. 1.17*a* and 1.17*b*. We will leave the details of this as an exercise, and the result is shown in Fig. 1.18. Each of the segments labeled 1, 2, and 3 is an arc of a parabola. Notice the smooth central tendency. With a little imagination one can see a similarity between this and a zero-mean normal density curve. If we were to go another step and convolve a rectangular density with that of Fig. 1.18, we would get the density for the sum of four independent random variables. The resulting function would consist of connected segments of cubic functions extending from -4 to $+4$. Its appearance, though not shown, would resemble the normal curves even more than that of Fig. 1.18. And on and on—each additional convolution results in a curve that resembles the normal curve more closely than the preceding one.

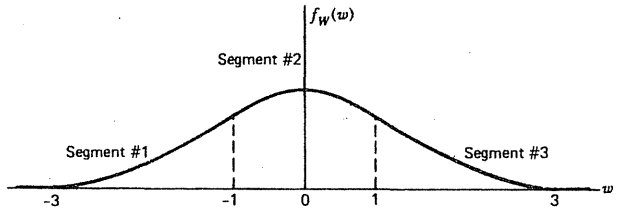


Figure 1.18 Probability density for the sum of three independent random variables with identical rectangular density functions.

This simple example is intended to demonstrate (not prove) that a superposition of independent random variables always tends toward normality, regardless of the distribution of the individual random variables contributing to the sum. This is known as the *central limit theorem* of statistics. It is a most remarkable theorem, and its validity is subject to only modest restrictions (3). In engineering applications the noise we must deal with is frequently due to a superposition of many small contributions. When this is so, we have good reason to make the assumption of normality. The central limit theorem says to do just that. Thus we have here one of the reasons for our seemingly exaggerated interest in normal random variables—they are a common occurrence in nature. ■

EXAMPLE 1.21

Let X and Y be independent normal random variables with zero means and variances σ_X^2 and σ_Y^2. We wish to find the probability density function for the sum of X and Y, which will again be denoted as Z. For variety, we illustrate the Fourier transform approach. The explicit expressions for f_X and f_Y are

$$f_X(t) = \frac{1}{\sqrt{2\pi}\sigma_X} e^{-t^2/2\sigma_X^2} \tag{1.13.9}$$

$$f_Y(t) = \frac{1}{\sqrt{2\pi}\sigma_Y} e^{-t^2/2\sigma_Y^2} \tag{1.13.10}$$

Note that we have used t as the dummy argument of the functions. It is of no consequence because it is integrated out in the transformation to the ω domain. Using Fourier transform tables, we find the transforms of f_X and f_Y to be

$$\mathcal{F}[f_x] = e^{-\sigma_X^2\omega^2/2} \tag{1.13.11}$$

$$\mathcal{F}[f_Y] = e^{-\sigma_Y^2\omega^2/2} \tag{1.13.12}$$

Forming their product yields

$$\mathcal{F}[f_X]\mathcal{F}[f_Y] = e^{-(\sigma_X^2+\sigma_Y^2)\omega^2/2} \tag{1.13.13}$$

Then the inverse gives the desired f_Z:

$$f_Z(z) = \mathcal{F}^{-1}\left[e^{-(\sigma_X^2+\sigma_Y^2)\omega^2/2}\right]$$

$$= \frac{1}{\sqrt{2\pi(\sigma_X^2+\sigma_Y^2)}} e^{-z^2/2(\sigma_X^2+\sigma_Y^2)} \tag{1.13.14}$$

Note that the density function for Z is also normal in form, and its variance is given by

$$\sigma_Z^2 = \sigma_X^2 + \sigma_Y^2 \tag{1.13.15}$$

The summation of any number of random variables can always be thought of as a sequence of summing operations on two variables; therefore, it should be clear that summing any number of independent normal random variables leads to a normal random variable. This rather remarkable result can be generalized further to include the case of dependent normal random variables, which we will discuss later. ■

1.14 TRANSFORMATION OF RANDOM VARIABLES

A mathematical transformation that takes one set of variables (say, inputs) into another set (say, outputs) is a common situation in systems analysis. Let us begin with a simple single-input, single-output situation where the input–output relationship is governed by the algebraic equation

$$y = g(x) \tag{1.14.1}$$

Here we are interested in random inputs, so think of x as a realization of the input random variable X, and y as the corresponding realization of the output Y. Assume we know the probability density function for X, and would like to find the corresponding density for Y. It is tempting to simply replace x in $f_X(x)$ with its equivalent in terms of y and pass it off at that. However, it is not quite that simple, as will be seen presently.

First, let us assume that the transformation $g(x)$ is one-to-one for all permissible x. By this we mean that the functional relationship given by Eq. (1.14.1) can be reversed, and x can be written uniquely as a function of y. Let the "reverse" relationship be

$$x = h(y) \tag{1.14.2}$$

The probabilities that X and Y lie within corresponding differential regions must be equal. That is,

$$P(X \text{ is between } x \text{ and } x + dx) = P(Y \text{ is between } y \text{ and } y + dy) \tag{1.14.3}$$

or

$$\int_x^{x+dx} f_X(u)\,du = \begin{cases} \displaystyle\int_y^{y+dy} f_Y(u)\,du, & \text{for } dy \text{ positive} \\ \displaystyle -\int_y^{y+dy} f_Y(u)\,du, & \text{for } dy \text{ negative} \end{cases} \tag{1.14.4}$$

One of the subtleties of this problem should now be apparent from Eq. (1.14.4). If positive dx yields negative dy (i.e., a negative derivative), the integral of f_Y must be taken from $y + dy$ to y in order to yield a positive probability. This is the equivalent of interchanging the limits and reversing the sign as shown in Eq. (1.14.4).

The differential equivalent of Eq. (1.14.4) is

$$f_X(x)\,dx = f_Y(y)|dy| \tag{1.14.5}$$

where we have tacitly assumed dx to be positive. Also, x is constrained to be $h(y)$. Thus, we have

$$f_Y(y) = \left|\frac{dx}{dy}\right| f_X(h(y)) \tag{1.14.6}$$

or, equivalently,

$$f_Y(y) = |h'(y)| f_X(h(y)) \tag{1.14.7}$$

where $h'(y)$ indicates the derivative of h with respect to y. Two examples will now be presented.

EXAMPLE 1.22

Find the appropriate output density functions for the case where the input X is $N(0, \sigma_X^2)$ and the transformation is

(a) $y = Kx$ (K is a given constant) (1.14.8)

(b) $y = x^3$ (1.14.9)

(c) $y = x^2$ (1.14.10)

We begin with the scale-factor transformation indicated by Eq. (1.14.8). We first solve for x in terms of y and then form the derivative. Thus,

$$x = \frac{1}{K} y \tag{1.14.11}$$

$$\left|\frac{dx}{dy}\right| = \left|\frac{1}{K}\right| \tag{1.14.12}$$

We can now obtain the equation for f_Y from Eq. (1.14.6). The result is

$$f_Y(y) = \frac{1}{|K|} \cdot \frac{1}{\sqrt{2\pi}\sigma_X} \exp\left[-\frac{\left(\frac{y}{K}\right)^2}{2\sigma_X^2} \right] \tag{1.14.13}$$

Or rewriting Eq. (1.14.13) in standard normal form yields

$$f_Y(y) = \frac{1}{\sqrt{2\pi(K\sigma_X)^2}} \exp\left[-\frac{y^2}{2(K\sigma_X)^2} \right] \tag{1.14.14}$$

It can now be seen that transforming a zero-mean normal random variable with a simple scale factor yields another normal random variable with a corresponding scale change in its standard deviation. It is important to note that normality is preserved in a linear transformation.

Next, consider part (b). This transformation is also one-to-one, so solving for x yields

$$x = \sqrt[3]{y} \tag{1.14.15}$$

In Eq. (1.14.15) we take the positive real root for $y > 0$ and the negative real root for $y < 0$. The derivative of x is then

$$\frac{dx}{dy} = \frac{1}{3} y^{-2/3} \tag{1.14.16}$$

The quantity $y^{2/3}$ can be written as $(y^{1/3})^2$, so $y^{2/3}$ is always positive, provided $y^{1/3}$ is real. This is consistent with the geometric interpretation of $y = x^3$, which always has a nonnegative slope. The density function for Y is then

$$f_Y(y) = \frac{1}{3y^{2/3}} \cdot \frac{1}{\sqrt{2\pi}\sigma_X} e^{-(y^{1/3})^2/2\sigma_X^2} \tag{1.14.17}$$

Since this does not reduce to normal form, we have here an example of a nonlinear transformation that converts a normal random variable to non-Gaussian form.

The transformation $y = x^2$ for part (c) is sketched in Fig. 1.19. It is obvious from the sketch that two values of x yield the same y. Thus, the function is not one-to-one, which violates one of the assumptions in deriving Eq. (1.14.6). The problem is solvable, though; we simply must go back to fundamentals and derive a new $f_Y(y)$ relationship. Note that we can consider $y = x^2$ in terms of two branches. These will be defined as

$$x = \begin{cases} \sqrt{y}, & x \geq 0 \quad \text{(Branch 1)} \\ -\sqrt{y}, & x < 0 \quad \text{(Branch 2)} \end{cases} \tag{1.14.18}$$

Now think of perturbing y a positive differential amount dy as shown in Fig. 1.19. This results in two corresponding differential regions on the x axis. Thus, we have the following relationship for probabilities:

$$P(y_0 \leq Y \leq y_0 + dy) = P(x_0 \leq X \leq x_0 + dx_1) + P(-x_0 + dx_2 \leq X \leq -x_0) \tag{1.14.19}$$

Note that dx_1 (Branch 1) is positive and dx_2 (Branch 2) is negative. Next, Eq. (1.14.19) can be rewritten in terms of integrals as

$$\int_{y_0}^{y_0+dy} f_Y(u)\, du = \int_{x_0}^{x_0+dx_1} f_X(v)\, dv + \int_{-x_0+dx_2}^{-x_0} f_X(v)\, dv \tag{1.14.20}$$

We now note that

$$dx_1 = \frac{1}{2\sqrt{y_0}}\, dy$$

$$dx_2 = -\frac{1}{2\sqrt{y_0}}\, dy \tag{1.14.21}$$

Substituting Eq. (1.14.21) into Eq. (1.14.20) and letting the range of integration be incremental yield

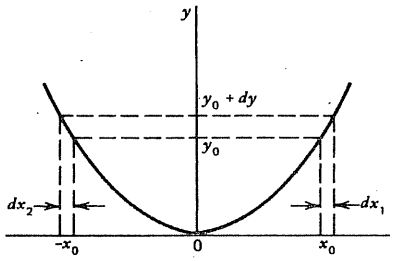


Figure 1.19 Sketch of $y = x^2$.

$$f_Y(y_0)\,dy = f_X(\sqrt{y_0}) \cdot \frac{1}{2\sqrt{y_0}}\,dy + f_X(\sqrt{y_0}) \cdot \frac{1}{2\sqrt{y_0}}\,dy, \qquad y_0 \geq 0 \tag{1.14.22}$$

Now, since y_0 can be any y greater than zero, Eq. (1.14.22) reduces to

$$f_Y(y) = \frac{1}{\sqrt{y}} f_X(\sqrt{y}), \qquad y \geq 0 \tag{1.14.23}$$

From the equation $y = x^2$, we see that no real values of x map into negative y. Therefore, $f_Y(y) = 0$ for negative y. The final f_Y is then

$$f_Y(y) = \begin{cases} \dfrac{1}{\sqrt{y}} f_X(\sqrt{y}), & y \geq 0 \\ 0, & y < 0 \end{cases} \tag{1.14.24}$$

We can now apply Eq. (1.14.24) to the case where X is normal with zero mean. This yields

$$f_Y(y) = \begin{cases} \dfrac{1}{\sqrt{2\pi}\sigma_X \cdot \sqrt{y}}\, e^{-y/2\sigma_X^2}, & y \geq 0 \\ 0, & y < 0 \end{cases} \tag{1.14.25}$$

Note that Y is not a normal random variable. Rather, it is a special case of what is known as a chi-square random variable. (See Problem 1.42 for a more general discussion of the chi-square distribution.)

We also have occasion to deal with transformations of multiple random variables. Consider a bivariate transformation example. Since the extension to higher-order transformations is fairly obvious, only the bivariate case will be discussed in detail.

EXAMPLE 1.23

In the target-shooting example (Example 1.19), the hit location was described in terms of rectangular coordinates. This led to the joint probability density function

$$f_{XY}(x, y) = \frac{1}{2\pi\sigma^2}\, e^{-(x^2+y^2)/2\sigma^2} \tag{1.14.26}$$

This is a special case of the bivariate normal density function where the two variates are independent and have zero means and equal variances. Suppose we wish to find the corresponding density in terms of polar coordinates r and θ. Formally, then, we wish to define new random variables R and Θ such that pairwise realizations (x, y) transform to (r, θ) in accordance with

$$r = \sqrt{x^2 + y^2}, \qquad r \geq 0$$
$$\theta = \tan^{-1}\frac{y}{x}, \qquad 0 \leq \theta < 2\pi \tag{1.14.27}$$

Or, equivalently,

$$x = r\cos\theta$$
$$y = r\sin\theta \tag{1.14.28}$$

We wish to find $f_{R\Theta}(r, \theta)$ and the unconditional density functions $f_R(r)$ and $f_\Theta(\theta)$.

The probability that a hit will lie within an area bounded by a closed contour C is given by

$$P(\text{Hit lies within } C) = \iint_{\substack{\text{Area} \\ \text{enclosed} \\ \text{by } C}} f_{XY}(x, y)\,dx\,dy \tag{1.14.29}$$

We know from multivariable calculus that the double integral in Eq. (1.14.29) can also be evaluated in the r, θ coordinate frame as

$$\iint_{\substack{\text{Region} \\ \text{enclosed} \\ \text{by } C}} f_{XY}(x, y)\,dx\,dy = \iint_{\substack{\text{Region} \\ \text{enclosed} \\ \text{by } C'}} f_{XY}(x(r, \theta), y(r, \theta))\left|J\left(\frac{x, y}{r, \theta}\right)\right| dr\,d\theta \tag{1.14.30}$$

where C' is the contour in the r, θ plane corresponding to C in the x, y plane. That is, points within C map into points within C'. (Note that it is immaterial as to how we draw the "picture" in the r, θ coordinate frame. For example, if we choose to think of r and θ as just another set of Cartesian coordinates for plotting purposes, C' becomes a distortion of C.) The J quantity in Eq. (1.14.30) is the Jacobian of the transformation, defined as

$$J\left(\frac{x, y}{r, \theta}\right) = \text{Det}\begin{bmatrix} \dfrac{\partial x}{\partial r} & \dfrac{\partial y}{\partial r} \\ \dfrac{\partial x}{\partial \theta} & \dfrac{\partial y}{\partial \theta} \end{bmatrix} \tag{1.14.31}$$

The vertical bars around J in Eq. (1.14.30) indicate absolute magnitude. We can argue now that if Eq. (1.14.30) is true for regions in general, it must also be true for differential regions. Let the differential region in the r, θ domain be bounded by r and $r + dr$ in one direction and by θ and $\theta + d\theta$ in the other. If it helps, think of plotting r and θ as Cartesian coordinates. The differential region

in the r, θ domain is then rectangular, and the corresponding one in the x, y domain is a curvilinear differential rectangle (see Fig. 1.20). Now, by the very definition of joint density, the quantity multiplying $dr\,d\theta$ in Eq. (1.14.30) is seen to be the desired density function. That is,

$$f_{R\Theta}(r, \theta) = f_{XY}[x(r, \theta), y(r, \theta)]\left|J\left(\frac{x, y}{r, \theta}\right)\right| \tag{1.14.32}$$

In the transformation of this example,

$$J\left(\frac{x, y}{r, \theta}\right) = \text{Det}\begin{bmatrix} \cos\theta & \sin\theta \\ -r\sin\theta & r\cos\theta \end{bmatrix} = r \tag{1.14.33}$$

Since the radial coordinate r is always positive, $|J| = r$. We can now substitute Eqs. (1.14.26) and (1.14.33) into (1.14.32) and obtain

$$\begin{aligned} f_{R\Theta}(r, \theta) &= r\frac{1}{2\pi\sigma^2}\exp\left[-\frac{(r\cos\theta)^2 + (r\sin\theta)^2}{2\sigma^2}\right] \\ &= \frac{r}{2\pi\sigma^2}e^{-r^2/2\sigma^2} \end{aligned} \tag{1.14.34}$$

Note that the density function of this example has no explicit dependence on θ. In other words, all angles between 0 and 2π are equally likely, which is what we would expect in the target-throwing experiment.

We get the unconditional density functions by integrating $f_{R\Theta}$ with respect to the appropriate variables. That is,

$$\begin{aligned} f_R(r) &= \int_0^{2\pi} f_{R\Theta}(r, \theta)\,d\theta = \frac{r}{2\pi\sigma^2}e^{-r^2/2\sigma^2}\int_0^{2\pi} d\theta \\ &= \frac{r}{\sigma^2}e^{-r^2/2\sigma^2} \end{aligned} \tag{1.14.35}$$

and

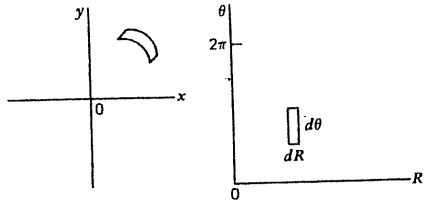


Figure 1.20 Corresponding differential regions for transformation of Example 1.23.

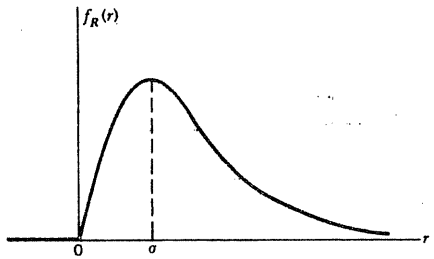


Figure 1.21 Rayleigh density function.

$$f_\Theta(\theta) = \int_0^\infty f_{R\Theta}(r, \theta)\,dr = \begin{cases} \dfrac{1}{2\pi}, & 0 \le \theta < 2\pi \\ 0, & \text{otherwise} \end{cases} \tag{1.14.36}$$

The single-variate density function given by Eq. (1.14.35) is called the *Rayleigh* density function. It is of considerable importance in applied probability, and it is sketched in Fig. 1.21. It is easily verified that the mode (peak value) of the Rayleigh density is equal to standard deviation of the x and y normal random variables from which it was derived. Thus, we see that similar independent, zero-mean normal densities in the x, y domain correspond to Rayleigh and uniform densities in the r, θ domain. ■

1.15 MULTIVARIATE NORMAL DENSITY FUNCTION

In Sections 1.9 and 1.11 examples of the single and bivariate normal density functions were presented. We work so much with normal random variables that we need to elaborate further and develop a general form for the n-dimensional normal density function. One can write out the explicit equations for the single and bivariate cases without an undue amount of "clutter" in the equations. However, beyond this, matrix notation is a virtual necessity. Otherwise, the explicit expressions are completely unwieldy.

Consider a set of n random variables $X_1, X_2, \ldots, X_n$ (also called variates). We define a vector random variable $\mathbf{X}$ as*

* Note that uppercase $\mathbf{X}$ denotes a column vector in this section. This is a departure from the usual notation of matrix theory, but it is necessitated by a desire to be consistent with the previous uppercase notation for random variables. The reader will simply have to remember this minor deviation in matrix notation. It appears in this section and also occasionally in Chapter 2.

$$\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_n \end{bmatrix} \tag{1.15.1}$$

In general, the components of $\mathbf{X}$ may be correlated and have nonzero means. We denote the respective means as $m_1, m_2, \ldots, m_n$, and thus we define a mean vector $\mathbf{m}$ as

$$\mathbf{m} = \begin{bmatrix} m_1 \\ m_2 \\ \vdots \\ m_n \end{bmatrix} \tag{1.15.2}$$

Also, if $x_1, x_2, \ldots, x_n$ is a set of realizations of $X_1, X_2, \ldots, X_n$, we can define a vector realization of $\mathbf{X}$ as

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \tag{1.15.3}$$

We next define a matrix that describes the variances and correlation structure of the n variates. The *covariance matrix* for $\mathbf{X}$ is defined as

$$\mathbf{C} = \begin{bmatrix} E[(X_1 - m_1)^2] & E[(X_1 - m_1)(X_2 - m_2)] & \cdots \\ E[(X_2 - m_2)(X_1 - m_1)] & & \\ \vdots & \cdots & E[(X_n - m_n)^2] \end{bmatrix} \tag{1.15.4}$$

The terms along the major diagonal of $\mathbf{C}$ are seen to be the variances of the variates, and the off-diagonal terms are the covariances.

The random variables $X_1, X_2, \ldots, X_n$ are said to be *jointly normal* or *jointly Gaussian* if their joint probability density function is given by

$$f_X(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}|\mathbf{C}|^{1/2}} \exp\left\{-\frac{1}{2}[(\mathbf{x} - \mathbf{m})^T\mathbf{C}^{-1}(\mathbf{x} - \mathbf{m})]\right\} \tag{1.15.5}$$

where $\mathbf{x}$, $\mathbf{m}$, and $\mathbf{C}$ are defined by Eqs. (1.15.2) to (1.15.4) and $|\mathbf{C}|$ is the determinant of $\mathbf{C}$. "Super -1" and "super T" denote matrix inverse and transpose, respectively. Note that the defining function for f_X is scalar and is a function of $x, x_2, \ldots, x_n$ when written out explicitly. We have shortened the indicated

functional dependence to $\mathbf{x}$ just for compactness in notation. Also note that $\mathbf{C}^{-1}$ must exist in order for f_X to be properly defined by Eq. (1.15.5). Thus, $\mathbf{C}$ must be nonsingular. More will be said of this later.

Clearly, Eq. (1.15.5) reduces to the standard normal form for the single variate case. For the bivariate case, we may write out f_X explicitly in terms of x_1 and x_2 without undue difficulty. Proceeding to do this, we have

$$\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}, \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}, \qquad \mathbf{m} = \begin{bmatrix} m_1 \\ m_2 \end{bmatrix} \tag{1.15.6}$$

and

$$\mathbf{C} = \begin{bmatrix} E[(X_1 - m_1)^2] & E[(X_1 - m_1)(X_2 - m_2)] \\ E[(X_1 - m_1)(X_2 - m_2)] & E[(X_2 - m_2)^2] \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix} \tag{1.15.7}$$

The second form for $\mathbf{C}$ in Eq. (1.15.7) follows directly from the definitions of variance and correlation coefficient. The determinant of $\mathbf{C}$ and its inverse are given by

$$|\mathbf{C}| = \begin{vmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{vmatrix} = (1 - \rho^2)\sigma_1^2\sigma_2^2 \tag{1.15.8}$$

$$\mathbf{C}^{-1} = \begin{bmatrix} \dfrac{\sigma_2^2}{|\mathbf{C}|} & -\dfrac{\rho\sigma_1\sigma_2}{|\mathbf{C}|} \\ -\dfrac{\rho\sigma_1\sigma_2}{|\mathbf{C}|} & \dfrac{\sigma_1^2}{|\mathbf{C}|} \end{bmatrix} = \begin{bmatrix} \dfrac{1}{(1 - \rho^2)\sigma_1^2} & \dfrac{-\rho}{(1 - \rho^2)\sigma_1\sigma_2} \\ \dfrac{-\rho}{(1 - \rho^2)\sigma_1\sigma_2} & \dfrac{1}{(1 - \rho^2)\sigma_2^2} \end{bmatrix} \tag{1.15.9}$$

Substituting Eqs. (1.15.8) and (1.15.9) into Eq. (1.15.5) then yields the desired density function in terms of x_1 and x_2.

$$f_{X_1X_2}(x_1, x_2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1 - \rho^2}} \exp\left\{-\frac{1}{2(1 - \rho^2)}\left[\frac{(x_1 - m_1)^2}{\sigma_1^2} - \frac{2\rho(x_1 - m_1)(x_2 - m_2)}{\sigma_1\sigma_2} + \frac{(x_2 - m_2)^2}{\sigma_2^2}\right]\right\} \tag{1.15.10}$$

It should be clear in Eq. (1.15.10) that $f_{X_1X_2}(x_1, x_2)$ is intended to mean the same as $f_X(\mathbf{x})$ in vector notation.

As mentioned previously, the third- and higher-order densities are very cumbersome to write out explicitly; therefore, we will examine the bivariate density in some detail in order to gain insight into the general multivariate normal density function. The bivariate normal density function is a smooth hill-shaped surface over the x_1, x_2 plane. This was sketched previously in Fig. 1.14

for the special case where $\sigma_1 = \sigma_2$ and $\rho = 0$. In the more general case, a constant probability density contour projects into the x_1, x_2 plane as an ellipse with its center at (m_1, m_2) as shown in Fig. 1.22. The orientation of the ellipse in Fig. 1.22 corresponds to a positive correlation coefficient. Points on the ellipse may be thought of as equally likely combinations of x_1 and x_2. If $\rho = 0$, we have the case where X_1 and X_2 are uncorrelated, and the ellipses have their semimajor and semiminor axes parallel to the x_1 and x_2 axes. If we specialize further and let $\sigma_1 = \sigma_2$ (and $\rho = 0$), the ellipses degenerate to circles. In the other extreme, as $|\rho|$ approaches unity, the ellipses become more and more eccentric.

The uncorrelated case where $\rho = 0$ is of special interest, and in this case $f_{X_1X_2}$ reduces to the form given in Eq. (1.15.11).

For uncorrelated X_1 and X_2

$$f_{X_1X_2}(x_1, x_2) = \frac{1}{2\pi\sigma_1\sigma_2} \exp\left\{-\frac{1}{2}\left[\frac{(x_1 - m_1)^2}{\sigma_1^2} + \frac{(x_2 - m_2)^2}{\sigma_2^2}\right]\right\}$$

$$= \frac{1}{\sqrt{2\pi}\sigma_1} e^{-(x_1-m_1)^2/2\sigma_1^2} \cdot \frac{1}{\sqrt{2\pi}\sigma_2} e^{-(x_2-m_2)^2/2\sigma_2^2} \qquad (1.15.11)$$

The joint density function is seen to factor into the product of $f_{X_1}(x_1)$ and $f_{X_2}(x_2)$. Thus, *two normal random variables that are uncorrelated are also statistically independent.* It is easily verified from Eq. (1.15.5) that this is also true for any number of uncorrelated normal random variables. This is exceptional, because in general zero correlation does not imply statistical independence. It does, however, in the Gaussian case.

Now try to visualize the three-variate normal density function. The locus of constant $f_{X_1X_2X_3}(x_1, x_2, x_3)$ will be a closed elliptically shaped surface with three axes of symmetry. These axes will be aligned with the x_1, x_2, x_3 axes for the

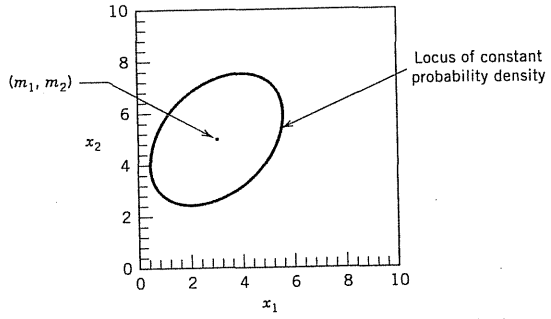


Figure 1.22 Contour projection of a bivariate normal density function to the (x_1, x_2) plane.

case of zero correlation among the three variates. If, in addition to zero correlation, the variates have equal variances, the surface becomes spherical.

If we try to extend the geometric picture beyond the three-variate case, we run out of Euclidean dimensions. However, conceptually, we can still envision equal-likelihood surfaces in hyperspace, but there is no way of sketching a picture of such surfaces. Some general properties of multivariate normal random variables will be explored further in the next section.

1.16 LINEAR TRANSFORMATION AND GENERAL PROPERTIES OF NORMAL RANDOM VARIABLES

The general linear transformation of one set of normal random variables to another is of special interest in noise analysis. This will now be examined in detail.

We have just seen that the density function for jointly normal random variables $X_1, X_2, \ldots, X_n$ can be written in matrix form as

$$f_X(\mathbf{x}) = \frac{1}{(2\pi)^{n/2}|\mathbf{C}_X|^{1/2}} \exp\left\{-\frac{1}{2}[(\mathbf{x} - \mathbf{m}_X)^T\mathbf{C}_X^{-1}(\mathbf{x} - \mathbf{m}_X)]\right\} \qquad (1.16.1)$$

We have added the subscript X to $\mathbf{m}$ and $\mathbf{C}$ to indicate that these are the mean and covariance matrices associated with the $\mathbf{X}$ random variable. We now define a new set of random variables $Y_1, Y_2, \ldots, Y_n$ that are linearly related to $X_1, X_2, \ldots, X_n$ via the equation

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b} \qquad (1.16.2)$$

where lowercase $\mathbf{x}$ and $\mathbf{y}$ indicate realizations of $\mathbf{X}$ and $\mathbf{Y}$, $\mathbf{b}$ is a constant vector, and $\mathbf{A}$ is a square matrix that will be assumed to be nonsingular (i.e., invertible). We wish to find the density function for $\mathbf{Y}$, and we can use the methods of Section 1.14 to do so. In particular, the transformation is one-to-one; therefore, a generalized version of Eq. (1.14.32) may be used.

$$f_Y(\mathbf{y}) = f_X(\mathbf{x}(\mathbf{y}))\left|J\left(\frac{\mathbf{x}}{\mathbf{y}}\right)\right| \qquad (1.16.3)$$

We must first solve Eq. (1.16.2) for $\mathbf{x}$ in terms of $\mathbf{y}$. The result is

$$\mathbf{x} = \mathbf{A}^{-1}\mathbf{y} - \mathbf{A}^{-1}\mathbf{b} \qquad (1.16.4)$$

Let the individual terms of $\mathbf{A}^{-1}$ be denoted as

$$\mathbf{A}^{-1} = \begin{bmatrix} d_{11} & d_{12} & \cdots & d_{1n} \\ d_{21} & d_{22} & \cdots & \\ \cdot & \cdot & \cdots & \\ d_{n1} & \cdot & \cdots & d_{nn} \end{bmatrix} \tag{1.16.5}$$

The scalar equations represented by Eq. (1.16.4) are then

$$\begin{aligned} x_1 &= (d_{11}y_1 + d_{12}y_2 + \cdots) - (d_{11}b_1 + d_{12}b_2 + \cdots) \\ x_2 &= (d_{21}y_1 + d_{22}y_2 + \cdots) - (d_{21}b_1 + d_{22}b_2 + \cdots) \\ x_3 &= \cdots \text{ etc.} \end{aligned} \tag{1.16.6}$$

The Jacobian of the transformation is then

$$\begin{aligned} J\left(\frac{\mathbf{x}}{\mathbf{y}}\right) = J\left(\frac{x_1, x_2, \ldots}{y_1, y_2, \ldots}\right) &= \text{Det}\begin{bmatrix} \dfrac{\partial x_1}{\partial y_1} & \dfrac{\partial x_2}{\partial y_1} & \cdots \\ \dfrac{\partial x_1}{\partial y_2} & \dfrac{\partial x_2}{\partial y_2} & \cdots \\ \cdots\cdots\cdots\cdots\cdots & & \end{bmatrix} \\ &= \text{Det}\begin{bmatrix} d_{11} & d_{21} & \cdots \\ d_{12} & d_{22} & \cdots \\ \cdots\cdots\cdots\cdots & & \end{bmatrix} = \text{Det}(\mathbf{A}^{-1})^T = \text{Det}(\mathbf{A}^{-1}) \end{aligned} \tag{1.16.7}$$

We can now substitute Eqs. (1.16.4) and (1.16.7) into Eq. (1.16.3). The result is

$$\begin{aligned} f_Y(\mathbf{y}) = &\frac{|\text{Det}(\mathbf{A}^{-1})|}{(2\pi)^{n/2}|\mathbf{C}_X|^{1/2}} \\ &\times \exp\left\{-\frac{1}{2}\left[(\mathbf{A}^{-1}\mathbf{y} - \mathbf{A}^{-1}\mathbf{b} - \mathbf{m}_X)^T\mathbf{C}_X^{-1}(\mathbf{A}^{-1}\mathbf{y} - \mathbf{A}^{-1}\mathbf{b} - \mathbf{m}_X)\right]\right\} \end{aligned} \tag{1.16.8}$$

We find the mean of $\mathbf{Y}$ by taking the expectation of both sides of the linear transformation

$$\mathbf{Y} = \mathbf{AX} + \mathbf{b}$$

Thus,

$$\mathbf{m}_Y = \mathbf{Am}_X + \mathbf{b} \tag{1.16.9}$$

The exponent in Eq. (1.16.8) may now be written as

$$\begin{aligned} &-\frac{1}{2}\left[(\mathbf{A}^{-1}\mathbf{y} - \mathbf{A}^{-1}\mathbf{b} - \mathbf{A}^{-1}\mathbf{Am}_X)^T\mathbf{C}_X^{-1}(\mathbf{A}^{-1}\mathbf{y} - \mathbf{A}^{-1}\mathbf{b} - \mathbf{A}^{-1}\mathbf{Am}_X)\right] \\ &= -\frac{1}{2}\left[(\mathbf{y} - \mathbf{m}_Y)^T(\mathbf{A}^{-1})^T\mathbf{C}_X^{-1}\mathbf{A}^{-1}(\mathbf{y} - \mathbf{m}_Y)\right] \\ &= -\frac{1}{2}\left[(\mathbf{y} - \mathbf{m}_Y)^T(\mathbf{AC}_X\mathbf{A}^T)^{-1}(\mathbf{y} - \mathbf{m}_Y)\right] \end{aligned} \tag{1.16.10}$$

The last step in Eq. (1.16.10) is accomplished by using the reversal rule for the inverse of triple products and noting that the order of the transpose and inverse operations may be interchanged. Also note that

$$|\text{Det}(\mathbf{A}^{-1})| = \frac{1}{|\text{Det }\mathbf{A}|} = \frac{1}{|\text{Det }\mathbf{A}|^{1/2} \cdot |\text{Det }\mathbf{A}^T|^{1/2}} \tag{1.16.11}$$

Substitution of the forms given in Eqs. (1.16.10) and (1.16.11) into Eq. (1.16.8) yields for f_Y

$$\begin{aligned} f_Y(\mathbf{y}) = &\frac{1}{(2\pi)^{n/2}|\mathbf{AC}_X\mathbf{A}^T|^{1/2}} \\ &\times \exp\left\{-\frac{1}{2}\left[(\mathbf{y} - \mathbf{m}_Y)^T(\mathbf{AC}_X\mathbf{A}^T)^{-1}(\mathbf{y} - \mathbf{m}_Y)\right]\right\} \end{aligned} \tag{1.16.12}$$

It is apparent now that f_Y is also normal in form with the mean and covariance matrix given by

$$\mathbf{m}_Y = \mathbf{Am}_X + \mathbf{b} \tag{1.16.13}$$

and

$$\mathbf{C}_Y = \mathbf{AC}_X\mathbf{A}^T \tag{1.16.14}$$

Thus, we see that *normality is preserved in a linear transformation.* All that is changed is the mean and the covariance matrix; the *form* of the density function remains unchanged.

There are, of course, an infinite number of linear transformations one can make on a set of normal random variables. Any transformation, say, $\mathbf{S}$, that produces a new covariance matrix $\mathbf{SC}_X\mathbf{S}^T$ that is diagonal is of special interest. Such a transformation will yield a new set of normal random variables that are uncorrelated, and thus they are also statistically independent. In a given problem, we may not choose to actually make this change of variables, but it is important just to know that the variables can be decoupled and under what circumstances this can be done. It works out that a diagonalizing transformation will always

exist if $\mathbf{C}_X$ is positive definite (8).* In the case of a covariance matrix, this implies that all the correlation coefficients are less than unity in magnitude. This will be demonstrated for the bivariate case, and the extension to higher-order cases is fairly obvious.

A symmetric matrix $\mathbf{C}$ is said to be positive definite if the scalar $\mathbf{x}^T\mathbf{C}\mathbf{x}$ is positive for all nontrivial $\mathbf{x}$, that is, $\mathbf{x} \neq 0$. Writing out $\mathbf{x}^T\mathbf{C}\mathbf{x}$ explicitly for the 2×2 case yields

$$[x_1 x_2]\begin{bmatrix} c_{11} & c_{12} \\ c_{12} & c_{22} \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = c_{11}x_1^2 + 2c_{12}x_1x_2 + c_{22}x_2^2 \tag{1.16.15}$$

But if $\mathbf{C}$ is a covariance matrix,

$$c_{11} = \sigma_1^2, \qquad c_{12} = \rho\sigma_1\sigma_2, \qquad c_{22} = \sigma_2^2 \tag{1.16.16}$$

Therefore, $\mathbf{x}^T\mathbf{C}\mathbf{x}$ is

$$\mathbf{x}^T\mathbf{C}\mathbf{x} = (\sigma_1 x_1)^2 + 2\rho(\sigma_1 x_1)(\sigma_2 x_2) + (\sigma_2 x_2)^2 \tag{1.16.17}$$

Equation (1.16.17) now has a simple geometric interpretation. Assume $|\rho| < 1$; ρ can be related to the negative cosine of some angle θ, where $0 < \theta < \pi$. Equation (1.16.17) will then be recognized as the equation for the square of the "opposite side" of a general triangle; and this, of course, must be positive. Thus, a 2×2 covariance matrix is positive definite, provided $|\rho| < 1$.

It is appropriate now to summarize some of the important properties of multivariate normal random variables:

1. The probability density function describing a vector random variable $\mathbf{X}$ is completely defined by specifying the mean and covariance matrix of $\mathbf{X}$.
2. The covariance matrix of $\mathbf{X}$ is positive definite. The magnitudes of all correlation coefficients are less than unity.
3. If normal random variables are uncorrelated, they are also statistically independent.
4. A linear transformation of normal random variables leads to another set of normal random variables. A decoupling (decorrelating) transformation will always exist if the original covariance matrix is positive definite.
5. If the joint density function for n random variables is normal in form, all marginal and conditional densities associated with the n variates will also be normal in form. (This was not shown; see Problem 1.36.)

* MATLAB's Cholesky factorization function is helpful in determining a diagonalizing transformation. This is discussed in detail in Section 5.4 on Monte Carlo simulation.

1.17 LIMITS, CONVERGENCE, AND UNBIASED ESTIMATORS

No discussion of probability could be complete without at least some mention of limits and convergence. To put this in perspective, we first review the usual deterministic concept of convergence. As an example, recall that the Maclaurin series for e^x is

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots \tag{1.17.1}$$

This series converges uniformly to e^x for all real x in any finite interval. By convergence we mean that if a given accuracy figure is specified, we can find an appropriate number of terms such that the specified accuracy is met by a truncated version of the series. In particular, note that once we have determined how many terms are needed in the truncated series, this same number is good for all x within the interval, and there is nothing "chancy" about it. In contrast, we will see presently that such "100 percent sure" statements cannot be made in probabilistic situations. A look at the sample mean of n random variables will serve to illustrate this.

Let $X_1, X_2, \ldots, X_n$ be independent random variables with identical probability density functions $f_X(x)$. In terms of an experiment, these may be thought of as ordered samples of the random variable X. Next, consider a sequence of random variables defined as follows:

$$\begin{aligned} Y_1 &= X_1 \\ Y_2 &= \frac{X_1 + X_2}{2} \\ Y_3 &= \frac{X_1 + X_2 + X_3}{3} \\ &\vdots \\ Y_n &= \frac{X_1 + X_2 + \cdots X_n}{n} \end{aligned} \tag{1.17.2}$$

The random variable Y_n is, of course, just the sample mean of the random variable X. We certainly expect Y_n to get closer to $E(X)$ as n becomes large. But closer in what sense? This is the crucial question. It should be clear that any particular experiment could produce an "unusual" event in which the sample mean would differ from $E(X)$ considerably. On the other hand, quite by chance, a similar experiment might yield a sample mean that was quite close to $E(X)$. Thus, in this probabilistic situation, we cannot expect to find a fixed number of

samples n that will meet a specified accuracy figure for all experiments. No matter how large we make n, there is always some nonzero probability that the very unusual thing will happen, and a particular experiment will yield a sample mean that is outside the specified accuracy. Thus, we can only hope for convergence in some sort of average sense and not in an absolute (100 percent sure) sense.

Let us now be more specific in this example, and let X (and thus X_1, $X_2, \ldots, X_n$) be normal with mean m_X and variance σ_X^2. From Section 1.16 we also know that the sample mean Y_n is a normal random variable. Since a normal random variable is characterized by its mean and variance, we now examine these parameters for Y_n. The expectation of a sum of elements is the sum of the expectations of the elements. Thus,

$$\begin{aligned} E(Y_n) &= E\left(\frac{X_1 + X_2 + \cdots X_n}{n}\right) \\ &= \frac{1}{n}\left[E(X_1) + E(X_2) + \cdots\right] \\ &= \frac{1}{n}\left[nE(X)\right] = m_X \end{aligned} \tag{1.17.3}$$

The sample mean is, of course, an estimate of the true mean of X, and we see from Eq. (1.17.3) that it at least yields $E(X)$ "on the average." Estimators that have this property are said to be unbiased. That is, an estimator is said to be *unbiased* if

$$E(\text{Estimate of } X) = E(X) \tag{1.17.4}$$

Consider next the variance of Y_n. Using Eq. (1.17.3) and recalling that the expectation of the sum is the sum of the expectations, we obtain

$$\begin{aligned} \text{Var } Y_n &= E[Y_n - E(Y_n)]^2 \\ &= E(Y_n^2 - 2Y_n m_X + m_X^2) \\ &= E(Y_n^2) - m_X^2 \end{aligned} \tag{1.17.5}$$

The sample mean Y_n may now be replaced with $(1/n)(X_1 + X_2 \cdots + X_n)$; and, after squaring and some algebraic simplification, Eq. (1.17.5) reduces to

$$\begin{aligned} \text{Var } Y_n &= \frac{1}{n} \text{Var } X \\ &= \frac{\sigma_X^2}{n} \end{aligned} \tag{1.17.6}$$

Thus, we see that the variance of the sample mean decreases with increasing n and eventually goes to zero as $n \to \infty$.

The probability density functions associated with the sample mean are shown in Fig. 1.23 for three values of n. It should be clear from the figure that convergence of some sort takes place as $n \to \infty$. However, no matter how large we make n, there will still remain a nonzero probability that Y_n will fall outside some specified accuracy interval. Thus, we have convergence in only a statistical sense and not in an absolute deterministic sense.

There are a number of types of statistical convergence that have been defined and are in common usage (9, 10). We look briefly at two of these. Consider a sequence of random variables $Y_1, Y_2, \ldots, Y_n$. The sequence Y_n is said to *converge in the mean* (or mean square) to Y if

$$\lim_{n\to\infty} E\left[(Y_n - Y)^2\right] = 0 \tag{1.17.7}$$

Convergence in the mean is sometimes abbreviated as

$$\text{l.i.m. } Y_n = Y \tag{1.17.8}$$

where l.i.m. denotes "limit in the mean."

The sequence Y_n *converges in probability* to Y if

$$\lim_{n\to\infty} P(|Y_n - Y| \geq \varepsilon) = 0 \tag{1.17.9}$$

where ε is an arbitrarily small positive number.

It should be clear from Eqs. (1.17.3) and (1.17.6) that the sample mean converges "in the mean" to m_X. It also converges in probability because the area under the "tails" of the probability density outside a specified interval about m_X goes to zero as $n \to \infty$. Roughly speaking, convergence in the mean indicates that the dispersion (variance) about the limiting value shrinks to zero in the limit. Similarly, convergence in probability means than an arbitrarily small accuracy criterion is met with a probability of one as $n \to \infty$. Davenport and Root (9) point out that convergence in the mean is a more severe requirement than

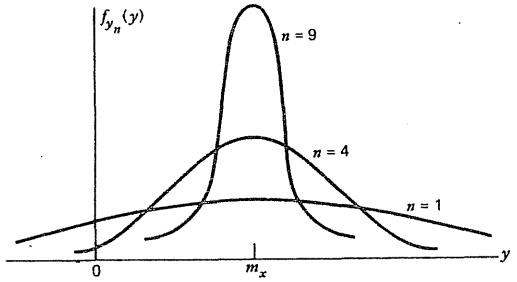


Figure 1.23 Probability density functions illustrating convergence of the sample mean.

convergence in probability. Thus, if a sequence converges in the mean, we are also assured that it will converge in probability. The converse is not true though, because convergence in probability is a "looser" sort of criterion than convergence in the mean.

PROBLEMS

1.1 In straight poker, five cards are dealt to each player from a deck of ordinary playing cards. What is the probability that a player will be dealt a flush (i.e., five cards all of one suit)?

1.2 In the game of blackjack, the player is initially dealt two cards from a deck of ordinary playing cards. Without going into all the details of the game, it will suffice to say here that the best possible hand one could receive on the initial deal is a combination of an ace of any suit and any face card or 10. What is the probability that the player will be dealt this combination?

1.3 An urn contains two white, one black, and four red balls. Three balls are drawn out simultaneously.

(a) How many elements are in the sample space for this experiment?
(b) What is the probability that the draw will produce one white and two red balls?

1.4 Five dice are rolled simultaneously.

(a) Describe the sample space for this experiment.
(b) What is the probability that exactly one 6 will be rolled?
(c) What is the probability that at least one 6 will be rolled?

1.5 Consider a horse race with five entries. The approximate win odds posted by the track just prior to the race are:

Horse Number	Approximate Odds
1	2 to 1
2	3 to 1
3	5 to 1
4	5 to 1
5	11 to 1

Tracks "odds" simply refer to an additional amount returned to the bettor in the event of a win. For example, if the bettor places a bet on horse 1 and that horse wins with 2 to 1 odds, the track will return 1 (the original wager) plus 2 (for winning), or 3 dollars for each dollar wagered.

Assume the race is a chance event and let the sample space consist of five elements: Horse 1 wins, Horse 2 wins, etc. Ignore the small percentage kept by the track (the odds are approximate anyway), and make a reasonable probability assignment to each element in the sample space for this situation.

1.6 Contract bridge is played with an ordinary deck of 52 playing cards. There are four players with players on opposite sides of the table being partners. One player acts as dealer and deals each player 13 cards. A bidding sequence then takes place, and this establishes the trump suit and names the player who is to attempt to take a certain number of tricks. This player is called the declarer. The play begins with the player to the declarer's left leading a card. The declarer's partner's hand is then laid out on the table face up for everyone to see. This then enables the declarer to see a total of 27 cards as the play begins. Knowledge of these cards will, of course, affect his or her strategy in the subsequent play.

Suppose the declarer sees 11 of the 13 trump cards as the play begins. We will assume that the opening lead was not a trump, which leaves 2 trumps outstanding in the opponents' hands. The disposition of these is, of course, unknown to the declarer. There are, however, a limited number of possibilities:

(a) Both trumps lie with the opponent to the left and none to the right.
(b) Both trumps are to the right and none to the left.
(c) The two trumps are split, one in each opponent's hand.

Compute the probabilities for each of the (a), (b), and (c) possibilities.

(*Hint:* Rather than look at all possible combinations, look at numbers of combinations for 25 specific cards held by the opponents just after the opening lead. Two of these will, of course, be specific trump cards. The resulting probability will be the same regardless of the particular choice of specific cards.)

1.7 In the game of craps the casino also has ways of making money other than betting against the player rolling the dice. The other participants standing around the table may place side bets with the casino while waiting their turn with the dice. One such side bet at some tables is "Don't Pass—Bar 12." The participant placing such a side bet wins if the player rolling the dice loses, except for the case of a 12 on the first roll of the dice, and in that case the result is a standoff. In other words, the side bettor is betting against the roller, except that the rules are changed slightly in the side-bet wager to make it such that 12 on the first roll does not win for either the casino or the side bettor. Presumably, this tips the odds in favor of the casino for the side bet.

(a) Find the probability of winning when placing a side bet. "Don't Pass—Bar 12."
(b) It is often said that the casinos make a higher percentage (on the average) on side bets than they do on the regular bet with the player rolling the dice. Compare the results of this problem with those of Example 1.2, relative to the average percentage "take" for the casino. [It works out that this side bet is fairly even. Others are not so favorable. See Goren (4).]

1.8 Assume equal likelihood for the birth of boys and girls. What is the probability that a four-child family chosen at random will have two boys and two girls, irrespective of the order of birth?

[*Note:* The answer is not $\frac{1}{2}$ as might be suspected at first glance.)

1.9 Consider a sequence of random binary digits, zeros and ones. Each digit may be thought of as an independent sample from a sample space containing two elements, each having a probability of $\frac{1}{2}$. For a six-digit sequence, what is the probability of having:

(a) Exactly 3 zeros and 3 ones arranged in any order?
(b) Exactly 4 zeros and 2 ones arranged in any order?
(c) Exactly 5 zeros and 1 one arranged in any order?
(d) Exactly 6 zeros?

1.10 A certain binary message is n bits in length. If the probability of making an error in the transmission of a single bit is p, and if the error probability does not depend on the outcome of any previous transmissions, show that the probability of occurrence of exactly k bit errors in a message is

$$P(k \text{ errors}) = \binom{n}{k} p^k (1-p)^{n-k} \tag{P1.10}$$

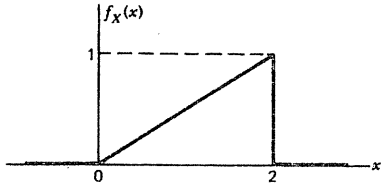
The quantity $\binom{n}{k}$ denotes the number of combinations of n things taken k at a time. (This is a generalization of Problems 1.8 and 1.9.)

1.11 Video poker has recently become a popular game in casinos in the United States (11). The player plays against the machine in much the same way as with slot machines, except that the machine displays cards on a video screen instead of the familiar bells, bars, etc., on spinning wheels. When a coin is put into the machine, it immediately displays five cards on the screen. After this initial five-card deal, the player is allowed to discard one to five cards at his or her discretion and obtain replacement cards (i.e., this is the "draw"). The object, of course, is to try to improve the poker hand with the draw.

(a) Suppose the player is dealt the 3, 7, 8, 10 of hearts and the queen of spades on the initial deál. The player then decides to keep the four hearts and discard the queen of spades in hopes of getting another heart on the draw, and thus obtain a flush (five cards, all of the same suit). The typical video poker machine pays out five coins for a flush. Assume that this is the payout and that the machine is statistically fair. What is the expected (i.e., average) return for this draw situation? (Note that an average return of 1.0 is the break-even return.)
(b) Consider another deal where the initial deal is 3 of spades, 3 of hearts, 10 of spaces, 10 of clubs, and 5 of clubs. Suppose the player decides to discard the 5 of clubs and draw one card in hope of improving the two-pair hand to a full house (i.e., any three-of-a-kind and a pair combination). The typical payouts for two-pair and full-house hands are 2 and 8, respectively. What is the expected return for this draw situation?
(c) Parts (a) and (b) of this problem can obviously be worked intuitively using the relative frequency of occurrence concept. Put your solutions in a more formal setting for each case by defining a suitable sample space, an associated random variable, and an appropriate expectation formula. (For this part, the conditioning may be assumed to be implied and thus suppressed in the sample space.)

1.12 The random variable X may take on all values between 0 and 2, with all values within this range being equally likely.

(a) Sketch the probability density function for X.
(b) Sketch the cumulative probability distribution function for X.
(c) Calculate $E(X)$, $E(X^2)$, and Var X.



Problem 1.13

1.13 A random variable X has a probability density function as shown.

(a) Sketch the cumulative distribution function for X.
(b) What is the variance of X?

1.14 Resistors are a common component in nearly all electronic equipment. Suppose a resistor company manufactures a large lot of resistors whose nominal value is to be 100 Ω. Because of manufacturing tolerances this results in resistance values spread between 90 and 100 Ω (±10 percent of nominal). For the sake of simplicity let us assume the spread is uniform between the two extremes. Let us further speculate that the manufacturer sorts out all the ±5 percent tolerance resistors for sale at a premium price; the remaining resistors are then classified as "10 percent resistors" and are to be sold at the regular price. Let the resistance value of such a "10 percent resistor" be a random variable X.

(a) What is the probability density function for X?
(b) What is the probability that X will lie within 8 percent of the nominal value?

1.15 A random variable X whose probability density function is given by

$$f_X(x) = \begin{cases} \alpha\, e^{-\alpha x}, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

is said to have an exponential probability density function. This density function is sometimes used to describe the failure of equipment components (12, 13). That is, the probability that a particular component will fail within time T is

$$P(\text{failure}) = \int_0^T f_X(x)\, dx \tag{P1.15}$$

Note that α is a parameter that may be adjusted to fit the situation at hand.

(a) Find α for an electronic component whose average lifetime is 10,000 hours. ("Average" is used synonymously with "expectation" here.)
(b) Suppose we wish the probability of failure for the component of part (a) to be less than .01, that is, we wish the reliability to be .99. For what time span might we expect a reliability of .99?

1.16 Consider a sack containing several identical coins whose sides are labeled +1 and −1. A certain number of coins are withdrawn and tossed simultaneously. The algebraic sum of the numbers resulting from the toss is a discrete random variable. Sketch the probability density function associated with the random variable for the following situations:

(a) One coin is tossed.
(b) Two coins are tossed.
(c) Five coins are tossed.
(d) Ten coins are tossed.

The density functions in this case consist of impulses. In the sketches, represent impulses with "arrows" whose lengths are proportional to the magnitudes of the impulses. (This is the discrete analog of Example 1.20. Note the tendency toward central distribution.)

1.17 Two players are matching pennies. Each player begins with a stack of pennies and they uncover their pennies one at a time. By mutual agreement, pennies that match are taken by one player and those that do not match are taken by the other. When either player's original stack is exhausted, the players restack their available pennies and repeat the matching process. This is continued until one or the other player has won all the pennies.

Suppose player A has two pennies and player B has one. Sketch the probability density function for the number of trials required to end the game. The game ends when either player runs out of pennies. (Just as in Problem 1.16, use arrows to represent impulses in the sketch.)

1.18 Let the sum of the dots resulting from the throw of two dice be a discrete random variable X. The probabilities associated with their permissible values 2, 3, . . . , 12 are easily found by itemizing all possible results of the throw (or from Table 1.2). Find $E(X)$ and Var X.

1.19 Discrete random variables X and Y may each take on integer values 1, 3, and 5, and the joint probability of X and Y is given in the table below.

X \ Y	1	3	5
1	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{18}$
3	$\frac{1}{18}$	$\frac{1}{18}$	$\frac{1}{6}$
5	$\frac{1}{18}$	$\frac{1}{6}$	$\frac{1}{3}$

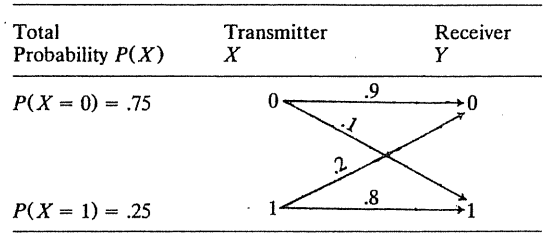
(a) Are random variables X and Y independent?
(b) Find the unconditional probability $P(Y = 5)$.
(c) What is the conditional probability $P(Y = 5|X = 3)$?

1.20 The diagram shown on the opposite page gives the error characteristics of a hypothetical binary transmission system. The numbers shown next to the arrows are the conditional probabilities of Y given X. The unconditional probabilities for X are shown to the left of the figure. Find:

(a) The conditional probabilities $P(X = 0|Y = 1)$ and $P(X = 0|Y = 0)$.
(b) The unconditional probabilities $P(Y = 0)$ and $P(Y = 1)$.
(c) The joint probability array for $P(X, Y)$.

1.21 The Poisson distribution is defined as

$$P(k) = \frac{\lambda^k}{k!} e^{-\lambda} \tag{P1.21}$$



Problem 1.20

where λ is a positive real parameter and $k = 0, 1, 2, \ldots$. This discrete distribution appears in a wide variety of applications in which events occur at random with time (6, 12). For example, a particular telephone call handled by an office might occur with equal likelihood at any time during the working day. Let the average number of calls handled per unit time be α. Then the probability of exactly k calls occurring in a time interval T is

$$P(k) = \frac{(\alpha T)^k}{k!} e^{-\alpha T}$$

Obviously, in this instance αT is the λ parameter of the Poisson distribution.

(a) Let αT be 2 and sketch the corresponding probability density as a function of k for $k = 0, 1, 2, \ldots, 10$. (As before, use arrows for impulses.)
(b) Find the mean for the Poisson distribution.
(*Hint:* The characteristic function discussed in Section 1.8 is useful here. *Answer:* λ.)

1.22 A large power system has an average of three outages per year somewhere in the system. If the outages are assumed to occur randomly, what is the probability of going without an outage for the next year? (See Problem 1.21 for a discussion of the Poisson distribution.)

1.23 Records show that a certain city has had an average of 1 earthquake per decade when averaged over the past 200 years. Assume that earthquakes occur at random and find the possibility that at least one earthquake will occur within:

(a) The next decade.
(b) A person's normal life span (about 70 years).

(See Problem 1.21 for a discussion of the Poisson distribution.)

1.24 The Rayleigh probability density function is defined as

$$f_R(r) = \frac{r}{\sigma^2} e^{-r^2/2\sigma^2} \tag{P1.24}$$

where σ^2 is a parameter of the distribution (see Example 1.23).

(a) Find the mean and variance of a Rayleigh distributed random variable R.
(b) Find the mode of R (i.e., the most likely value of R).

1.25 Three similar "unfair" coins are tossed simultaneously. The coins are unfair in that $P(\text{Heads}) = .6$ and $P(\text{Tails}) = .4$. Let the discrete random variable

X be defined to be the number of heads that result from the toss of the three coins. Find the discrete probability distribution associated with the random variable X.

1.26 The target shooting example of Section 1.14 led to the Rayleigh density function specified by Eq. (1.14.35) (also repeated in Problem 1.24).

(a) Show that the probability that a hit will lie within a specified distance R_0 from the origin is given by

$$P(\text{Hit lies within } R_0) = 1 - e^{-R_0^2/2\sigma^2} \tag{P1.26}$$

(b) The value of R_0 in Eq. (P1.26) that yields a probability of .5 is known as the circular probable error (or circular error probable, CEP). Find the CEP in terms of σ.

1.27 Find the mean and variance of the output of a half-wave rectifier driven by Gaussian noise. (The probability density function for the output is given in Example 1.17.)

1.28 Consider a random variable X with an exponential probability density function

$$f_X(x) = \begin{cases} e^{-x}, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

Find:

(a) $P(X \geq 2)$.
(b) $P(1 \leq X \leq 2)$.
(c) $E(X)$, $E(X^2)$, and Var X.

1.29 Random variables X and Y have a joint probability density function defined as follows:

$$f_{XY}(x, y) = \begin{cases} .25, & -1 \leq x \leq 1 \quad \text{and} \quad -1 \leq y \leq 1 \\ 0, & \text{otherwise} \end{cases}$$

Are random variables X and Y statistically independent?
[*Hint:* Integrate with respect to appropriate dummy variables to obtain $f_X(x)$ and $f_Y(y)$. Then check to see if the product of f_X and f_Y is equal to f_{XY}.]

1.30 Random variables X and Y have a joint probability density function

$$f_{XY}(x, y) = \begin{cases} e^{-(x+y)}, & x \geq 0 \quad \text{and} \quad y \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

Find:

(a) $P(X \leq \frac{1}{2})$.
(b) $P[(X + Y) \leq 1]$.
(c) $P[(X \text{ or } Y) \geq 1]$.
(d) $P[(X \text{ and } Y) \geq 1]$.

1.31 Are the random variables of Problem 1.30 statistically independent?

1.32 Random variables X and Y are statistically independent and their respective probability density functions are

$$f_X(x) = \tfrac{1}{2}e^{-|x|}$$
$$f_Y(y) = e^{-2|y|}$$

Find the probability density function associated with $X + Y$.
(*Hint:* Fourier transforms are helpful here.)

1.33 Random variable X has a probability density function

$$f_X(x) = \begin{cases} \frac{1}{2}, & -1 \leq x \leq 1 \\ 0, & \text{otherwise} \end{cases}$$

Random variable Y is related to X through the equation

$$y = x^3 + 1$$

What is the probability density function for Y?

1.34 X and Y are independent, zero-mean random variables with variances σ_X^2 and σ_Y^2. Another set of random variables U and V are related to X and Y through the equations

$$u = 2x + y$$
$$v = x - y$$

Find the correlation coefficient of U and V. Let $\sigma_X^2 = \sigma_Y^2$.

1.35 The vector Gaussian random variable

$$\mathbf{X} = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$$

is completely described by its mean and covariance matrix. In this case, they are

$$\mathbf{m}_X = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$
$$\mathbf{C}_X = \begin{bmatrix} 4 & 1 \\ 1 & 1 \end{bmatrix}$$

Now consider another vector random variable $\mathbf{Y}$ that is related to $\mathbf{X}$ by the equation

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b}$$

where

$$\mathbf{A} = \begin{bmatrix} 2 & 1 \\ 1 & -1 \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Find the mean and covariance matrix for $\mathbf{Y}$.

1.36 The general bivariate normal density function is given by Eq. (1.16.1). It should be apparent that the unconditional density functions $f_{X_1}(x_1)$ and $f_{X_2}(x_2)$ are also normal in form. (A linear transformation yielding uncorrelated random variables may be made, and then the normal form is obvious.) Show that the conditional density functions $f_{X_1|X_2}(x_1)$ and $f_{X_2|X_1}(x_2)$ are also normal in form.

(*Hint:* Simply noting the quadratic form in the exponential is sufficient justification of normality.)

1.37 A pair of random variables, X and Y, have a joint probability density function

$$f_{XY}(x, y) = \begin{cases} 1, & 0 \le y \le 2x \quad \text{and} \quad 0 \le x \le 1 \\ 0, & \text{elsewhere} \end{cases}$$

Find:

$$E(X|Y = .5)$$

[*Hint:* Use Eq. (1.11.19) to find $f_{X|Y}(x)$ for $y = .5$, and then integrate $xf_{X|Y}(x)$ to find $E(X|Y = .5)$.]

1.38 Two continuous random variables X and Y have a joint probability density function that is uniform inside the unit circle and zero outside, that is,

$$f_{XY}(x, y) = \begin{cases} 1/\pi, & (x^2 + y^2) \le 1 \\ 0, & (x^2 + y^2) > 1 \end{cases}$$

(a) Find the unconditional probability density function for the random variable Y and sketch the probability density as a function of Y.
(b) Are the random variables X and Y statistically independent?

1.39 A normal random variable X is described by $N(0, 4)$. Similarly, Y is normal and is described by $N(1, 9)$. X and Y are independent. Another random variable Z is defined by the additive combination

$$Z = X + 2Y$$

Write the explicit expression for the probability density function for Z. Use appropriate numerical values for all parameters.

1.40 In the following exercises, X denotes a $N(0, 1)$ random variable.
(a) Calculate $P(X > 4.5)$ using both MATLAB's quad and quad8 integration functions, and compare the results with those obtained from Table 1.9. Note that in the integration you will have to experiment with a numerical upper limit that will give a good approximation to the infinite limit. Feller (5) gives an approximate formula for the area under the "tail" of the normal density function:

$$1 - F_X(x) \approx \frac{1}{\sqrt{2\pi}x} e^{-\frac{1}{2}x^2} \qquad \text{(P1.40)}$$

This can be used to find a suitable starting point in the trial-and-error experimentation.
(b) Feller (5) states that the approximation made in Eq. (P1.40) improves rapidly as x increases. To verify this statement (or otherwise), calculate $1 - F_X(x)$ from Eq. (P1.40) for $x = 4.5$, and then again for $x = 6.5$. Compare the results with the more accurate values obtained using MATLAB.

[*Note:* The calculated probability is reduced by roughly five orders of magnitude in going from 4.5 to 6.5, but the improvement in the accuracy of Eq. (P1.40) is not nearly this dramatic.]

1.41 A probability formula for calculating the result of n trials of a binary statistical experiment is given in Problem 1.10. This formula is deceptively simple. If n and k are small, the indicated probability can be computed in a matter of seconds with just a hand-held calculator. However, the calculation can be both laborious and tricky when k and n are large. Numerical problems due to overflow/underflow can easily be encountered under these conditions, because k and $(n - k)$ appear in the equation as exponents. Thus, the formula given by Eq. (P1.10) should be programmed with care. The following example is intended to illustrate a case where n is relatively large.

One of the test procedures for GPS navigation equipment involves simulating a satellite range error and then testing the receiver equipment to see if it can reliably detect the range error with its built-in measurement consistency-checking algorithm. (See Chapter 11 for a brief discussion of the GPS navigation system.) The specifications for detecting the satellite range error are quite severe, and they require that the probability of a missed detection (in the presence of noise) be no greater than .001 (14).

Consider a simulation test where there are to be 5000 trials. There will be only two possible results from each trial: (1) equipment fails to detect the satellite range error (can be interpreted as "error" in Eq. P1.10), and (2) equipment successfully detects the range error (interpreted as "no error" in Eq. P1.10). Suppose that the testing procedure is to be designed to give the receiver-equipment manufacturer a generous chance of passing the 5000-trial test. How many failures (i.e., "errors" in the binary probability formula) should the equipment manufacturer be allowed in order that the probability of passing the test be approximately .98?

[*Hint:* Write and execute a MATLAB program to calculate the probabilities of occurrence of exactly 0, 1, 2, . . . , m failures, and then sum the results to get the cumulative probability of obtaining no more than m failures in the 5000-trial test. The m parameter is variable and will have to be determined by trial-and-error, so write your program such that this parameter can be changed easily in the editor. Note that the expected number of failures is 5 (for $p = .001$ and $n = 5000$), so this is a good starting point for your trial-and-error iteration. Also, as a precaution against overflow/underflow problems, it is best to calculate the log of the various probabilities first, and then obtain the desired probability as exp(log prob) at the end of the calculation.]

1.42 Consider a random variable that is defined to be the sum of the squares of n independent normal random variables, all of which are $N(0, 1)$. The parameter n is any positive integer. Such a sum-of-the-squares random variable is called a chi-square random variable with n degrees of freedom. The probability density function associated with a chi-square random variable X is

$$f_X(x) = \begin{cases} \dfrac{x^{(n/2-1)}e^{-x/2}}{2^{n/2}\Gamma\left(\dfrac{n}{2}\right)}, & x > 0 \\ 0, & x \le 0 \end{cases}$$

where Γ indicates the gamma function (3, 10). It is not difficult to show that the mean and variance of X are given by

$$E(X) = n$$

$$\text{Var } X = 2n$$

[This is easily derived by noting that the defining integral expressions for the first and second moments of X are in the exact form of a single-sided Laplace transform with $s = \frac{1}{2}$. See Appendix A for a table of Laplace transforms and note that $n! = \Gamma(n + 1)$.]

(a) Make rough sketches of the chi-square probability density for $n = 1$, 2, and 4. (You will find MATLAB useful here.)

(b) Note that the chi-square random variable for $n > 1$ is the sum of independent random variables that are radically non-Gaussian (e.g., look at the sketch of f_X for $n = 1$). According to the central limit theorem, there should be a tendency toward normality as we sum more and more such random variables. Using MATLAB, create an m-file for the chi-square density function for $n = 16$. Plot this function along with the normal density function for a $N(16, 32)$ random variable (same mean and sigma as the chi-square random variable). This is intended to demonstrate the tendency toward normality, even when the sum contains only 16 terms.

REFERENCES CITED IN CHAPTER 1

1. N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series,* Cambridge, MA: MIT Press and New York: Wiley, 1949.
2. S. O. Rice, "Mathematical Analysis of Noise," *Bell System Tech. J., 23,* 282–332 (1994); *24,* 46–256 (1945).
3. A. M. Mood, F. A. Graybill, and D. C. Boes, *Introduction to the Theory of Statistics,* 3rd ed., New York: McGraw-Hill, 1974.
4. C. H. Goren, *Go With the Odds,* New York: Macmillan, 1969.
5. W. Feller, *An Introduction to Probability Theory and Its Applications,* Vol. 1, 2nd ed., New York: Wiley, 1957.
6. P. Beckmann, *Probability in Communication Engineering,* New York: Harcourt, Brace, and World, 1967.
7. A. Papoulis, *Probability, Random Variables, and Stochastic Processes,* 2nd ed., New York: McGraw-Hill, 1984.
8. G. H. Golub and C. F. Van Loan, *Matrix Computations,* 2nd ed., Baltimore, MD: The Johns Hopkins University Press, 1989.
9. W. B. Davenport, Jr. and W. L. Root, *An Introduction to the Theory of Random Signals and Noise,* New York: McGraw-Hill, 1958.
10. K. S. Shanmugan and A. M. Breipohl, *Random Signals: Detection, Estimation, and Data Analysis,* New York: Wiley, 1988.
11. D. Gerhardt and T. Korfman, *Video Poker, Playing to Win,* Las Vegas, NV: Gaming Books International, Inc. 1987.
12. I. F. Blake, *An Introduction to Applied Probability,* New York: Wiley, 1979.
13. H. J. Larson and B. O. Shubert, *Probabilistic Models in Engineering Sciences,* Vol. 1, New York: Wiley, 1979.
14. *Minimum Operational Performance Standards for Airborne Supplemental Navigation Equipment Using Global Positioning System (GPS),* Document No. RTCA/DO-208, RTCA, Washington, DC, July 1991.

Additional References on Probability

15. P. Z. Peebles, Jr., *Probability, Random Variables, and Random Signal Principles,* 3rd ed., New York: McGraw-Hill, 1993.
16. G. R. Cooper and C. D. McGillem, *Probabilistic Methods of Signal and System Analysis,* 2nd ed., New York: Holt, Rinehart, and Winston, 1986.
17. A. M. Breipohl, *Probabilistic Systems Analysis,* New York: Wiley, 1970.
18. J. L. Melsa and A. P. Sage, *An Introduction to Probability and Stochastic Processes,* Englewood Cliffs, NJ: Prentice-Hall, 1973.

2

Mathematical Description of Random Signals

The concept of frequency spectrum is familiar from elementary physics, and so it might seem appropriate to begin our discussion of noiselike signals with their spectral description. This approach, while intuitively appealing, leads to all sorts of difficulties. The only really careful way to describe noise is to begin with a probabilistic description and then proceed to derive the associated spectral characteristics from the probabilistic model. We now proceed toward this end.

2.1 CONCEPT OF A RANDOM PROCESS

We should begin by distinguishing between deterministic and random signals. Usually, the signals being considered here will represent some physical quantity such as voltage, current, distance, temperature, and so forth. Thus they are real variables. Also, time will usually be the independent variable, although this does not necessarily need to be the case. A signal is said to be *deterministic* if it is exactly predictable for the time span of interest. Examples would be

(a) $x(t) = 10 \sin 2\pi t$ (sine wave)

(b) $x(t) = \begin{cases} 1, & t \geq 0 \\ 0, & t < 0 \end{cases}$ (unit step)

(c) $x(t) = \begin{cases} 1 - e^{-t}, & t \geq 0 \\ 0, & t < 0 \end{cases}$ (exponential response)

Notice that there is nothing "chancy" about any of these signals. They are described by functions in the usual mathematical sense; that is, specify a numerical value of t and the corresponding value of x is determined. We are usually

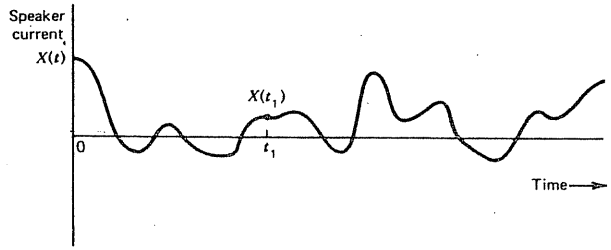


Figure 2.1 Typical radio noise signal.

able to write the functional relationship between x and t explicitly. However, this is not really necessary. All that is needed is to know conceptually that a functional relationship exists.

In contrast with a deterministic signal, a *random signal* always has some element of chance associated with it. Thus, it is not predictable in a deterministic sense. Examples of random signals are:

(d) $X(t) = 10 \sin(2\pi t + \theta)$, where θ is a random variable uniformly distributed between 0 and 2π.

(e) $X(t) = A \sin(2\pi t + \theta)$, where θ and A are independent random variables with known distributions.

(f) $X(t)$ = A noiselike signal with no particular deterministic structure—one that just wanders on aimlessly ad infinitum.

Since all of these signals have some element of chance associated with them, they are random signals. Signals such as (d), (e), and (f) are formally known as *random* or *stochastic processes,* and we will use the terms *random* and *stochastic* interchangeably throughout the remainder of the book.*

Let us now consider the description of signal (f) in more detail. It might be the common audible radio noise that was mentioned in Chapter 1. If we looked at an oscillographic recording of the radio speaker current, it might appear as shown in Fig. 2.1. We might expect such a signal to have some kind of spectral description, because the signal is audible to the human ear. Yet the precise mathematical description of such a signal is remarkably elusive, and it eluded investigators prior to the 1940s (3, 4).

Imagine sampling the noise shown in Fig. 2.1 at a particular point in time, say, t_1. The numerical value obtained would be governed largely by chance, which suggests it might be considered to be a random variable. However, with

* We need to recognize a notational problem here. Denoting the random process as $X(t)$ implies that there is a functional relationship between X and t. This, of course, is not the case because $X(t)$ is governed by chance. For this reason, some authors (1, 2) prefer to use a subscript notation, that is, X_t rather than $X(t)$, to denote a random time signal. X_t then "looks" like a random variable with time as a parameter, which is precisely what it is. This notation, however, is not without its own problems. Suffice it to say, in most engineering literature, time random processes are denoted with "parentheses t" rather than "subscript t." We will do likewise, and the reader will simply have to remember that $X(t)$ does not mean function in the usual mathematical sense when $X(t)$ is a random process.

random variables we must be able to visualize a conceptual statistical experiment in which samples of the random variable are obtained under identical chance circumstances. It would not be proper in this case to sample X by taking successive time samples of the same signal, because, if they were taken very close together, there would be a close statistical connection among nearby samples. Therefore, the conceptual experiment in this case must consist of many "identical" radios, all playing simultaneously, all being tuned away from regular stations in different portions of the broadcast band, and all having their volumes turned up to the same sound level. This then leads to the notion of an ensemble of similar noiselike signals as shown in Fig. 2.2.

It can be seen then that a random process is a set of random variables that unfold with time in accordance with some conceptual chance experiment. Each of the noiselike time signals so generated is called a *sample realization* of the process. Samples of the individual signals at a particular time t_1 would then be sample realizations of the random variable $X(t_1)$. Four of these are illustrated in Fig. 2.2 as $X_A(t_1)$, $X_B(t_1)$, $X_C(t_1)$, and $X_D(t_1)$. If we were to sample at a different time, say, t_2, we would obtain samples of a different random variable $X(t_2)$, and so forth. Thus, in this example, an infinite set of random variables is generated by the random process $X(t)$.

The radio experiment just described is an example of a *continuous-time random process* in that time evolves in a continuous manner. In this example, the probability density function describing the amplitude variation also happens to be continuous. However, random processes may also be discrete in either time or amplitude, as will be seen in the following two examples.

EXAMPLE 2.1

Consider a card player with a deck of standard playing cards numbered from 1 (ace) through 13 (king). The deck is shuffled and the player picks a card at

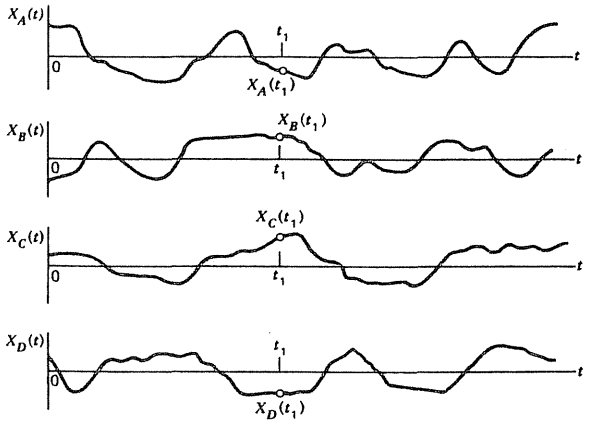


Figure 2.2 Ensemble of sample realizations of a random process.

random and observes its number. The card is then replaced, the deck reshuffled, and another card observed. This process is then repeated at unit intervals of time and continued on ad infinitum. The random process so generated would be discrete in both time and "amplitude," provided we say that the observed number applies only at the precise instant of time it is observed.

The preceding description would, of course, generate only one sample realization of the process. In order to obtain an ensemble of sample signals, we need to imagine an ensemble of card players, each having similar decks of cards and each generating a different (but statistically similar) sample realization of the process. ▩

EXAMPLE 2.2

Imagine a sack containing a large quantity of sample numbers taken from a zero-mean, unity-variance normal distribution. An observer reaches into the sack at unit intervals of time and observes a number with each trial. In order to avoid exact repetition, he does not replace the numbers during the experiment. This process would be discrete in time, as before, but continuous in amplitude. Also, the conceptual experiment leading to an ensemble of sample realizations of the process would involve many observers, each with a separate sack of random numbers. ▩

We will concentrate mostly on continuous-time processes in this chapter. However, a brief discussion of discrete-time Markov and Wiener processes is given in Problems 2.33, 2.34, and 2.35 at the end of the chapter. Scalar discrete-time processes are then considered further in Chapter 3 with the discussion of the ARMA model in Section 3.9. Finally, vector discrete-time models are introduced in Chapter 5 and then used in all of the subsequent chapters.

Auto Progressive Moving Average

2.2 PROBABILISTIC DESCRIPTION OF A RANDOM PROCESS

As mentioned previously, one can usually write out the functional form for a deterministic signal explicitly; for example, $s(t) = 10 \sin 2\pi t$, or $s(t) = t^2$, and so on. No such deterministic description is possible for random signals because the numerical value of the signal at any particular time is governed by chance. Thus, we should expect our description of noiselike signals to be somewhat vaguer than that for deterministic signals. One way to specify a random process is to describe in detail the conceptual chance experiment giving rise to the process. Examples 2.1 and 2.2 illustrate this way of describing a random process. The following two examples will illustrate this further.

EXAMPLE 2.3

Consider a time signal (e.g., a voltage) that is generated according to the following rules: (a) The waveform is generated with a sample-and-hold arrangement where the "hold" interval is 1 sec; (b) the successive amplitudes are

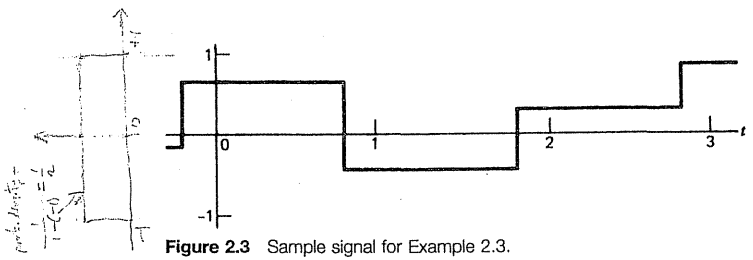


Figure 2.3 Sample signal for Example 2.3.

independent samples taken from a set of random numbers with uniform distribution from -1 to $+1$; and (c) the first switching time after $t = 0$ is a random variable with uniform distribution from 0 to 1. (This is equivalent to saying the time origin is chosen at random.) A typical sample realization of this process is shown in Fig. 2.3. Note that the process mean is zero and its mean-square value works out to be one-third. [This is obtained from item (b) of the description.] ■

EXAMPLE 2.4

Consider another time function generated with a sample-and-hold arrangement with these properties: (a) The "hold" interval is 0.2 sec, (b) the successive amplitudes are independent samples obtained from a zero-mean normal distribution with a variance of one-third, and (c) the switching points occur at multiples of .2 units of time; that is, the time origin is not chosen at random in this case. A sketch of a typical waveform for this process is shown in Fig. 2.4. ■

Now, from Examples 2.3 and 2.4 it should be apparent that if we simply say, "Noiselike waveform with zero mean and mean-square value of one-third," we really are not being very definite. Both processes of Examples 2.3 and 2.4 would satisfy these criteria, but yet they are quite different. Obviously, more information than just mean and variance is needed to completely describe a random process. We will now explore the "description" problem in more detail.

A more typical "noiselike" signal is shown in Fig. 2.5. The times indicated, $t_1, t_2, \ldots, t_k$, have been arranged in ascending order, and the corresponding sample values $X_1, X_2, \ldots, X_k$ are, of course, random variables. Note that we

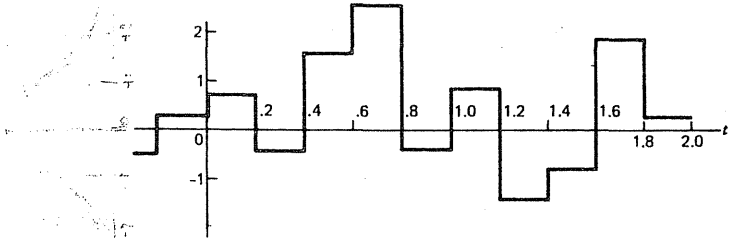


Figure 2.4 Typical waveform for Example 2.4.

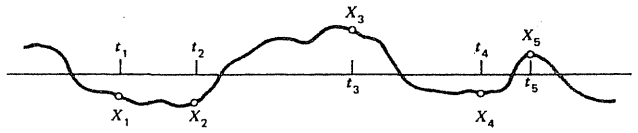


Figure 2.5 Sample signal of a typical noise process.

have abbreviated the notation and have let $X(t_1) = X_1$, $X(t_2) = X_2$, . . . , and so on. Obviously, the first-order probability density functions $f_{X_1}(x)$, $f_{X_2}(x)$, . . . , $f_{X_k}(x)$ are important in describing the process because they tell us something about the process amplitude distribution. In Example 2.3, $f_{X_1}(x)$, $f_{X_2}(x)$, . . . , $f_{X_k}(x)$, are all identical density functions and are given by [using $f_{X_1}(x)$ as an example]

$$f_{X_1}(x) = \begin{cases} \frac{1}{2}, & -1 \le x \le 1 \\ 0, & |x| > 1 \end{cases}$$

The density functions are not always identical for $X_1, X_2, \ldots, X_k$; they just happened to be in this simple example. In Example 2.4, the density functions describing the amplitude distribution of the $X_1, X_2, \ldots, X_k$ random variables are again all the same, but in this case they are normal in form with a variance of one-third. Note that the first-order densities tell us something about the relative distribution of the process amplitude as well as its mean and mean-square value.

It should be clear that the joint densities relating any pair of random variables, for example, $f_{X_1X_2}(x_1, x_2)$, $f_{X_1X_3}(x_1, x_3)$, and so forth, are also important in our process description. It is these density functions that tell us something about how rapidly the signal changes with time, and these will eventually tell us something about the signal's spectral content. Continuing on, the third, fourth, and subsequent higher-order density functions provide even more detailed information about the process in probabilistic terms. However, this leads to a formidable description of the process, to say the least, because a k-variate density function is required where k can be any positive integer. Obviously, we will not usually be able to specify this kth order density function explicitly. Rather, this usually must be done more subtly by providing, with a word description or otherwise, enough information about the process to enable one to write out any desired higher-order density function; but the actual "writing it out" is usually not done.

Recall from probability theory that two random variables X and Y are said to be statistically independent if their joint density function can be written in product form

$$f_{XY}(x, y) = f_X(x) f_Y(y) \tag{2.2.1}$$

Similarly, random processes $X(t)$ and $Y(t)$ are statistically independent if the joint density for any combination of random variables of the two processes can be written in product form, that is, $X(t)$ and $Y(t)$ are independent if

$$f_{X_1X_2\ldots Y_1Y_2\ldots} = f_{X_1X_2\ldots}f_{Y_1Y_2\ldots} \tag{2.2.2}$$

In Eq. (2.2.2) we are using the shortened notation $X_1 = X(t_1)$, $X_2 = X(t_2)$, . . . , and $Y_1 = Y_1(t_1')$, $Y_2 = Y_2(t_2')$, . . . , where the sample times do not have to be the same for the two processes.

In summary, the test for completeness of the process description is this: Is enough information given to enable one, conceptually at least, to write out the kth order probability density function for any k? If so, the description is as complete as can be expected; if not, it is incomplete to some extent, and radically different processes may fit the same incomplete description.

2.3 GAUSSIAN RANDOM PROCESS

There is one special situation where an explicit probability density description of the random process is both feasible and appropriate. This case is the *Gaussian* or *normal* random process. It is defined as one in which *all the density functions describing the process are normal in form.* Note that it is not sufficient that just the "amplitude" of the process be normally distributed; all higher-order density functions must also be normal! As an example, the process defined in Example 2.4 has a normal first-order density function, but closer scrutiny will reveal that its second-order density function is not normal in form. Thus, the process is not a Gaussian process.

The multivariate normal density function was discussed in Section 1.15. It was pointed out there that matrix notation makes it possible to write out all k-variate density functions in the same compact matrix form, regardless of the size of k. All we have to do is specify the vector random-variable mean and covariance matrix, and the density function is specified. In the case of a Gaussian random process the "variates" are the random variables $X(t_1)$, $X(t_2)$, . . . , $X(t_k)$, where the points in time may be chosen arbitrarily. Thus, enough information must be supplied to specify the mean and covariance matrix regardless of the choice of t_1, t_2, . . . , t_k. Examples showing how to do this will be deferred for the moment, because it is expedient first to introduce the basic ideas of stationarity and correlation functions.

2.4 STATIONARITY, ERGODICITY, AND CLASSIFICATION OF PROCESSES

A random process is said to be *time stationary* or simply *stationary* if the density functions describing the process are invariant under a translation of time. That is, if we consider a set of random variables $X_1 = X(t_1)$, $X_2 = X(t_2)$, . . . , $X_k = X(t_k)$, and also a translated set $X_1' = X(t_1 + \tau)$, $X_2' = X(t_2 + \tau)$, . . . , $X_k' = X(t_k + \tau)$, the density functions f_{X_1}, $f_{X_1X_2}$,...,$f_{X_1X_2\ldots X_k}$ describing the first set would be identical in form to those describing the translated set. Note that this applies to

all the higher-order density functions. The adjective *strict* is also used occasionally with this type of stationarity to distinguish it from wide-sense stationarity, which is a less restrictive form of stationarity. This will be discussed later in Section 2.5 on correlation functions.

A random process is said to be *ergodic* if time averaging is equivalent to ensemble averaging. In a qualitative sense this implies that a single sample time signal of the process contains all possible statistical variations of the process. Thus, no additional information is to be gained by observing an ensemble of sample signals over the information obtained from a one-sample signal, for example, one long strip-chart recording. An example will illustrate this concept.

EXAMPLE 2.5

Consider a somewhat trivial process defined to be a constant with time, the constant being a random variable with zero-mean normal distribution. An ensemble of sample realizations for this process is shown in Fig. 2.6. A common physical situation where this kind of process model would be appropriate is random instrument bias. In many applications, some small residual random bias will remain in spite of all attempts to eliminate it, and the bias will be different

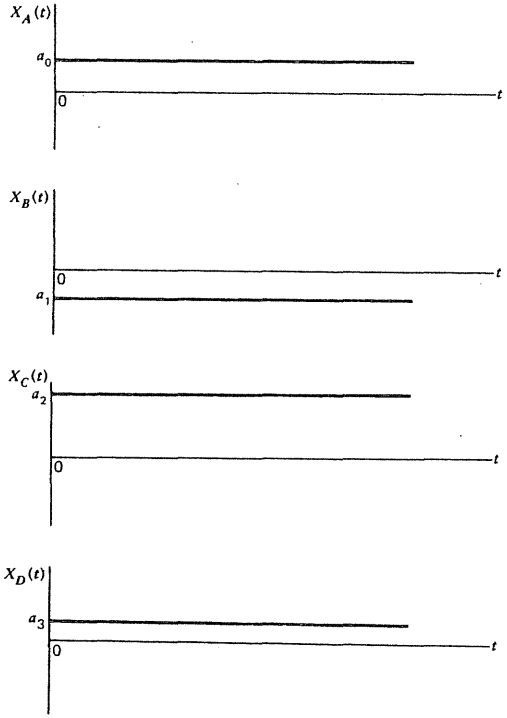


Figure 2.6 Ensemble of random constants.

for each instrument in the batch. In Fig. 2.6 we see that time samples collected from a single sample signal, say the first one, will all have the same value a_0. The average of these is, of course, just a_0. On the other hand, if we were to collect samples in an ensemble sense, the values $a_0, a_1, a_2, \ldots, a_n$, would be obtained. These would have a normal distribution with zero mean. Obviously, time and ensemble sampling do not lead to the same result in this case, so the process is not ergodic. It is, however, a stationary process because the "statistics" of the process do not change with time. ■

In the case of physical noise processes, one can rarely justify strict stationarity or ergodicity in a formal sense. Thus, we often lean on heuristic knowledge of the processes involved and simply make assumptions accordingly.

Random processes are sometimes classified according to two categories, *deterministic* and *nondeterministic*. As might be expected, a deterministic random process resembles a deterministic nonrandom signal in that it has some special deterministic structure. Specifically, if the process description is such that knowledge of a sample signal's past enables exact prediction of its future, it is classified as a deterministic random process. Examples are:

1. $X(t) = a$; a is normal, $N(m, \sigma^2)$.
2. $X(t) = A \sin \omega t$; A is Rayleigh distributed.
3. $X(t) = A \sin(\omega t + \theta)$; A and θ are independent, and Rayleigh and uniformly distributed, respectively.

In each case, if one were to specify a particular sample signal prior to some time, say, t_1, the sample realizations for that particular signal would be indirectly specified, and the signal's future values would be exactly predictable.

Random processes that are not deterministic are classified as nondeterministic. These processes have no special functional structure that enables their exact prediction by specification of certain key parameters or their past history. Typical "noise" is a good example of a nondeterministic random process. It wanders on aimlessly, as determined by chance, and has no particular deterministic structure.

2.5 AUTOCORRELATION FUNCTION

The autocorrelation function for a random process $X(t)$ is defined as*

$$R_X(t_1, t_2) = E[X(t_1)X(t_2)] \tag{2.5.1}$$

* In describing the correlation properties of random processes, some authors prefer to work with the *autocovariance function* rather than the autocorrelation function as defined by Eq. (2.5.1). The autocovariance function is defined as

$$\text{Autocovariance function} = E\{[X(t_1) - m_x(t_1)][X(t_2) - m_x(t_2)]\}$$

The two functions are obviously related. In one case the mean is included in the product (autocorrelation), and in the other the mean is subtracted out (autocovariance). That is the essential difference. The two functions are, of course, identical for zero-mean processes. The autocorrelation function is probably the more common of the two in engineering literature, so it will be used throughout this text.

where t_1 and t_2 are arbitrary sampling times. Clearly, it tells how well the process is correlated with itself at two different times. If the process is stationary, its probability density functions are invariant with time, and the autocorrelation function depends only on the time difference $t_2 - t_1$. Thus, R_X reduces to a function of just the time difference variable τ, that is,

$$R_X(\tau) = E[X(t)X(t + \tau)] \quad \text{(stationary case)} \tag{2.5.2}$$

where t_1 is now denoted as just t and t_2 is $(t + \tau)$. Stationarity assures us that the expectation is not dependent on t.

Note that the autocorrelation function is the ensemble average (i.e., expectation) of the product of $X(t_1)$ and $X(t_2)$; therefore, it can formally be written as

$$R_X(t_1, t_2) = E[X_1X_2] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_1x_2 f_{X_1X_2}(x_1, x_2)\, dx_1\, dx_2 \tag{2.5.3}$$

where we are using the shortened notation $X_1 = X(t_1)$ and $X_2 = X(t_2)$. However, Eq. (2.5.3) is often not the simplest way of determining R_X because the joint density function $f_{X_1X_2}(x_1, x_2)$ must be known explicitly in order to evaluate the integral. If the ergodic hypothesis applies, it is often easier to compute R_X as a time average rather than an ensemble average. An example will illustrate this.

EXAMPLE 2.6

Consider the same process defined in Example 2.3. A typical sample signal for this process is shown in Fig. 2.7 along with the same signal shifted in time an amount τ. Now, the process under consideration in this case is ergodic, so we should be able to interchange time and ensemble averages. Thus, the autocorrelation function can be written as

$$\begin{aligned} R_X(\tau) &= \text{time average of } X_A(t) \cdot X_A(t + \tau) \\ &= \lim_{T\to\infty} \frac{1}{T}\int_0^T X_A(t)X_A(t + \tau)\, dt \end{aligned} \tag{2.5.4}$$

It is obvious that when $\tau = 0$, the integral of Eq. (2.5.4) is just the mean square value of $X_A(t)$, which is $\frac{1}{3}$ in this case. On the other hand, when τ is unity or larger, there is no overlap of the correlated portions of $X_A(t)$ and $X_A(t + \tau)$, and

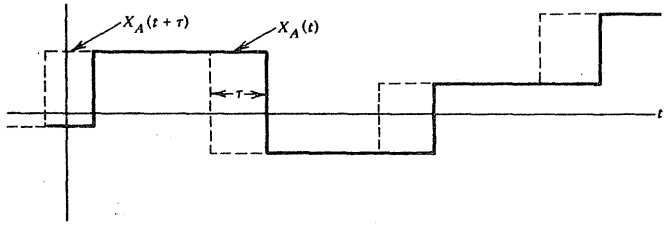


Figure 2.7 Random waveform for Example 2.6.

thus the average of the product is zero. Now, as the shift τ is reduced from 1 to 0, the overlap of correlated portions increases linearly until the maximum overlap occurs at $\tau = 0$. This then leads to the autocorrelation function shown in Fig. 2.8. Note that for stationary ergodic processes, the direction of time shift τ is immaterial, and hence the autocorrelation function is symmetric about the origin. Also, note that we arrived at $R_X(\tau)$ without formally finding the joint density function $f_{X_1X_2}(x_1, x_2)$. ■

Sometimes, the random process under consideration is not ergodic, and it is necessary to distinguish between the usual autocorrelation function (ensemble average) and the time-average version. Thus, we define the *time autocorrelation function* as

$$\mathcal{R}_{X_A}(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_0^T X_A(t)X_A(t + \tau)\, dt \tag{2.5.4a}$$

where $X_A(t)$ denotes a sample realization of the $X(t)$ process. There is the tacit assumption that the limit indicated in Eq. (2.5.4a) exists. Also note that script $\mathcal{R}$ rather than italic R is used as a reminder that this is a time average rather than an ensemble average.

EXAMPLE 2.7

To illustrate the difference between the usual autocorrelation function and the time autocorrelation function, consider the deterministic random process

$$X(t) = A \sin \omega t \tag{2.5.5}$$

where A is a normal random variable with zero mean and variance σ^2, and ω is a known constant. Suppose we obtain a single sample of A and its numerical value is A_1. The corresponding sample of $X(t)$ would then be

$$X_A(t) = A_1 \sin \omega t \tag{2.5.6}$$

According to Eq. (2.5.4a), its time autocorrelation function would then be

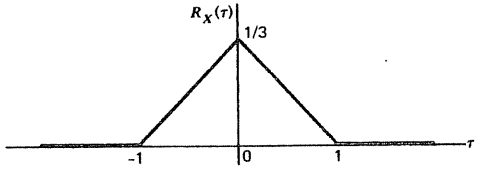


Figure 2.8 Autocorrelation function for Example 2.6.

$$\mathcal{R}_{X_A}(\tau) = \lim_{T\to\infty} \frac{1}{T} \int_0^T A_1 \sin \omega t \cdot A_1 \sin \omega(t + \tau)\, dt$$

$$= \frac{A_1^2}{2} \cos \omega\tau \tag{2.5.7}$$

On the other hand, the usual autocorrelation function is calculated as an ensemble average, that is, from Eq. (2.5.1). In this case, it is

$$R_X(t_1, t_2) = E[X(t_1)X(t_2)]$$

$$= E[A \sin \omega t_1 \cdot A \sin \omega t_2]$$

$$= \sigma^2 \sin \omega t_1 \sin \omega t_2 \tag{2.5.8}$$

Note that this expression is quite different from that obtained for $\mathcal{R}_{X_A}(\tau)$. Clearly, time averaging does not yield the same result as ensemble averaging, so the process is not ergodic. Furthermore, the autocorrelation function given by Eq. (2.5.8) does not reduce to simply a function of $t_2 - t_1$. Therefore, the process is not stationary. ■

General Properties of Autocorrelation Functions

There are some general properties that are common to all autocorrelation functions for stationary processes. These will now be enumerated with a brief comment about each:

1. $R_X(0)$ is the mean-square value of the process $X(t)$. This is self-evident from Eq. (2.5.2).
2. $R_X(\tau)$ is an even function of τ. This results from the stationarity assumption. [In the nonstationary case there is symmetry with respect to the two arguments t_1 and t_2. In Eq. (2.5.1) it certainly makes no difference in which order we multiply $X(t_1)$ and $X(t_2)$. Thus, $R_X(t_1, t_2) = R_X(t_2, t_1)$.]
3. $|R_X(\tau)| \leq R_X(0)$ for all τ. We have assumed $X(t)$ is stationary and thus the mean-square values of $X(t)$ and $X(t + \tau)$ must be the same. Also the magnitude of the correlation coefficient relating two random variables is never greater than unity. Thus, $R_X(\tau)$ can never be greater in magnitude than $R_X(0)$.
4. If $X(t)$ contains a periodic component, $R_X(t)$ will also contain a periodic component with the same period. This can be verified by writing $X(t)$ as the sum of the nonperiodic and periodic components and then applying the definition given by Eq. (2.5.2). It is of interest to note that if the process is ergodic as well as stationary and if the periodic component is sinusoidal, then $R_X(\tau)$ will contain no information about the phase of the sinusoidal component. The harmonic component always appears in

the autocorrelation function as a cosine function, irrespective of its phase.

5. If $X(t)$ does not contain any periodic components, $R_X(\tau)$ tends to zero as $\tau \to \infty$. This is just a mathematical way of saying that $X(t + \tau)$ becomes completely uncorrelated with $X(t)$ for large τ if there are no hidden periodicities in the process. Note that a constant is a special case of a periodic function. Thus, $R_X(\infty) = 0$ implies zero mean for the process.
6. The Fourier transform of $R_X(\tau)$ is real, symmetric, and nonnegative. The real, symmetric property follows directly from the even property of $R_X(\tau)$. The nonnegative property is not obvious at this point. It will be justified later in Section 2.7, which deals with the spectral density function for the process.

It was mentioned previously that strict stationarity is a severe requirement, because it requires that all the higher-order probability density functions be invariant under a time translation. This is often difficult to verify. Thus, a less demanding form of stationarity is often used, or assumed. A random process is said to be *covariance stationary* or *wide-sense stationary* if $E[X(t_1)]$ is independent of t_1 and $E[X(t_1)X(t_2)]$ is dependent only on the time difference $t_2 - t_1$. Obviously, if the second-order density $f_{X_1X_2}(x_1, x_2)$ is independent of the time origin, the process is covariance stationary.

Further examples of autocorrelation functions will be given as this chapter progresses. We will see that the autocorrelation function is an important descriptor of a random process and one that is relatively easy to obtain because it depends on only the second-order probability density for the process.

2.6 CROSSCORRELATION FUNCTION

The crosscorrelation function between the processes $X(t)$ and $Y(t)$ is defined as

$$R_{XY}(t_1, t_2) = E[X(t_1)Y(t_2)] \tag{2.6.1}$$

Again, if the processes are stationary, only the time *difference* between sample points is relevant, so the crosscorrelation function reduces to

$$R_{XY}(\tau) = E[X(t)Y(t + \tau)] \quad \text{(stationary case)} \tag{2.6.2}$$

Just as the autocorrelation function tells us something about how a process is correlated with itself, the crosscorrelation function provides information about the mutual correlation between the two processes.

Notice that it is important to order the subscripts properly in writing $R_{XY}(\tau)$. A skew-symmetric relation exists for stationary processes as follows. By definition,

$$R_{XY}(\tau) = E[X(t)Y(t + \tau)] \tag{2.6.3}$$

$$R_{YX}(\tau) = E[Y(t)X(t + \tau)] \tag{2.6.4}$$

The expectation in Eq. (2.6.4) is invariant under a translation of $-\tau$. Thus, $R_{YX}(\tau)$ is also given by

$$R_{YX}(\tau) = E[Y(t - \tau)X(t)] \tag{2.6.5}$$

Now, comparing Eqs. (2.6.3) and (2.6.5), we see that

$$R_{XY}(\tau) = R_{YX}(-\tau) \tag{2.6.6}$$

Thus, interchanging the order of the subscripts of the crosscorrelation function has the effect of changing the sign of the argument.

EXAMPLE 2.8

Let $X(t)$ be the same random process of Example 2.6 and illustrated in Fig 2.7. Let $Y(t)$ be the same signal as $X(t)$, but delayed one-half unit of time. The crosscorrelation $R_{XY}(\tau)$ would then be shown in Fig. 2.9. Note that $R_{XY}(\tau)$ is not an even function of τ, nor does its maximum occur at $\tau = 0$. Thus, the crosscorrelation function lacks the symmetry possessed by the autocorrelation function. ▩

We frequently need to consider additive combinations of random processes. For example, let the process $Z(t)$ be the sum of stationary processes $X(t)$ and $Y(t)$:

$$Z(t) = X(t) + Y(t) \tag{2.6.7}$$

The autocorrelation function of the summed process is then

$$\begin{aligned} R_Z(\tau) &= E\{[X(t) + Y(t)][X(t + \tau) + Y(t + \tau)]\} \\ &= E[X(t)X(t + \tau)] + E[Y(t)X(t + \tau)] \\ &\quad + E[X(t)Y(t + \tau)] + E[Y(t)Y(t + \tau)] \\ &= R_X(\tau) + R_{YX}(\tau) + R_{XX}(\tau) + R_Y(\tau) \end{aligned} \tag{2.6.8}$$

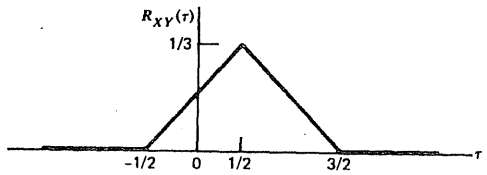


Figure 2.9 Crosscorrelation function for Example 2.8.

Now, if X and Y are zero-mean uncorrelated processes, the middle terms of Eq. (2.6.8) are zero, and we have

$$R_Z(\tau) = R_X(\tau) + R_Y(\tau) \quad \text{(for zero crosscorrelation)} \tag{2.6.9}$$

This can obviously be extended to the sum of more than two processes. Equation (2.6.9) is a much-used relationship, and it should always be remembered that it applies only when the processes being summed have zero crosscorrelation.

2.7 POWER SPECTRAL DENSITY FUNCTION

It was mentioned in Section 2.6 that the autocorrelation function is an important descriptor of a random process. Qualitatively, if the autocorrelation function decreases rapidly with τ, the process changes rapidly with time; conversely, a slowly changing process will have an autocorrelation function that decreases slowly with τ. Thus, we would suspect that this important descriptor contains information about the frequency content of the process; and this is in fact the case. For stationary processes, there is an important relation known as the *Wiener–Khinchine relation:*

$$S_X(j\omega) = \mathfrak{F}[R_X(\tau)] = \int_{-\infty}^{\infty} R_X(\tau)e^{-j\omega\tau}\,d\tau \tag{2.7.1}$$

where $\mathfrak{F}[\cdot]$ indicates Fourier transform and ω has the usual meaning of (2π) (frequency in hertz). S_X is called the *power spectral density function* or simply the *spectral density function* of the process.

The adjectives *power* and *spectral* come from the relationship of $S_X(j\omega)$ to the usual spectrum concept for a deterministic signal. However, some care is required in making this connection. If the process $X(t)$ is time stationary, it wanders on ad infinitum and is not absolutely integrable. Thus, the defining integral for the Fourier transform does not converge. When considering the Fourier transform of the process, we are forced to consider a truncated version of it, say, $X_T(t)$, which is truncated to zero outside a span of time T. The Fourier transform of a sample realization of the truncated process will then exist.

Let $\mathfrak{F}\{X_T\}$ denote the Fourier transform of $X_T(t)$, where it is understood that for any given ensemble of samples of $X_T(t)$ there will be corresponding ensemble of $\mathfrak{F}\{X_T(t)\}$. That is, $\mathfrak{F}\{X_T(t)\}$ has stochastic attributes just as does $X_T(t)$. Now look at the following expectation:

$$E\left[\frac{1}{T}|\mathfrak{F}\{X_T(t)\}|^2\right]$$

For any particular sample realization of $X_T(t)$, the quantity inside the brackets is known as the *periodogram* for that particular signal. It will now be shown that averaging over an ensemble of periodograms for large T yields the power spectral density function.

The expectation of the periodogram of a signal spanning the time interval $[0, T]$ can be manipulated as follows:

$$\begin{aligned} &E\left[\frac{1}{T}|\mathfrak{F}\{X_T(t)\}|^2\right] \\ &= E\left[\frac{1}{T}\int_0^T X(t)e^{-j\omega t}\,dt \int_0^T X(s)e^{j\omega s}\,ds\right] \\ &= \frac{1}{T}\int_0^T\int_0^T E[X(t)X(s)]e^{-j\omega(t-s)}\,dt\,ds \end{aligned} \tag{2.7.2}$$

Note that we were able to drop the subscript T on $X(t)$ because of the restricted range of integration. If we now assume $X(t)$ is stationary, $E[X(t)X(s)]$ becomes $R_X(t-s)$ and Eq. (2.7.2) becomes

$$E\left[\frac{1}{T}|\mathfrak{F}\{X_T(t)\}|^2\right] = \frac{1}{T}\int_0^T\int_0^T R_X(t-s)e^{-j\omega(t-s)}\,ds\,dt \tag{2.7.3}$$

The appearance of $t - s$ in two places in Eq. (2.7.3) suggests a change of variables. Let

$$\tau = t - s \tag{2.7.4}$$

Equation (2.7.3) then becomes

$$\frac{1}{T}\int_0^T\int_0^T R_X(t-s)e^{-j\omega(t-s)}\,ds\,dt = -\frac{1}{T}\int_0^T\int_t^{t-T} R_X(\tau)e^{-j\omega\tau}\,d\tau\,dt \tag{2.7.5}$$

The new region of integration in the τt plane is shown in Fig. 2.10.

Next we interchange the order of integration and integrate over the two triangular regions separately. This leads to

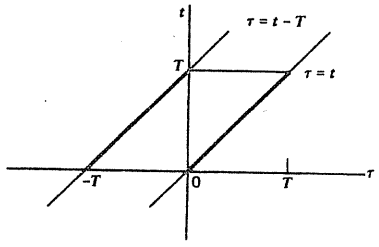


Figure 2.10 Region of integration in the τt plane.

$$E\left[\frac{1}{T}|\mathfrak{F}\{X_T(t)\}|^2\right] = \frac{1}{T}\int_{-T}^{0}\int_{0}^{\tau+T} R_X(\tau)e^{-j\omega\tau}\,dt\,d\tau + \frac{1}{T}\int_{0}^{T}\int_{\tau}^{T} R_X(\tau)e^{-j\omega\tau}\,dt\,d\tau \quad (2.7.6)$$

We now integrate with respect to t with the result

$$E\left[\frac{1}{T}|\mathfrak{F}\{X_T(t)\}|^2\right] = \frac{1}{T}\int_{-T}^{0}(\tau+T)R_X(\tau)e^{-j\omega\tau}\,d\tau + \frac{1}{T}\int_{0}^{T}(T-\tau)R_X(\tau)e^{-j\omega\tau}\,d\tau \quad (2.7.7)$$

Finally, Eq. (2.7.7) may be written in more compact form as

$$E\left[\frac{1}{T}|\mathfrak{F}\{X_T(t)\}|^2\right] = \int_{-T}^{T}\left(1-\frac{|\tau|}{T}\right)R_X(\tau)e^{-j\omega\tau}\,d\tau \quad (2.7.8)$$

The factor $1 - |\tau|/T$ that multiplies $R_X(\tau)$ may be thought of as a triangular weighting factor that approaches unity as T becomes large; at least this is true if $R_X(\tau)$ approaches zero as τ becomes large, which it will do if $X(t)$ contains no periodic components. Thus, as T becomes large, we have the following relationship:

$$E\left[\frac{1}{T}|\mathfrak{F}\{X_T(t)\}|^2\right] \Rightarrow \int_{-\infty}^{\infty} R_X(\tau)e^{-j\omega\tau}\,d\tau \qquad \text{as } T\to\infty \quad (2.7.9)$$

Or, in other words,

Average periodogram for large $T \Rightarrow$ power spectral density (2.7.10)

Note especially the "for large T" qualification in Eq. (2.7.10). (This is pursued further in Section 2.15 and Problem 2.34.)

Equation (2.7.9) is a most important relationship, because it is this that ties the spectral function $S_X(j\omega)$ to "spectrum" as thought of in the usual deterministic sense. Remember that the spectral density function, as formally defined by Eq. (2.7.1), is a probabilistic concept. On the other hand, the periodogram is a spectral concept in the usual sense of being related to the Fourier transform of a time signal. The relationship given by Eq. (2.7.9) then provides the tie between the probabilistic and spectral descriptions of the process, and it is this equation that suggests the name for $S_X(j\omega)$, power *spectral* density function. More will be said of this in Section 2.14, which deals with the determination of the spectral function from experimental data.

Because of the spectral attributes of the autocorrelation function $R_X(\tau)$, its Fourier transform $S_X(j\omega)$ always works out to be a real, nonnegative, symmetric function of ω. This should be apparent from the left side of Eq. (2.7.9), and will be illustrated in Example 2.9.

EXAMPLE 2.9

Consider a random process $X(t)$ whose autocorrelation function is given by

$$R_X(\tau) = \sigma^2 e^{-\beta|\tau|} \quad (2.7.11)$$

where σ^2 and β are known constants. The spectral density function for the $X(t)$ process is

$$S_X(j\omega) = \mathfrak{F}[R_X(\tau)] = \frac{\sigma^2}{j\omega+\beta} + \frac{\sigma^2}{-j\omega+\beta} = \frac{2\sigma^2\beta}{\omega^2+\beta^2} \quad (2.7.12)$$

Both R_X and S_X are sketched in Fig. 2.11. ■

Occasionally, it is convenient to write the spectral density function in terms of the complex frequency variable s rather than ω. This is done by simply replacing $j\omega$ with s; or, equivalently, replacing ω^2 with $-s^2$. For Example 2.9, the spectral density function in terms of s is then

$$S_X(s) = \left.\frac{2\sigma^2\beta}{\omega^2+\beta^2}\right|_{\omega^2=-s^2} = \frac{2\sigma^2\beta}{-s^2+\beta^2} \quad (2.7.13)$$

It should be clear now why we chose to include the "j" with ω in $S_X(j\omega)$, even though $S_X(j\omega)$ always works out to be a real function of ω. By writing the argument of $S_X(j\omega)$ as $j\omega$, rather than just ω, we can use the same symbol for spectral function in either the complex or real frequency domain. That is,

$$S_X(s) = S_X(j\omega) \quad (2.7.14)$$

is correct notation in the usual mathematical sense.

From Fourier transform theory, we know that the inverse transform of the spectral function should yield the autocorrelation function, that is,

$$\mathfrak{F}^{-1}[S_X(j\omega)] = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_X(j\omega)e^{j\omega\tau}\,d\omega = R_X(\tau) \quad (2.7.15)$$

If we let $\tau = 0$ in Eq. (2.7.15), we get

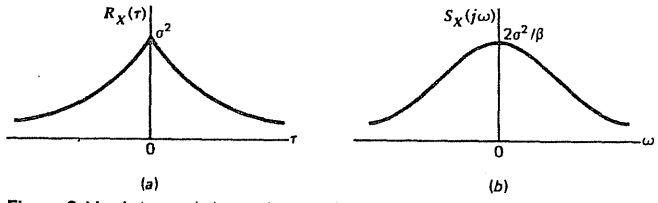


Figure 2.11 Autocorrelation and spectral density functions for Example 2.9. (*a*) Autocorrelation function. (*b*) Spectral function.

$$R_X(0) = E[X^2(t)] = \frac{1}{2\pi}\int_{-\infty}^{\infty} S_X(j\omega)\,d\omega \tag{2.7.16}$$

Equation (2.7.16) provides a convenient means of computing the mean square value of a stationary process, given its spectral function. As mentioned before, it is sometimes convenient to use the complex frequency variable s rather than $j\omega$. If this is done, Eq. (2.7.16) becomes

$$E[X^2(t)] = \frac{1}{2\pi j}\int_{-j\infty}^{j\infty} S_X(s)\,ds \tag{2.7.17}$$

Equation (2.7.16) suggests that we can consider the signal power as being distributed in frequency in accordance with $S_X(j\omega)$, thus, the terms *power* and *density* in power spectral density function. Using this concept, we can obtain the power in a finite band by integrating over the appropriate range of frequencies, that is,

$$\left[\begin{array}{c}\text{"Power" in}\\ \text{range } \omega_1 \le \omega \le \omega_2\end{array}\right] = \frac{1}{2\pi}\int_{-\omega_2}^{-\omega_1} S_X(j\omega)\,d\omega + \frac{1}{2\pi}\int_{\omega_1}^{\omega_2} S_X(j\omega)\,d\omega \tag{2.7.18}$$

An example will now be given to illustrate the use of Eqs. (2.7.16) and (2.7.17).

EXAMPLE 2.10

Consider the spectral function of Example 2.9:

$$S_X(j\omega) = \frac{2\sigma^2\beta}{\omega^2 + \beta^2} \tag{2.7.19}$$

Application of Eq. (2.7.16) should yield the mean square value σ^2. This can be verified using conventional integral tables.

$$E(X^2) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{2\sigma^2\beta}{\omega^2+\beta^2}\,d\omega = \frac{\sigma^2\beta}{\pi}\left[\frac{1}{\beta}\tan^{-1}\frac{\omega}{\beta}\right]_{-\infty}^{\infty} = \sigma^2 \tag{2.7.20}$$

Or equivalently, in terms of s:

$$E(X^2) = \frac{1}{2\pi j}\int_{-j\infty}^{j\infty}\frac{2\sigma^2\beta}{-s^2+\beta^2}\,ds = \sigma^2 \tag{2.7.21}$$

More will be said about evaluating integrals of the type in Eq. (2.7.21) later, in Chapter 3. ■

In summary, we see that the autocorrelation function and spectral density function are Fourier transform pairs. Thus, both contain the same basic information about the process, but in different forms. Since we can easily transform back and forth between the time and frequency domains, the manner in which the information is presented is purely a matter of convenience for the problem at hand.

2.8 CROSS SPECTRAL DENSITY FUNCTION

Cross spectral density functions for stationary processes $X(t)$ and $Y(t)$ are defined as

$$S_{XY}(j\omega) = \mathfrak{F}[R_{XY}(\tau)] = \int_{-\infty}^{\infty} R_{XY}(\tau)e^{-j\omega\tau}\,d\tau \tag{2.8.1}$$

$$S_{YX}(j\omega) = \mathfrak{F}[R_{YX}(\tau)] = \int_{-\infty}^{\infty} R_{YX}(\tau)e^{-j\omega\tau}\,d\tau \tag{2.8.2}$$

The crosscorrelation functions $R_{XY}(\tau)$ and $R_{YX}(\tau)$ are not necessarily even functions of τ, and thus the corresponding cross spectral densities are usually not real functions of ω. It was noted in Section 2.6 that $R_{XY}(\tau) = R_{YX}(-\tau)$. Thus, S_{XY} and S_{YX} are complex conjugates of each other:

$$S_{XY}(j\omega) = S_{YX}^*(j\omega) \tag{2.8.3}$$

and the sum of S_{XY} and S_{YX} is real.

Another function that is closely related to the cross spectral density is the *coherence function*. It is defined as

$$\gamma_{XY}^2 = \frac{|S_{XY}(j\omega)|^2}{S_X(j\omega)S_Y(j\omega)} \tag{2.8.4}$$

The coherence function can be seen to be normalized, and it is sort of a "correlation coefficient in the frequency domain." To see the normalization, let $X(t) = Y(t)$ (maximum correlation) and then

$$\gamma_{XX}^2 = \frac{|S_{XX}(j\omega)|^2}{S_X(j\omega)S_X(j\omega)} = 1 \tag{2.8.5}$$

On the other extreme, if $X(t)$ and $Y(t)$ have zero crosscorrelation, $S_{XY}(j\omega) = 0$ and $\gamma_{XY}^2 = 0$. Both the cross spectral density and coherence functions are useful in analysis of experimental data, because modern computer technology has made it possible to transform time data to the frequency domain with ease. (See Section 2.15 and references 5 and 6 for more on the analysis of experimental data. Also, see Problems 3.23 and 3.25.)

If $Z(t)$ is the sum of zero-mean processes $X(t)$ and $Y(t)$, the spectral density of $Z(t)$ is given by

$$S_Z(j\omega) = \mathfrak{F}[R_{X+Y}(\tau)] \tag{2.8.6}$$

Referring to Eq. (2.6.8), we then have

$$S_Z(j\omega) = S_X(j\omega) + S_{YX}(j\omega) + S_{XY}(j\omega) + S_Y(j\omega) \tag{2.8.7}$$

Just as in the case of the autocorrelation function, the two middle terms in Eq. (2.8.7) are zero if the X and Y processes have zero crosscorrelation. So, for this special case,

$$S_{X+Y}(j\omega) = S_X(j\omega) + S_Y(j\omega) \quad \text{(for zero crosscorrelation)} \tag{2.8.8}$$

2.9 WHITE NOISE

White noise is defined to be a stationary random process having a constant spectral density function. The term "white" is a carryover from optics, where white light is light containing all visible frequencies. Denoting the white-noise spectral amplitude as A, we then have

$$S_{wn}(j\omega) = A \tag{2.9.1}$$

The corresponding autocorrelation function for white noise is then

$$R_{wn}(\tau) = A\delta(\tau) \tag{2.9.2}$$

These functions are sketched in Fig. 2.12.

In analysis, one frequently makes simplifying assumptions in order to make the problem mathematically tractable. White noise is a good example of this. However, by assuming the spectral amplitude of white noise to be constant for all frequencies (for the sake of mathematical simplicity), we find ourselves in the awkward situation of having defined a process with infinite variance. Qualitatively, white noise is sometimes characterized as noise that is jumping around infinitely far, infinitely fast! This is obviously physical nonsense but it is a useful

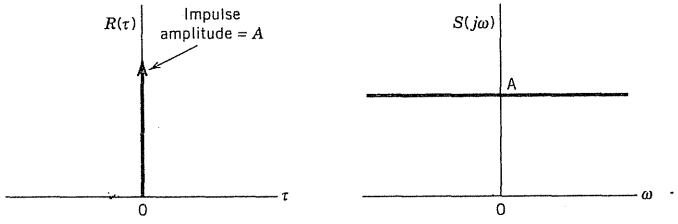


Figure 2.12 White noise. (*a*) Autocorrelation function. (*b*) Spectral density function.

abstraction. The saving feature is that all physical systems are bandlimited to some extent, and a bandlimited system driven by white noise yields a process that has finite variance; that is, the end result makes sense. We will elaborate on this further in Chapter 3.

Bandlimited white noise is a random process whose spectral amplitude is constant over a finite range of frequencies, and zero outside that range. If the bandwidth includes the origin (sometimes called baseband), we then have

$$S_{bwn}(j\omega) = \begin{cases} A, & |\omega| \le 2\pi W \\ 0, & |\omega| > 2\pi W \end{cases} \tag{2.9.3}$$

where W is the physical bandwidth in hertz. The corresponding autocorrelation function is

$$R_{bwn}(\tau) = 2WA\,\frac{\sin(2\pi W\tau)}{2\pi W\tau} \tag{2.9.4}$$

Both the autocorrelation and spectral density functions for baseband bandlimited white noise are sketched in Fig. 2.13. It is of interest to note that the autocorrelation function for baseband bandlimited white noise is zero for $\tau = 1/2W$, $2/2W$, $3/2W$, etc. From this we see that if the process is sampled at a rate of $2W$ samples/second (sometimes called the Nyquist rate), the resulting set of random variables are uncorrelated. Since this usually simplifies the analysis, the white bandlimited assumption is frequently made in bandlimited situations.

The frequency band for bandlimited white noise is sometimes offset from the origin and centered about some center frequency W_0. It is easily verified that the autocorrelation/spectral-function pair for this situation is

$$S(j\omega) = \begin{cases} A, & 2\pi W_1 \le |\omega| \le 2\pi W_2 \\ 0, & |\omega| < 2\pi W_1 \quad \text{and} \quad |\omega| > 2\pi W_2 \end{cases} \tag{2.9.5}$$

$$R(\tau) = A\left[2W_2\,\frac{\sin 2\pi W_2\tau}{2\pi W_2\tau} - 2W_1\,\frac{\sin 2\pi W_1\tau}{2\pi W_1\tau}\right]$$

$$= A2\,\Delta W\,\frac{\sin \pi\Delta W\tau}{\pi\Delta W\tau}\cos 2\pi W_0\tau \tag{2.9.6}$$

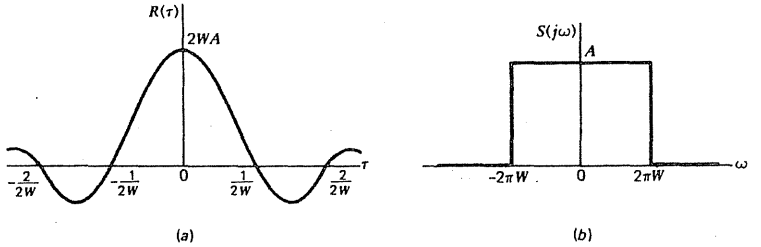


Figure 2.13 Baseband bandlimited white noise. (*a*) Autocorrelation function. (*b*) Spectral density function.

where

$$\Delta W = W_2 - W_1 \text{ Hz}$$

$$W_0 = \frac{W_1 + W_2}{2} \text{ Hz}$$

These functions are sketched in Fig. 2.14.

It is worth noting the bandlimited white noise has a finite mean-square value, and thus it is physically plausible, whereas pure white noise is not. However, the mathematical forms for the autocorrelation and spectral functions in the bandlimited case are more complicated than for pure white noise.

Before leaving the subject of white noise, it is worth mentioning that the analogous discrete-time process is referred to as a white sequence. A *white sequence* is defined simply as a sequence of zero-mean, uncorrelated random variables. That is, all members of the sequence have zero means and are mutually uncorrelated with all other members of the sequence. If the random variables are also normal, then the sequence is a *Gaussian* white sequence.

2.10 GAUSS–MARKOV PROCESS

A stationary Gaussian process $X(t)$ that has an exponential autocorrelation is called a *Gauss–Markov* process. The autocorrelation and spectral functions for this process are then of the form

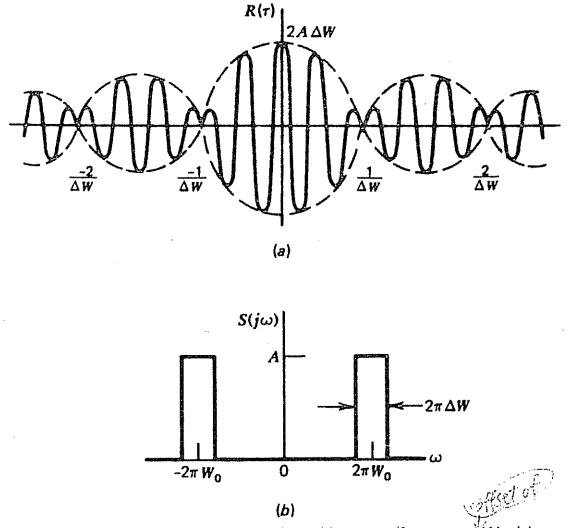


Figure 2.14 Bandlimited white noise with center frequency W_0. (*a*) Autocorrelation function. (*b*) Spectral density.

$$R_X(\tau) = \sigma^2 e^{-\beta|\tau|} \tag{2.10.1}$$

$$S_X(j\omega) = \frac{2\sigma^2\beta}{\omega^2 + \beta^2} \quad \left[\text{or } S_X(s) = \frac{2\sigma^2\beta}{-s^2 + \beta^2}\right] \tag{2.10.2}$$

These functions are sketched in Fig. 2.15. The mean-square value and time constant for the process are given by the σ^2 and $1/\beta$ parameters, respectively. The process is nondeterministic, so a typical sample time function would show no deterministic structure and would look like typical "noise." The exponential autocorrelation function indicates that sample values of the process gradually become less and less correlated as the time separation between samples increases. The autocorrelation function approaches zero as $\tau \to \infty$, and thus the mean value of the process must be zero. The reference to Markov in the name of this process is not obvious at this point, but it will be after the discussion on optimal prediction in Chapter 4.

The Gauss–Markov process is an important process in applied work because (1) it seems to fit a large number of physical processes with reasonable accuracy, and (2) it has a relatively simple mathematical description. As in the case of all stationary Gaussian processes, *specification of the process autocorrelation function completely defines the process*. This means that any desired higher-order probability density function for the process may be written out explicitly, given the autocorrelation function. An example will illustrate this.

EXAMPLE 2.11

Let us say that a Gauss–Markov process $X(t)$ has autocorrelation function

$$R_X(\tau) = 100e^{-2|\tau|} \tag{2.10.3}$$

We wish to write out the third-order probability density function

$$f_{X_1X_2X_3}(x_1, x_2, x_3) \qquad \text{where } X_1 = X(0),\ X_2 = X(.5), \text{ and } X_3 = X(1)$$

First we note that the process mean is zero. The covariance matrix in this case is a 3×3 matrix and is obtained as follows:

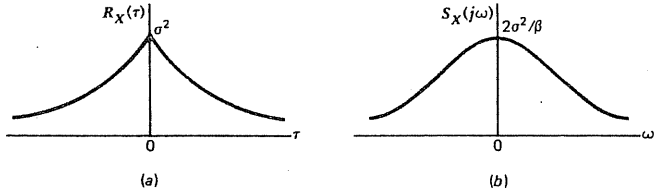


Figure 2.15 Autocorrelation and spectral density functions for Gauss–Markov process. (*a*) Autocorrelation function. (*b*) Spectral density.

$$\mathbf{C}_X = \begin{bmatrix} E(X_1^2) & E(X_1X_2) & E(X_1X_3) \\ E(X_2X_1) & E(X_2^2) & E(X_2X_3) \\ E(X_3X_1) & E(X_3X_2) & E(X_3^2) \end{bmatrix} = \begin{bmatrix} R_X(0) & R_X(.5) & R_X(1) \\ R_X(.5) & R_X(0) & R_X(.5) \\ R_X(1) & R_X(.5) & R_X(0) \end{bmatrix}$$

$$= \begin{bmatrix} 100 & 100e^{-1} & 100e^{-2} \\ 100e^{-1} & 100 & 100e^{-1} \\ 100e^{-2} & 100e^{-1} & 100 \end{bmatrix} \tag{2.10.4}$$

Now that $\mathbf{C}_X$ has been written out explicitly, we can use the general normal form given by Eq. (1.15.5). The desired density function is then

$$f_X(\mathbf{x}) = \frac{1}{(2\pi)^{3/2}|\mathbf{C}_X|^{1/2}} e^{-\frac{1}{2}[\mathbf{x}^T\mathbf{C}_X^{-1}\mathbf{x}]} \tag{2.10.5}$$

where $\mathbf{x}$ is the 3-tuple

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \tag{2.10.6}$$

and $\mathbf{C}_x$ is given by Eq. (2.10.4). ■

The simple scalar Gauss–Markov process whose autocorrelation function is exponential is sometimes referred to as a first-order Gauss–Markov process. This is because the discrete-time version of the process is described by a first-order difference equation of the form

$$X(t_{k+1}) = e^{-\beta\Delta t}X(t_k) + W(t_k)$$

where $W(t_k)$ is an uncorrelated zero-mean Gaussian sequence. Discrete-time Gaussian processes that satisfy higher-order difference equations are also often referred to as Gauss–Markov processes of the appropriate order. Such processes are best described in vector form, and this is discussed in detail in Sections 5.2 and 5.3. Also, an example of a second-order Gauss–Markov process that is of some importance in satellite navigation systems is discussed in Problem 5.4 and Example 6.1. (p. 233)

2.11 RANDOM TELEGRAPH WAVE

Consider a binary voltage waveform that is generated according to the following rules:

1. The voltage is either +1 or −1 V.
2. The state at $t = 0$ may be either +1 or −1 V with equal likelihood.
3. As time progresses, the voltage switches from one state to the other at random. Specifically, the probability of k switches in a time interval T

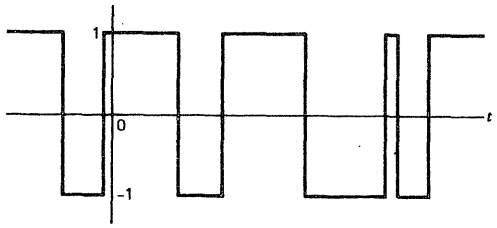


Figure 2.16 Random telegraph wave.

is governed by the Poisson distribution

$$P(k) = \frac{(aT)^k}{k!} e^{-aT} \tag{2.11.1}$$

where a is the average number of switches per unit time.

This random process is called the random telegraph wave, and a sample waveform is shown in Fig. 2.16. It is worth noting that the likelihood of switching at any point in time does not depend on the particular state or the length of time the system has been in that state.

A rigorous derivation of the autocorrelation function for the random telegraph wave can be found in a number of references (7, 8). For our purposes here, we can use the following heuristic argument. Consider the product of successive time samples $X(t_1)$ and $X(t_2)$; the result must be either of two possibilities, +1 or −1. If the time interval $t_2 - t_1$ is small, $X(t_1)$ and $X(t_2)$ are highly correlated, so that $X_1(t_1)X(t_2)$ is nearly unity. Then, as the spacing between samples is increased, their correlation gradually decreases and approaches zero as the spacing between samples goes to infinity. This leads to an exponential autocorrelation function of the form

$$R_X(\tau) = e^{-2a|\tau|} \tag{2.11.2}$$

where a is the average number of switches per unit time.

It should be apparent that the random telegraph wave is not a Gaussian process—far from it! Yet the Gauss–Markov process described in Section 2.10 has an autocorrelation function identical in form to that of the random telegraph wave. For the purposes of comparison, a typical Gauss–Markov signal with about the same time constant as the random telegraph wave of Fig. 2.16 is shown in Fig. 2.17. The difference is striking. This is a vivid example of two processes

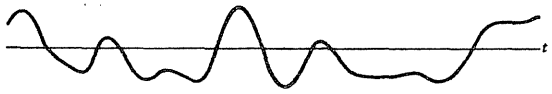


Figure 2.17 Gauss–Markov signal with about the same time constant as for the random telegraph wave of Figure 2.16.

that have radically different random structures, but still have the same autocorrelation functions and, of course, identical spectral characteristics. The moral is this: The autocorrelation function and/or spectral characteristics do not tell the whole story; all the probability density functions must be specified in order for the process to be completely described. In the Gauss–Markov case, this was done by specifying "Gaussian process" in addition to the autocorrelation function. In the random telegraph wave case, the higher-order densities were indirectly specified by describing a conceptual chance experiment creating the waveform. Obviously, the probability density functions are radically different for the two processes in this case.

2.12 NARROWBAND GAUSSIAN PROCESS

In both control and communication systems, we frequently encounter situations where a very narrowband system is excited by wideband Gaussian noise. A high-Q tuned circuit and/or a lightly damped mass–spring arrangement are examples of narrowband systems. The resulting output is a noise process with essentially all its spectral content concentrated in a narrow frequency range. If one were to observe the output of such a system, the time function would appear to be nearly sinusoidal, especially if just a few cycles of the output signal were observed. However, if one were to carefully examine a long record of the signal, it would be seen that the quasi-sinusoid is slowly varying in both amplitude and phase. Such a signal is called narrowband noise and, if it is the result of passing wideband Gaussian noise through a linear narrowband system, then it is also Gaussian. We are assured of this because any linear operation on a set of normal variates results in another set of normal variates. The quasi-sinusoidal character depends only on the narrowband property, and the exact spectral shape within the band is immaterial.

The mathematical description of narrowband Gaussian noise follows. We first write the narrowband signal as

$$S(t) = X(t) \cos \omega_c t - Y(t) \sin \omega_c t \tag{2.12.1}$$

where $X(t)$ and $Y(t)$ are independent Gaussian random processes with similar narrowband spectral functions that are centered about zero frequency. The frequency ω_c is usually called the carrier frequency, and the effect of multiplying $X(t)$ and $Y(t)$ by $\cos \omega_c t$ and $\sin \omega_c t$ is to translate the baseband spectrum up to a similar spectrum centered about ω_c (see Problem 2.32). The independent $X(t)$ and $Y(t)$ processes are frequently called the in-phase and quadrature components of $S(t)$. Now, think of time t as a particular time, and think of $X(t)$ and $Y(t)$ as the corresponding random variables. Then make the usual rectangular to polar transformation via the equations

$$\begin{aligned} X &= R \cos \Theta \\ Y &= R \sin \Theta \end{aligned} \tag{2.12.2}$$

or, equivalently,

$$\begin{aligned} R &= \sqrt{X^2 + Y^2} \\ \Theta &= \tan^{-1} \frac{Y}{X} \end{aligned} \tag{2.12.3}$$

By substituting Eq. (2.12.2) into Eq. (2.12.1), we can now write $S(t)$ in the form

$$\begin{aligned} S(t) &= R(t) \cos \Theta(t) \cos \omega_c t - R(t) \sin \Theta(t) \sin \omega_c t \\ &= R(t) \cos[\omega_c t + \Theta(t)] \end{aligned} \tag{2.12.4}$$

It is from Eq. (2.12.4) that we get the physical interpretation of "slowly varying envelope (amplitude) and phase."

Before we proceed, a word or two about the probability densities for X, Y, R, and Θ is in order. If X and Y are independent normal random variables with the same variance σ^2, their individual and joint densities are

$$f_X(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-x^2/2\sigma^2} \tag{2.12.5}$$

$$f_Y(y) = \frac{1}{\sqrt{2\pi\sigma}} e^{-y^2/2\sigma^2} \tag{2.12.6}$$

and

$$f_{XY}(x, y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{2.12.7}$$

The corresponding densities for R and Θ are Rayleigh and uniform (see Example 1.23). The mathematical forms are

$$f_R(r) = \frac{r}{\sigma^2} e^{-r^2/2\sigma^2}, \qquad r \geq 0 \quad \text{(Rayleigh)} \tag{2.12.8}$$

$$f_\Theta(\theta) = \begin{cases} \dfrac{1}{2\pi}, & 0 \leq \theta < 2\pi \\ 0, & \text{otherwise} \end{cases} \quad \text{(uniform)} \tag{2.12.9}$$

Also, the joint density function for R and Θ is

$$f_{R\Theta}(r, \theta) = \frac{r}{2\pi\sigma^2} e^{-r^2/2\sigma^2}, \qquad r \geq 0 \quad \text{and} \quad 0 \leq \theta < 2\pi \tag{2.12.10}$$

It is of interest to note here that if we consider simultaneous time samples of envelope and phase, the resulting random variables are statistically independent. However, the *processes* $R(t)$ and $\Theta(t)$ are not statistically independent (7). This is due to the fact that the joint probability density associated with adjacent samples cannot be written in product form, that is,

$$f_{R_1R_2\Theta_1\Theta_2}(r_1, r_2, \theta_1, \theta_2) \neq f_{R_1R_2}(r_1, r_2) f_{\Theta_1\Theta_2}(\theta_1, \theta_2) \tag{2.12.11}$$

We have assumed that $S(t)$ is a Gaussian process, and from Eq. (2.12.1) we see that

$$\text{Var } S = \tfrac{1}{2} (\text{Var } X) + \tfrac{1}{2} (\text{Var } Y) = \sigma^2 \tag{2.12.12}$$

Thus,

$$f_S(s) = \frac{1}{\sqrt{2\pi}\sigma} e^{-s^2/2\sigma^2} \tag{2.12.13}$$

The higher-order density functions for S will, of course, depend on the specific shape of the spectral density for the process.

2.13 WIENER OR BROWNIAN-MOTION PROCESS

Suppose we start at the origin and take n steps forward or backward at random, with equal likelihood of stepping in either direction. We pose two questions: After taking n steps, (1) what is the average distance traveled, and (2) what is the variance of the distance? This is the classical random-walk problem of statistics. The averages considered here must be taken in an ensemble sense; for example, think of running simultaneous experiments and then averaging the results for a given number of steps. It should be apparent that the average distance traveled is zero, provided we say "forward" is positive and "backward" is negative. However, the square of the distance is always positive (or zero), so its average for a large number of trials will not be zero. It is shown in elementary statistics that the variance after n unit steps is just n, or the standard deviation is $\sqrt{n}$ (see Problem 2.21). Note that this increases without bound as n increases, and thus the process is nonstationary.

The continuous analog of random-walk is the output of an integrator driven with white noise. This is shown in block-diagram form in Fig. 2.18*a*. Here we consider the input switch as closing at $t = 0$ and the initial integrator output as

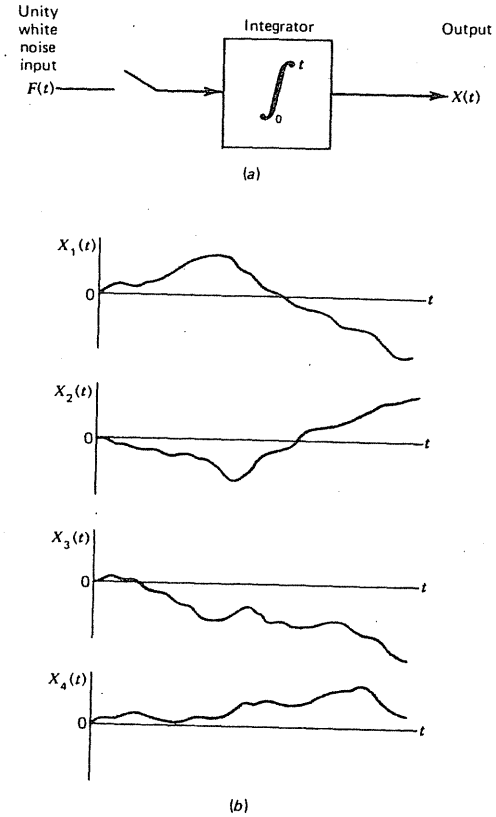


Figure 2.18 Continuous analog of random walk. (*a*) Block diagram. (*b*) Ensemble of output signals.

being zero. An ensemble of typical output time signals is shown in Fig. 2.18*b*. The system response $X(t)$ is given by

$$X(t) = \int_0^t F(u)\, du \tag{2.13.1}$$

Clearly, the average of the output is

$$E[X(t)] = E\left[\int_0^t F(u)\, du\right] = \int_0^t E[F(u)]\, du = 0 \tag{2.13.2}$$

Also, the mean-square-value (variance) is

$$E[X^2(t)] = E\left[\int_0^t F(u)\,du \int_0^t F(v)\,dv\right] = \int_0^t \int_0^t E[F(u)F(v)]\,du\,dv \quad (2.13.3)$$

But $E[F(u)F(v)]$ is just the autocorrelation function $R_F(u - v)$, which in this case is a Dirac delta function. Thus,

$$E[X^2(t)] = \int_0^t \int_0^t \delta(u - v)\,du\,dv = \int_0^t dv = t \quad (2.13.4)$$

So, $E[X^2(t)]$ increases linearly with time and the rms value increases in accordance with $\sqrt{t}$ (for unity white noise input). (Problem 2.35 provides a demonstration of this.)

Now, add the further requirement that the input be *Gaussian* white noise. The output will then be a Gaussian process because integration is a linear operation on the input. The resulting continuous random-walk process is known as the *Wiener* or *Brownian-motion* process. The process is nonstationary, it is Gaussian, and its mean, mean-square value, and autocorrelation function are given by

$$E[X(t)] = 0 \quad (2.13.5)$$

$$E[X^2(t)] = t \quad (2.13.6)$$

$$R_X(t_1, t_2) = E[X(t_1)X(t_2)] = E\left[\int_0^{t_1} F(u)\,du \cdot \int_0^{t_2} F(v)\,dv\right]$$

$$= \int_0^{t_2} \int_0^{t_1} E[F(u)F(v)]\,du\,dv = \int_0^{t_2} \int_0^{t_1} \delta(u - v)\,du\,dv$$

Evaluation of the double integral yields

$$R_X(t_1, t_2) = \begin{cases} t_2, & t_1 \geq t_2 \\ t_1, & t_1 < t_2 \end{cases} \quad (2.13.7)$$

Since the process is nonstationary, the autocorrelation function is a general function of the two arguments t_1 and t_2. With a little imagination, Eq. (2.13.7) can be seen to describe two faces of a pyramid with the sloping ridge of the pyramid running along the line $t_1 = t_2$.

It was mentioned before that there are difficulties in defining directly what is meant by Gaussian white noise. This is because of the "infinite variance" problem. The Wiener process is well behaved, though. Thus, we can reverse the argument given here and begin by arbitrarily defining it as a Gaussian process with an autocorrelation function given by Eq. (2.13.7). This completely specifies the process. We can now describe Gaussian white noise in terms of its integral. That is, Gaussian white noise is that hypothetical process which, when integrated, yields a Wiener process.

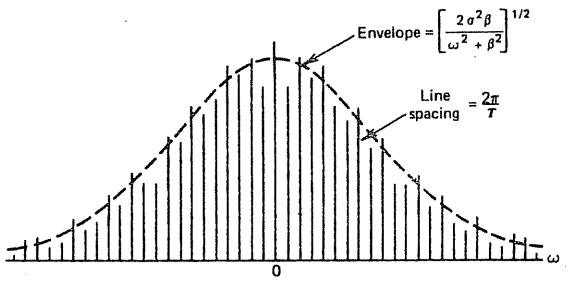


Figure 2.19 Spectrum of noise sample looped back on itself.

2.14 PSEUDORANDOM SIGNALS

As the name implies, pseudorandom signals have the appearance of being random, but are not truly random. In order for a signal to be truly random, there must be some uncertainty about it that is governed by chance. Pseudorandom signals do not have this "chance" property. Two examples of pseudorandom signals will now be presented.

EXAMPLE 2.12

Consider a sample realization of finite length T of a Gauss–Markov process. Let the time length T of the sample be large relative to the time constant of the process. After the sample is taken, of course, the time function in all its intimate detail is known to the observer. After the fact, nothing remains to chance insofar as the observer in concerned. Next, imagine folding this record back on itself into a single loop (it might be on magnetic tape), and then imagine playing the loop continuously. It should be clear that the resulting signal would be periodic and completely known (determined), at least to the original observer. Yet to a second observer casually looking at a small portion of the loop, the signal would *appear* to be just random noise. It should be mentioned that this is not a completely hypothetical situation; experimental spectral analysis was frequently implemented in just this way in times prior to the modern on-line digital methods.

The "looped" signal that goes on ad infinitum is periodic, so it would have line type rather than continuous spectral characteristics. See Fig. 2.19. ▩

Line-type spectra are characteristic of all pseudorandom signals. The line spacing may be extremely small, as is the case for very large T, but it is there, nevertheless. Note that the envelope of the lines would approximate the square root of the spectral density of the process from which the sample was taken, provided the record length T is large.* In this case, the usual laboratory analog

* Strictly speaking, the *average* envelope would approximate the square root of the spectral density (see Section 2.15).

spectrum analyzer would not be able to resolve individual lines, and it would indicate a smooth spectrum proportional to the average spectrum. This would be just fine and what was desired if the original experimental problem was to determine the spectral characteristics of random process from a single long sample of the process. The point of all this is that the typical analog spectrum analyzer could not distinguish between pseudorandom noise and true random noise if the line spacing for the pseudorandom noise is very small.

EXAMPLE 2.13

Binary sequences generated by shift registers with feedback have periodic properties and have found extensive application in ranging and communication systems (9, 10, 11, 12). We will use the simple 5-bit shift register shown in Fig. 2.20 to demonstrate how a pseudorandom binary signal can be generated. In this system the bits are shifted to the right with each clock pulse, and the input on the left is determined by the feedback arrangement. For the initial condition shown, it can be verified that the output sequence is

$$\underbrace{1111100011011010100001001011 00}_{31 \text{ bits}}\ \underbrace{1111100.....}_{\text{same 31 bits}}\ \underbrace{\quad}_{\text{etc.}}$$

Note that the sequence repeats itself after 31 bits. This periodic property is characteristic of a shift register with feedback. The maximum length of the sequence (before repetition) is given by $(2^n - 1)$, where n is the register length (9). The 5-bit example used here is then a maximum-length sequence. These sequences are especially interesting because of their pseudorandom appearance. Note that there are nearly the same number of zeros and ones in the sequence (16 ones and 15 zeros), and that they appear to occur more or less at random. If we consider a longer shift register, say, one with 10 bits, its maximum-length sequence would be 1023 bits. It would have 512 ones and 511 zeros; again, these would appear to be distributed at random. A casual look at any interior string of bits would not reveal anything systematic. Yet the string of bits so generated is entirely deterministic. Once the feedback arrangement and initial condition are specified, the output is determined forever after. Thus, the sequence is pseudorandom, not random. ■

When converted to voltage waveforms, maximum-length sequences also have special autocorrelation properties. Returning to the 31-bit example, let bi-

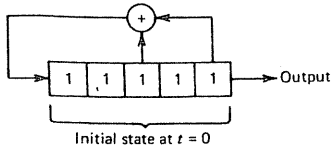


Figure 2.20 Binary shift register with feedback.

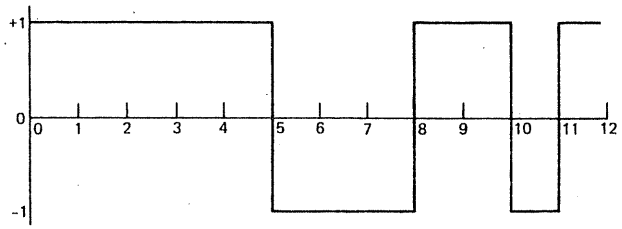


Figure 2.21 Pseudorandom binary waveform.

nary one be 1 V and binary zero be −1 V, and let the voltage level be held constant during the clock-pulse interval. The resulting time waveform is shown in Fig. 2.21, and its time autocorrelation function is shown in Fig. 2.22. Note that the unique distribution of zeros and ones for this sequence is such that the autocorrelation function is a small constant value after a shift of one unit of time (i.e., one bit). This is typical of all maximum-length sequences. When the sequence length is long, the correlation after a shift of one unit is quite small. This has obvious advantages in correlation detection schemes, and such schemes have been used extensively in electronic ranging applications (10, 11).

The spectral density function for the waveform of Fig. 2.21 is shown in Fig. 2.23. As with all pseudorandom signals, the spectrum is line-type rather than continuous (9, 12).

2.15 DETERMINATION OF AUTOCORRELATION AND SPECTRAL DENSITY FUNCTIONS FROM EXPERIMENTAL DATA

The determination of spectral characteristics of a random process from experimental data is a common engineering problem. All of the optimization techniques presented in the following chapters depend on prior knowledge of the spectral density of the processes involved. Thus, the designer needs this infor-

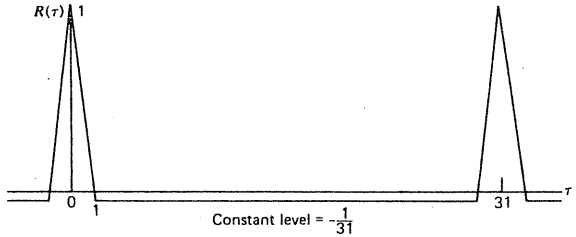


Figure 2.22 Time autocorrelation function for waveform of Figure 2.21.

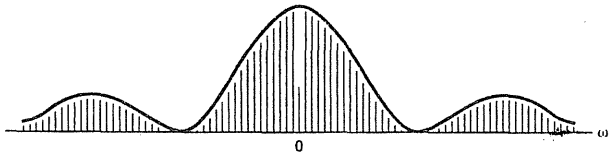


Figure 2.23 Spectral density for pseudorandom binary waveform.

mation and it usually must come from experimental evidence. Spectral determination is a relatively complicated problem with many pitfalls, and one should approach it with a good deal of caution. It is closely related to the larger problem of digital data processing, because the amount of data needed is usually large, and processing it either manually or in analog form is often not feasible. We first consider the span of observation time of the experimental data, which is a fundamental limitation, irrespective of the means of processing the data.

The time span of the data to be analyzed must, of course, be finite; and, as a practical matter, we prefer not to analyze any more data than is necessary to achieve reasonable results. Remember that since this is a matter of statistical inference, there will always remain some statistical uncertainty in the result. One way to specify the accuracy of the experimentally determined spectrum or autocorrelation function is to say that its variance must be less than a specified value. General accuracy bounds applicable to all processes are not available but there is one special case, the Gaussian process, that is amenable to analysis. We will not give the proof here, but it is shown in a number of references (5, 13) that the variance of an experimentally determined autocorrelation function satisfies the inequality

$$\text{Var } V_X(\tau) \leq \frac{4}{T}\int_0^\infty R_X^2(\tau)\, d\tau \tag{2.15.1}$$

where it is assumed that a single sample realization of the process is being analyzed, and

T = time length of the experimental record
$R_X(\tau)$ = autocorrelation function of the Gaussian process under consideration
$V_X(\tau)$ = time average of $X_T(t)X_T(t+\tau)$ where $X_T(t)$ is the finite-length sample of $X(t)$ [i.e., $V_X(\tau)$ is the experimentally determined autocorrelation function based on a finite record length]

It should be mentioned that in determining the time average of $X_T(t)X_T(t+\tau)$, we cannot use the whole span of time T, because $X_T(t)$ must be shifted an amount of τ with respect to itself before multiplication. The true extension of $X_T(t)$ beyond the experimental data span is unknown; therefore, we simply omit the nonoverlapped portion in the integration:

$$V_X(\tau) = [\text{time avg. of } X_T(t)X_T(t+\tau)] = \frac{1}{T-\tau}\int_0^{T-\tau} X_T(t)X_T(t+\tau)\, dt \tag{2.15.2}$$

It will be assumed from this point on that the range of τ being considered is much less than the total data span T, that is, $\tau \ll T$.

We first note that $V_X(\tau)$ is the result of analyzing a single time signal; therefore, $V_X(\tau)$ is itself just a sample function from an ensemble of functions. It is hoped that $V_X(\tau)$ as determined by Eq. (2.15.2) will yield a good estimate of $R_X(\tau)$ and, in order to do so, it should be an unbiased estimator. This can be verified by computing its expectation:

$$\begin{aligned} E[V_X(\tau)] &= E\left[\frac{1}{T-\tau}\int_0^{T-\tau} X_T(t)X_T(t+\tau)\, dt\right] \\ &= \frac{1}{T-\tau}\int_0^{T-\tau} E[X_T(t)X_T(t+\tau)]\, dt \\ &= \frac{1}{T-\tau}\int_0^{T-\tau} R_X(\tau)\, dt = R_X(\tau) \end{aligned} \tag{2.15.3}$$

Thus, $V_X(\tau)$ is an unbiased estimator of $R_X(\tau)$. Also, it can be seen from the equation for Var $V_X(\tau)$, Eq. (2.15.1), that if the integral of R_X^2 converges (e.g., R_X decreases exponentially with τ), then the variance of $V_X(\tau)$ approaches zero as T becomes large. Thus, $V_X(\tau)$ would appear to be a well-behaved estimator of $R_X(\tau)$, that is, $V_X(\tau)$ converges in the mean to $R_X(\tau)$. We will now pursue the estimation accuracy problem further.

Equation (2.15.1) is of little value if the process autocorrelation function is not known. So, at this point, we assume that $X(t)$ is a Gauss–Markov process with an autocorrelation function

$$R_X(\tau) = \sigma^2 e^{-\beta|\tau|} \tag{2.15.4}$$

The σ^2 and β parameters may be difficult to determine in a real-life problem, but we can get at least a rough estimate of the amount of experimental data needed for a given required accuracy. Substituting the assumed Markov autocorrelation function into Eq. (2.15.1) then yields

$$\text{Var}[V_X(\tau)] \leq \frac{2\sigma^4}{\beta T} \tag{2.15.5}$$

We now look at an example illustrating the use of Eq. (2.15.5).

EXAMPLE 2.14

Let us say that the process being investigated is thought to be a Gauss–Markov process with an estimated time constant $(1/\beta)$ of 1 sec. Let us also say that we

wish to determine its autocorrelation function within an accuracy of 10 percent, and we want to know the length of experimental data needed. By "accuracy" we mean that the experimentally determined $V(\tau)$ should have a standard deviation less than .1 of the σ^2 of the process, at least for a reasonably small range of τ near zero. Therefore, the ratio of $\text{Var}[V(\tau)]$ to $(\sigma^2)^2$ must be less than .01. Using Eq. (2.15.5), we can write

$$\frac{\text{Var}[V(\tau)]}{\sigma^4} \leq \frac{2}{\beta T}$$

Setting the quantity on the left side equal to .01 and using the equality condition yield

$$T = \frac{1}{(.1)^2} \cdot \frac{2}{\beta} \doteq 200 \text{ sec} \tag{2.15.6}$$

A sketch indicating a typical sample experimental autocorrelation function is shown in Fig. 2.24. Note that 10 percent accuracy is really not an especially demanding requirement, but yet the data required is 200 times the time constant of the process. To put this in more graphic terms, if the process under investigation were random gyro drift with an estimated time constant of 10 hours, 2000 hours of continuous data would be needed to achieve 10 percent accuracy. This could very well be in the same range as the mean time to failure for the gyro. Were we to be more demanding and ask for 1 percent accuracy, about 23 years of data would be required! It can be seen that accurate determination of the autocorrelation function is not a trivial problem in some applications. (This example is pursued further in Problem 2.33.) ■

The main point to be learned from this example is that reliable determination of the autocorrelation function takes considerably more experimental data than one might expect intuitively. The spectral density function is just the Fourier transform of the autocorrelation function, so we might expect a similar accuracy problem in its experimental determination.

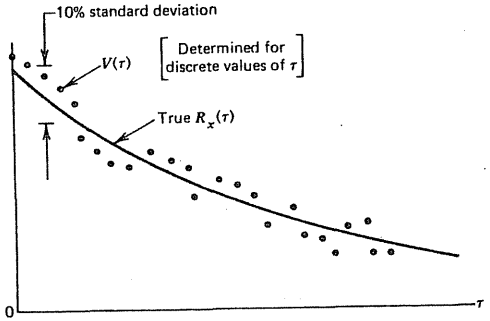


Figure 2.24 Experimental and true autocorrelation functions for Example 2.14.

As just mentioned, the spectral density function for a given sample signal may be estimated by taking the Fourier transform of the experimentally determined autocorrelation function. This, of course, involves a numerical procedure of some sort because the data describing $V_X(\tau)$ will be in numerical form. The spectral function may also be estimated directly from the periodogram of the sample signal. Recall from Section 2.7 that the average periodogram (the square of the magnitude of the Fourier transform of X_T) is proportional to the spectral density function (for large T). Unfortunately, since we do not usually have the luxury of having a large ensemble of periodograms to average, there are pitfalls in this approach, just as there are in going the autocorrelation route. Nevertheless, modern digital processing methods using fast Fourier transform (FFT) techniques have popularized the periodogram approach. Thus, it is important to understand its limitations (5, 6).

First, there is the truncation problem. When the time record being analyzed is finite in length, we usually assume that the signal will "jump" abruptly to zero outside the valid data interval. This causes frequency spreading and gives rise to high-frequency components that are not truly representative of the process under consideration, which is assumed to ramble on indefinitely in a continuous manner. An extreme case of this would occur if we were to chop up one long record into many very short records and then average the periodograms of the short records. The individual periodograms, with their predominance of high-frequency components due to the truncation, would not be at all representative of the spectral content of the original signal; nor would their average! Thus, the first rule is that we must have a long time record relative to the typical time variations in the signal. This is true regardless of the method used in analyzing the data. There is, however, a statistical convergence problem that arises as the record length becomes large, and this will now be examined.

In Section 2.7 it was shown that the expectation of the periodogram approaches the spectral density of the process for large T. This is certainly desirable, because we want the periodogram to be an unbiased estimate of the spectral density. It is also of interest to look at the behavior of the variance of the periodogram as T becomes large. Let us denote the periodogram of $X_T(\tau)$ as $M(\omega, T)$, that is,

$$M(\omega, T) = \frac{1}{T} |\mathfrak{F}\{X_T(t)\}|^2 \tag{2.15.7}$$

Note that the periodogram is a function of the record length T as well as ω. The variance of $M(\omega, T)$ is

$$\text{Var } M = E(M^2) - [E(M)]^2 \tag{2.15.8}$$

Since we have already found $E(M)$ as given by Eqs. (2.7.8) and (2.7.9), we now need to find $E(M)^2$. Squaring Eq. (2.15.7) leads to

$$E(M^2) = \frac{1}{T^2} E\left[\int_0^T \int_0^T \int_0^T \int_0^T X(t)X(s)X(u)X(v)e^{-j\omega(t-s+u-v)}\, dt\, ds\, du\, dv\right] \tag{2.15.9}$$

It can be shown that if $X(t)$ is a Gaussian process,*

$$E[X(t)X(s)X(u)X(v)] = R_X(t-s)R_X(u-v) + R_X(t-u)R_X(s-v) + R_X(t-v)R_X(s-u) \tag{2.15.10}$$

Thus, moving the expectation operator inside the integration in Eq. (2.15.9) and using Eq. (2.15.10) lead to

$$\begin{aligned} E(M^2) &= \frac{1}{T^2}\int_0^T\int_0^T\int_0^T\int_0^T [R_X(t-s)R_X(u-v) + R_X(t-u)R_X(s-v) \\ &\quad + R_X(t-v)R_X(s-u)]e^{-j\omega(t-s+u-v)}\,dt\,ds\,du\,dv \\ &= \frac{1}{T^2}\int_0^T\int_0^T R_X(t-s)e^{-j\omega(t-s)}\,dt\,ds \int_0^T\int_0^T R_X(u-v)e^{-j\omega(u-v)}\,du\,dv \\ &\quad + \frac{1}{T^2}\int_0^T\int_0^T R_X(t-v)e^{-j\omega(t-v)}\,dt\,dv \int_0^T\int_0^T R_X(s-u)e^{-j\omega(s-u)}\,ds\,du \\ &\quad + \frac{1}{T^2}\left|\int_0^T\int_0^T R_X(t-u)e^{-j\omega(t+u)}\,dt\,du\right|^2 \end{aligned} \tag{2.15.11}$$

Next, substituting Eq. (2.7.3) into (2.15.11) leads to

$$E(M^2) = 2[E(M)]^2 + \frac{1}{T^2}\left|\int_0^T\int_0^T R_X(t-u)e^{-j\omega(t+u)}\,dt\,du\right|^2 \tag{2.15.12}$$

Therefore,

$$\begin{aligned} \text{Var } M &= E(M^2) - [E(M)]^2 \\ &= [E(M)]^2 + \frac{1}{T^2}\left|\int_0^T\int_0^T R_X(t-u)e^{-j\omega(t+u)}\,dt\,du\right|^2 \end{aligned} \tag{2.15.13}$$

The second term of Eq. (2.15.13) is nonnegative, so it should be clear that

$$\text{Var } M \geq [E(M)]^2 \tag{2.15.14}$$

But $E(M)$ approaches the spectral function as $T \to \infty$. Thus, the variance of the periodogram does not go to zero as $T \to \infty$ (except possibly at those exceptional points where the spectral function is zero). In other words, the periodogram does not converge in the mean as $T \to \infty$! This is most disturbing, especially in view of the popularity of the periodogram method of spectral determination. The dilemma is summarized in Fig. 2.25. Increasing T will not help reduce the ripples in the individual periodogram. It simply makes M "jump around" faster with ω. This does help, though, with the subsequent averaging that must accompany the spectral analysis. Recall that it is the *average* periodogram that is the measure of the spectral density function. Averaging may not be essential in the analysis of deterministic signals, but it is for random signals. Averaging in both frequency and time is easily accomplished in analog spectrum analyzers by appropriate adjustment of the width of the scanning window and the sweep speed. In digital analyzers, similar averaging over a band of discrete frequencies can be implemented in software. Also, further averaging in time may be accomplished by averaging successive periodograms before displaying the spectrum graphically. In either event, analog or digital, some form of averaging is essential when analyzing noise. (Averaging over a window of frequencies is illustrated in Problem 2.34.)

Our treatment of the general problem of autocorrelation and spectral determination from experimental data must be brief. However, the message here should be clear. Treat this problem with respect. It is fraught with subtleties and pitfalls. Engineering literature abounds with reports of shoddy spectral analysis methods and the attendant questionable results. Know your digital signal processing methods and recognize the limitations of the results.

We will pursue the subject of digital spectral analysis further in Section 2.17. But first we digress to present Shannon's sampling theorems, which play an important role in digital signal processing.

* See Problem 2.30.

2.16 SAMPLING THEOREM

Consider a time function $g(t)$ that is bandlimited, that is,

$$\mathcal{F}[g(t)] = G(\omega) = \begin{cases} \text{Nontrivial}, & |\omega| \leq 2\pi W \\ 0, & |\omega| > 2\pi W \end{cases} \tag{2.16.1}$$

Under the conditions of Eq. (2.16.1), the time function can be written in the form

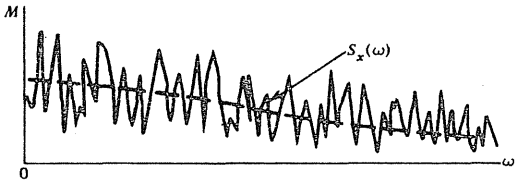


Figure 2.25 Typical periodogram for long record length.

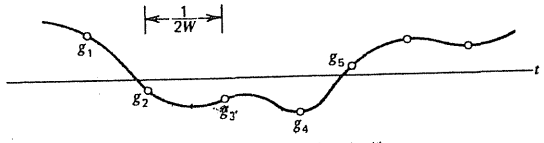


Figure 2.26 Samples of bandlimited signal $g(t)$.

$$g(t) = \sum_{n=-\infty}^{\infty} g\left(\frac{n}{2W}\right) \frac{\sin(2\pi Wt - n\pi)}{2\pi Wt - n\pi} \tag{2.16.2}$$

This remarkable theorem is due to C. E. Shannon (14, 15), and it has special significance when dealing with bandlimited noise.* The theorem says that if one were to specify an infinite sequence of sample values . . . , g_1, g_2, g_3, . . . , uniformly spaced $1/2W$ sec apart as shown in Fig. 2.26, then there would be one and only one bandlimited function that would go through all the sample values. In other words, specifying the signal sample values and requiring $g(t)$ to be bandlimited indirectly specify the signal in between the sample points as well. The sampling rate of $2W$ Hz is known as the *Nyquist rate*. This represents the minimum sampling rate needed to preserve all the information content in the continuous signal. If we sample $g(t)$ at less than the Nyquist rate, some information will be lost, and the original signal cannot be exactly reconstructed on the basis of the sequence of samples. Sampling at a rate higher than the Nyquist rate is not necessary, but it does no harm because this simply extends the allowable range of signal frequencies beyond W Hz. Certainly, a signal lying within the bandwidth W also lies within a bandwidth greater than W.

In describing a stationary random process that is bandlimited, it can be seen that we need to consider only the statistical properties of samples taken at the Nyquist rate of $2W$ Hz. This simplifies the process description considerably. If we add the further requirement that the process is Gaussian and white within the bandwidth W, then the joint probability density for the samples may be written as a simple product of single-variate normal density functions. This simplification is frequently used in noise analysis in order to make the problem mathematically tractable.

Since there is symmetry in the direct and inverse Fourier transforms, we would expect there to be a corresponding sampling theorem in the frequency domain. It may be stated as follows. Consider the time function $g(t)$ to be time limited, that is, nontrivial over a span of time T and zero outside this interval; then its Fourier transform $G(\omega)$ may be written as

* The basic concept of sampling at twice the highest signal frequency is usually attributed to H. Nyquist (16). However, the explicit form of the sampling theorem given by Eq. (2.16.2) and its associated signal bandwidth restriction was first introduced into communication theory by C. E. Shannon. You may wish to refer to Shannon (15) or Black (17) for further comments on the history of sampling theory.

$$G(\omega) = \sum_{n=-\infty}^{\infty} G\left(\frac{2\pi n}{T}\right) \frac{\sin\left(\frac{\omega T}{2} - n\pi\right)}{\frac{\omega T}{2} - n\pi} \tag{2.16.3}$$

All of the previous comments relative to time domain sampling have their corresponding frequency-domain counterparts.

Frequently, it is useful to consider time functions that are limited in both time and frequency. Strictly speaking, this is not possible, but it is a useful approximation. This being the case, the time function can be uniquely represented by $2WT$ samples. These may be specified either in the time or frequency domain.

Sampling theorems have also been worked out for the nonbaseband case (18, 19). These are somewhat more involved than the baseband theorems and will not be given here.

2.17 DISCRETE FOURIER TRANSFORM AND FAST FOURIER TRANSFORM

The subject of digital signal processing has received considerable attention in the past few decades, and it is only natural that this would occur concurrently with the advancement of computer technology. Whole books have been devoted to the subject (20, 21), so that we cannot expect to do the matter justice in one short section. We can, however, present a brief overview in order to place digital spectral analysis in proper perspective. We will then proceed on to the main subject of this book, namely, filter analysis.

Modern computer technology has made it possible to perform an efficient discrete version of the Fourier transform. Thus, nearly all spectral analysis is now done using the direct periodogram approach rather than the more roundabout approach via the autocorrelation function. The sampling theorems presented in Section 2.16 dictate some constraints in the choice of sampling rates and the total amount of data analyzed in any one batch. Since these constraints play an important role in digital signal processing, we will examine their consequences in some detail.

In spectral analysis, we usually have at least a rough idea as to the bandwidth of the signal to be analyzed; therefore, let us say this is approximately 0 to W Hz. The sampling theorem, Eq. (2.16.2), says that the sampling rate should be $2W$ samples/sec or, equivalently, the sample spacing should be $1/2W$ sec. Let us further say that we wish to analyze N samples in one batch where N is yet to be determined. The total time span of the samples would then be

$$\text{Total time span of data} = T = \frac{N}{2W} \tag{2.17.1}$$

The frequency-domain sampling theorem, Eq. (2.16.3), states that our truncated time signal could be represented in the spectral domain with discrete samples spaced $1/T$ or $2W/N$ apart. That is, we have N samples uniformly spaced in the time domain and $N/2$ corresponding samples spaced uniformly from 0 to W Hz in the frequency domain. Since each spectral sample is a complex number, the number of degrees of freedom is N in either the time or frequency domain. This one-to-one correspondence of scalar elements in the two domains suggests a transform-pair relationship, and this will now be formalized.

As a matter of notation, let the truncated time signal be $g(t)$ and its Fourier transform be

$$G(j\omega) = \int_0^T g(t)e^{-j\omega t}\,dt \tag{2.17.2}$$

It is tacitly assumed here that $g(t)$ is real. Consider next a discrete approximation for $G(j\omega)$ as follows:

$$G(jn2\pi\,\Delta f) \approx \sum_{k=0}^{N-1} g_k e^{-jn2\pi\,\Delta f k\,\Delta T}\,\Delta T \tag{2.17.3}$$

where

g_k = sequence of N time samples of $g(t)$, $\quad k = 0, 1, \ldots, N-1$

$\Delta T = \dfrac{1}{2W}$ = sample spacing in time domain

$\Delta f = \dfrac{2W}{N}$ = sample spacing in frequency domain (in hertz)

We note now that

$$\Delta f\,\Delta T = \frac{1}{N} \tag{2.17.4}$$

Thus, Eq. (2.17.3) can be rewritten in the form

$$G(jn2\pi\,\Delta f)\,\Delta f \approx \frac{1}{N}\sum_{k=0}^{N-1} g_k \exp\left(-j\frac{2\pi nk}{N}\right) \tag{2.17.5}$$

Now, given the time sequence $g_0, g_1, \ldots, g_{N-1}$, think of the right side of Eq. (2.17.5) as defining another sequence $\mathcal{G}_0, \mathcal{G}_1, \ldots, \mathcal{G}_{N-1}$ as n is indexed from 0 to $N-1$. That is, the $\mathcal{G}_n$ sequence is defined as

$$\mathcal{G}_n = \frac{1}{N}\sum_{k=0}^{N-1} g_k \exp\left(-j\frac{2\pi nk}{N}\right), \qquad n = 0, 1, \ldots, N-1 \tag{2.17.6}$$

Note that we do not claim the $\mathcal{G}_n$'s to be exact samples of $G(j\omega)$, but, it is hoped, they will be reasonable approximations thereof.

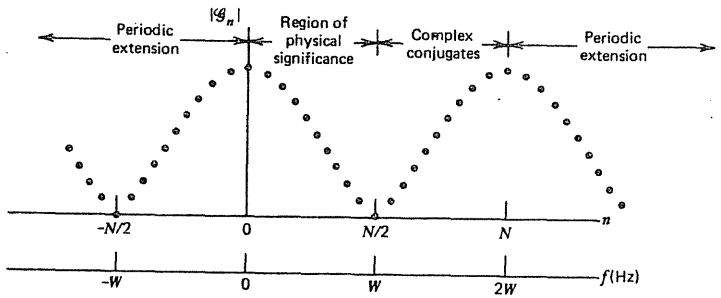


Figure 2.27 Discrete Fourier transform magnitudes and their significance.

The $\mathcal{G}_n$ sequence as defined by Eq. (2.17.6) exhibits certain symmetry that is worth noting. First, if we extend the index n beyond $N-1$, we simply get a periodic extension of the sequence:

$$\begin{aligned} \mathcal{G}_N &= \mathcal{G}_0 \\ \mathcal{G}_{N+1} &= \mathcal{G}_1 \\ &\vdots \\ &\text{etc.} \end{aligned} \tag{2.17.7}$$

Also, there is symmetry about the midpoint of the sequence in that

$$\begin{aligned} \mathcal{G}_{N-1} &= \mathcal{G}_1^* \\ \mathcal{G}_{N-2} &= \mathcal{G}_2^* \\ &\vdots \\ &\text{etc.} \end{aligned} \tag{2.17.8}$$

In other words, half of the defined $\mathcal{G}_n$'s are complex conjugates of the other half and are thus redundant (see Problem 2.31). This is as expected; there can only be N degrees of freedom in the frequency domain, just as in the time domain. Figure 2.27 summarizes the symmetry properties of the $\mathcal{G}$ sequence.

Once the $\mathcal{G}_n$ sequence is defined as per Eq. (2.17.6), it can be shown that an exact inverse relationship exists as follows (20, 21):

$$g_k = \sum_{n=0}^{N-1} \mathcal{G}_n \exp\left(j\frac{2\pi nk}{N}\right), \qquad k = 0, 1, \ldots, N-1 \tag{2.17.9}$$

The $\mathcal{G}_n$ and g_k sequences then form a transform pair in the usual sense; that is, given one, we can find the other, and vice versa.* As might be expected, the $\mathcal{G}_n$ sequence is called the *discrete Fourier transform* (DFT) of g_k and, of course, g_k is the discrete inverse transform of $\mathcal{G}_n$. It should be emphasized that there is no approximation involved in the discrete transform pair relationship. This relationship is exact, irrespective of the sampling rate or frequency content of $g(t)$. The approximation comes in the *interpretation* of $\mathcal{G}_n$ as samples of the continuous signal spectrum. This is a relatively complicated matter and the references cited (5, 20, 21) may be consulted for further details. It suffices here to say that this has been studied extensively and, with appropriate cautions and proper weighting of the time data, the discrete Fourier transform can provide meaningful results.

We might also note that all the preceding remarks about digital signal processing apply to deterministic as well as random signals. In analyzing deterministic signals, we usually compute the magnitude of the discrete Fourier transform and call this the signal spectrum. In the random signal case, usually the square of the magnitudes of the $\mathcal{G}_n$ terms are formed, and this sequence (i.e., $|\mathcal{G}_0|^2$, $|\mathcal{G}_1|^2$, . . .) becomes an approximation of the periodogram of the signal being analyzed. The periodogram is, in turn, statistically related to the power spectral density function, which is usually the desired end result of the analysis. (See Problem 2.29 for comments about the crossperiodogram.)

Digital implementation of the discrete Fourier transform is not a trivial matter. High resolution and reliable results in the frequency domain are obtained by making N large. However, if one programs the transform literally as given by Eq. (2.17.6), the number of multiplications required is of the order N^2. This can easily get out of hand, especially in "on-line" applications. Fortunately, fast, efficient algorithms have been developed for which the number of required multiplications is of the order $N \log_2 N$ rather than N^2 (20, 21). The computational saving is spectacular for large N. For example, let N be $2^{10} = 1024$, which is a modest number of time samples for many applications. Then N^2 would be about 10^6, whereas $N \log_2 N$ is only about 10^4. This represents a saving of about a factor of 100 and reflects directly into the time required for the transformation.

All of the fast discrete Fourier transform algorithms require that the number of samples be an integer power 2 (which usually presents no particular problem), and they all go under the generic name of *fast Fourier transform* (FFT). The FFT cannot perform any wondrous, magical tricks on the basic data (as some seem to believe); it is simply an efficient means of implementing the DFT. Thus, all of the cautions that apply to the DFT also apply to the FFT. Because of its efficiency, though, the FFT is used almost universally in on-line spectral analysis applications. Occasionally, though, in off-line applications where speed is of little concern, the straightforward programming of the DFT, as given by Eq. (2.17.6), is advantageous. For example, if only a limited amount of data is available and it is desirable to achieve as fine a resolution as possible in the frequency domain, the straightforward DFT might as well be preferred, because N is not restricted to integer powers of 2 as it is with the FFT.

* Some authors prefer to associate the $1/N$ factor of Eq. (2.17.6) with the inverse relationship rather than the direct transform. Since it is a simple proportionality factor, this is perfectly proper.

Before we leave the subject of the discrete Fourier transform, it is worth mentioning that the DFT is easily generalized to the case where the time-domain samples are complex, rather than real as assumed here. The direct and inverse transform Eqs. (2.17.6) and (2.17.9) still apply in the complex case. If one begins with N complex samples in the time domain, then there will be a corresponding set of N complex samples in the frequency domain. Also, if no special symmetry exists among the time-domain samples, then there will be no special symmetry or "folding over" in the frequency domain. The periodic extension stated in Eq. (2.17.7) still applies in the complex case, though.

PROBLEMS

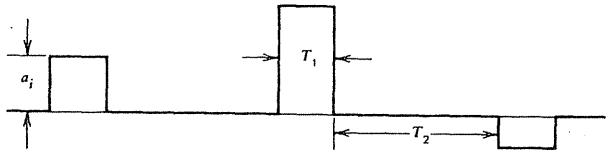
2.1 Noise measurements at the output of a certain amplifier (with its input shorted) indicate that the rms output voltage due to internal noise is 100 μV. If we assume that the frequency spectrum of the noise is flat from 0 to 10 MHz and zero above 10 MHz, find:

(a) The spectral density function for the noise.

(b) The autocorrelation function for the noise.

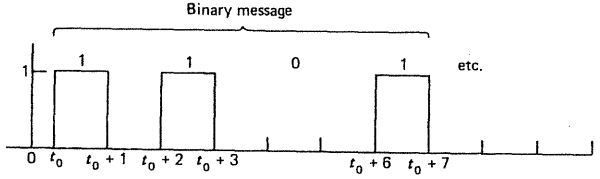
Give proper units for both the spectral density and autocorrelation functions.

2.2 A sketch of a sample realization of a random process $X(t)$ is shown in the figure. The pulse amplitudes a_i are independent samples of a normal random variable with zero mean and variance σ^2. The time origin is completely random. Find the autocorrelation function for the process.



Problem 2.2

2.3 The waveform shown is an example of a digital-coded waveform. The signal is equally likely to be zero or one in the intervals $(t_0, t_0 + 1)$, $(t_0 + 2, t_0 + 3)$, etc., and it is always zero in the "in between" intervals $(t_0 + 1, t_0 + 2)$, $(t_0 + 3, t_0 + 4)$, etc. The switching time t_0 is random and uniformly distributed between zero and two. The presence or absence of a pulse in the "pulse possible" intervals is the code for a binary digit. There is no statistical correlation among any of the bits of the message. Find the autocorrelation and spectral density functions for this process.



Problem 2.3

2.4 Find the autocorrelation function corresponding to the spectral density function

$$S(j\omega) = \delta(\omega) + \tfrac{1}{2}\delta(\omega - \omega_0) + \tfrac{1}{2}\delta(\omega + \omega_0) + e^{-2|\omega|}$$

2.5 A stationary Gaussian random process $X(t)$ has an autocorrelation function of the form

$$R_X(\tau) = 4e^{-|\tau|}$$

What fraction of the time will $|X(t)|$ exceed four units?

2.6 A random process $X(t)$ is generated according to the following rules:

(a) The waveform is generated with a sample-and-hold arrangement where the "hold" interval is 1 sec.

(b) The successive amplitudes for each 1-sec interval are independent samples taken from a zero-mean normal distribution with a variance of σ^2.

(c) The first switching time is a random variable with uniform distribution from 0 to 1 (i.e., the time origin is random).

Let X_1 denote $X(t_1)$ and X_2 denote $X(t_1 + \tau)$.

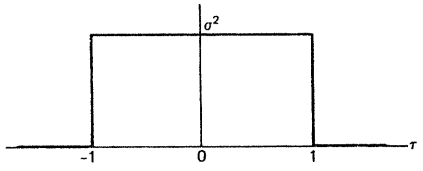
(a) Find the joint probability density function $f_{X_1X_2}(x_1, x_2)$.

(b) Is this process a Gaussian process?

2.7 Find the autocorrelation function for the process described in Problem 2.6 using the expectation formula

$$R_X(\tau) = E[X_1X_2] = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x_1x_2 f_{X_1X_2}(x_1, x_2)\, dx_1\, dx_2$$

2.8 It is suggested that a certain real process has an autocorrelation function as shown in the figure. Is this possible? Justify your answer.
(*Hint:* Calculate the spectral density function and see if it is plausible.)



Problem 2.8

2.9 Consider the random process $X(t) = 2 \sin \omega t$ where ω is a random variable with uniform distribution between $\omega = 2$ and $\omega = 6$. Is the process (a) stationary, (b) ergodic, and (c) deterministic or nondeterministic?

2.10 The input to an ideal rectifier (unity forward gain, zero reverse gain) is a stationary Gaussian process.

(a) Is the output stationary?

(b) Is the output a Gaussian process?

Give a brief justification for both answers.

2.11 A random process $X(t)$ has sample realizations of the form

$$X(t) = at + Y$$

where a is a known constant and Y is a random variable whose distribution is $N(0, \sigma^2)$. Is the process (a) stationary and (b) ergodic? Justify your answers.

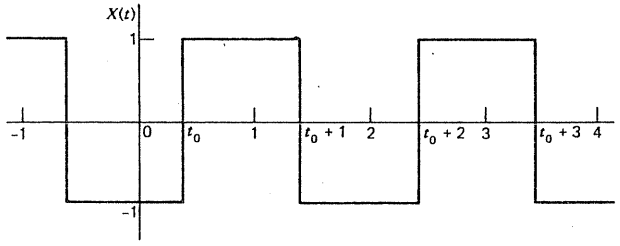
2.12 What is the autocorrelation function for $X(t)$ of Problem 2.11?

2.13 A sample realization of a random process $X(t)$ is shown in the figure. The time t_0 when the transition from the -1 state to the $+1$ state takes place is a random variable that is uniformly distributed between 0 and 2 units.

(a) Is the process stationary?

(b) Is the process deterministic or nondeterministic?

(c) Find the autocorrelation function and spectral density function for the process.



Problem 2.13

2.14 A common autocorrelation function encountered in physical problems is

$$R(\tau) = \sigma^2 e^{-\beta|\tau|} \cos \omega_0 \tau$$

(a) Find the corresponding spectral density function.

(b) $R(\tau)$ will be recognized as a damped cosine function. Sketch both the autocorrelation and spectral density functions for the lightly damped case.

2.15 Show that a Gauss–Markov process described by the autocorrelation function

$$R(\tau) = \sigma^2 e^{-\beta|\tau|}$$

becomes Gaussian white noise if we let $\beta \to \infty$ and $\sigma^2 \to \infty$ in such a way that the area under the autocorrelation-function curve remains constant in the limiting process.

2.16 A stationary random process $X(t)$ has a spectral density function of the form

$$S_X(\omega) = \frac{6\omega^2 + 12}{(\omega^2 + 4)(\omega^2 + 1)}$$

What is the mean-square value of $X(t)$?
(*Hint:* $S_X(\omega)$ may be resolved into a sum of two terms of the form: $[A/(\omega^2 + 4)] + [B/(\omega^2 + 1)]$. Each term may then be integrated using standard integral tables.)

2.17 The stationary process $X(t)$ has an autocorrelation function of the form

$$R_X(\tau) = \sigma^2 e^{-\beta|\tau|}$$

Another process $Y(t)$ is related to $X(t)$ by the deterministic equation

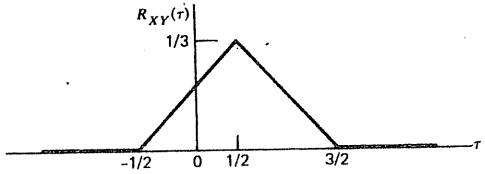
$$Y(t) = aX(t) + b$$

where a and b are known constants.

(a) What is the autocorrelation function for $Y(t)$?

(b) What is the crosscorrelation function $R_{XY}(\tau)$?

2.18 The crosscorrelation function $R_{XY}(\tau)$ for Example 2.8 is sketched below. What is the corresponding cross spectral density function?



Problem 2.18

2.19 The random telegraph wave is described in Section 2.11. Let this process be the input to an ideal rectifier (unity forward gain, zero reverse gain).

(a) What is the autocorrelation function of the output?

(b) What is the crosscorrelation function $R_{XY}(\tau)$ (X is input, Y is output)?

2.20 Two deterministic random processes are defined by

$$X(t) = A \sin(\omega t + \theta)$$

$$Y(t) = B \sin(\omega t + \theta)$$

where θ is a random variable with uniform distribution between 0 and 2π, and ω is a known constant. The A and B coefficients are both normal random variables $N(0, \sigma^2)$ and are correlated with a correlation coefficient ρ. What is the crosscorrelation function $R_{XY}(\tau)$? (Assume A and B are independent of θ.)

2.21 The discrete random walk process is discussed in Section 2.13. Assume each step is of length l and that the steps are independent and equally likely to be positive or negative. Show that the variance of the total distance D traveled in N steps is given by

$$\text{Var } D = l^2 N$$

(*Hint:* First write D as the sum $l_1 + l_2 + \cdots l_N$ and note that $l_1, l_2, \ldots, l_N$ are independent random variables. Then form $E(D)$ and $E(D^2)$ and compute Var D as $E(D^2) - [E(D)]^2$.)

2.22 Let the process $Z(t)$ be the product of two independent stationary processes $X(t)$ and $Y(t)$. Show that the spectral density function for $Z(t)$ is given by (in the s domain)

$$S_Z(s) = \frac{1}{2\pi j} \int_{-j\infty}^{j\infty} S_X(w) S_Y(s - w)\, dw$$

[*Hint:* First show that $R_Z(\tau) = R_X(\tau) R_Y(\tau)$.]

2.23 The spectral density function for the stationary process $X(t)$ is

$$S_X(j\omega) = \frac{1}{(1 + \omega^2)^2}$$

Find the autocorrelation function for $X(t)$.

2.24 A stationary process $X(t)$ is Gaussian and has an autocorrelation function of the form

$$R_X(\tau) = 4e^{-|\tau|}$$

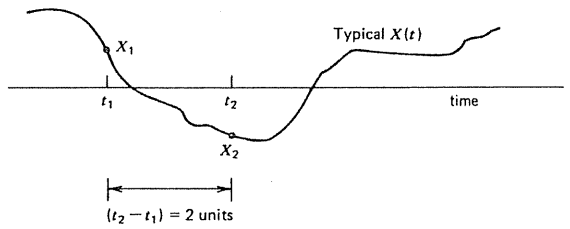
Let the random variable X_1 denote $X(t_1)$ and X_2 denote $X(t_1 + 1)$. Write the expression for the joint probability density function $f_{X_1 X_2}(x_1, x_2)$.

2.25 A stationary Gaussian process $X(t)$ has a power spectral density function

$$S_X(j\omega) = \frac{2}{\omega^4 + 1}$$

Find $E(X)$ and $E(X^2)$.

2.26 A typical sample function of a stationary Gauss–Markov process is shown in the sketch. The process has a mean-square value of 9 units, and the random variables X_1 and X_2 indicated on the waveform have a correlation coefficient of 0.5. Write the expression for the autocorrelation function of $X(t)$.



Problem 2.26

2.27 We wish to determine the autocorrelation function a random signal empirically from a single time record. Let us say we have good reason to believe the process is ergodic and at least approximately Gaussian and, furthermore, that the autocorrelation function of the process decays exponentially with a time constant no greater than 10 sec. Estimate the record length needed to achieve 5 percent accuracy in the determination of the autocorrelation function. (By 5 percent accuracy, assume we mean that for any τ, the standard deviation of the experimentally determined autocorrelation function will not be more than 5 percent of the maximum value of the true autocorrelation function.)

2.28 In Problem 2.27 the signal is not known to be truly bandlimited, but it is reasonable to assume that essentially all the signal energy will lie between 0

and .1 Hz. Let us assume .1 Hz to be the signal bandwidth, and let us say we wish to sample the signal at the Nyquist rate.

(a) How many discrete samples would be required to describe the record length of Problem 2.27?

(b) Suppose that we wish to compute the discrete Fourier transform of the finite-time signal using the fast Fourier transform. This requires that the number of samples N be an integer power of 2. What should N be in this case?

(c) The value of N in part (b) should work out to be greater than that found in part (a). In order to achieve the appropriate N for the FFT algorithm, would we be better off (in terms of accuracy) to increase the sampling rate for the length of time computed in Problem 2.27 or should we keep the sampling rate at .2 Hz and increase the time length of the record accordingly? Presumably, the computational effort would be the same either way.

2.29 Calculating either the autocorrelation or crosscorrelation function can be an onerous task, especially if there is a large amount of data to be processed. It is sometimes easier to compute the appropriate spectral function first and then inverse transform it, rather than do the computation as an averaging process directly in the time domain. This may seem to be a "roundabout" approach at first glance, but the ease of using the FFT often makes this approach attractive.

Suppose we have time records of length T of two stationary random processes $x(t)$ and $y(t)$, and suppose we are interested in obtaining the crosscorrelation $R_{xy}(\tau)$ of these two processes. Show that a crossperiodogram-type function can be formed just as was done for the usual power spectral density, and that it is given by

$$\text{Crossperiodogram} = M_{xy}(\omega, T) \triangleq \frac{1}{T} X_T(-j\omega) Y_T(j\omega)$$

where X_T and Y_T are the Fourier transforms of the truncated $x_T(t)$ and $y_T(t)$ functions. Note that it is relatively easy to form the indicated product function in the complex domain and then do another FFT to get the crosscorrelation function in the time domain. (It should be mentioned that all of the same statistical convergence problems mentioned in Section 2.15 relative to the periodogram also apply to the crossperiodogram.)

2.30 Let X_1, X_2, X_3, X_4 be zero-mean Gaussian random variables. Show that

$$E(X_1X_2X_3X_4) = E(X_1X_2)E(X_3X_4) + E(X_1X_3)E(X_2X_4) + E(X_1X_4)E(X_2X_3) \tag{P2.30.1}$$

(*Hint:* The characteristic function was discussed briefly in Section 1.8.) The multivariate version of the characteristic function is useful here. Let $\psi(\omega_1, \omega_2, \ldots, \omega_n)$ be the multidimensional Fourier transform of $f_{X_1X_2\cdots X_n}(x_1, x_2, \ldots, x_n)$ (but with the signs reversed on $\omega_1, \omega_2, \ldots, \omega_n$). Then it can be readily verified that

$$(-j)^n \left.\frac{\partial^n \psi(\omega_1, \omega_2, \ldots, \omega_n)}{\partial\omega_1\, \partial\omega_2, \ldots, \partial\omega_n}\right|_{\substack{\omega_1=0 \\ \omega_2=0 \\ \text{etc.}}}$$

$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty} x_1, x_2, \ldots,$$

$$x_n f_{X_1X_2,\ldots,X_n}(x_1, x_2, \ldots, x_n)\, dx_1, dx_2, \ldots, dx_n$$

$$= E(X_1, X_2, \ldots, X_n) \tag{P2.30.2}$$

The characteristic function for a zero-mean, vector Gaussian random variable $\mathbf{X}$ is

$$\psi(\boldsymbol{\omega}) = e^{-\frac{1}{2}\boldsymbol{\omega}^T \mathbf{C}_x \boldsymbol{\omega}} \tag{P2.30.3}$$

where $\mathbf{C}_X$ is the covariance matrix for $\mathbf{X}$. This, along with Eq. (P2.30.2), may now be used to justify the original statement given by Eq. (P2.30.1).

2.31 It was mentioned in Section 2.17 that half of the discrete Fourier transform elements are complex conjugates of the other half. This statement deserves closer scrutiny because N may be either odd or even insofar as the basic discrete transform is concerned (and not its efficient FFT implementation).

(a) For the case where N is even, show that $\mathcal{G}_0$ and $\mathcal{G}_{N/2}$ are both real, and thus the total number of nonredundant scalar elements in the frequency domain is N, just as in the time domain.

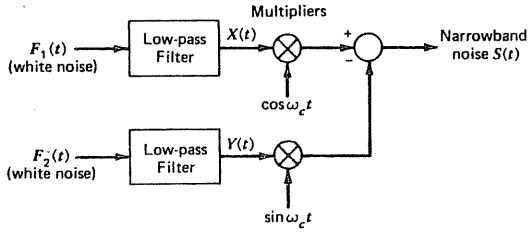
(b) For the case where N is odd, show that only $\mathcal{G}_0$ is constrained to be real, and thus the count of nonredundant scalar elements is N, just as in the previous case.

2.32 The accompanying figure shows a means of generating narrowband noise from two independent baseband noise sources. (See Section 2.12.) The bandwidth of the resulting narrowband noise is controlled by the cutoff frequency of the low-pass filters, which are assumed to have identical characteristics. Assume that $F_1(t)$ and $F_2(t)$ are independent white Gaussian noise processes with similar spectral amplitudes. The resulting noise processes after low-pass filtering will then have identical autocorrelation functions that will be denoted $R_X(\tau)$.

(a) Show that the narrowband noise $S(t)$ is a stationary Gaussian random process whose autocorrelation function is

$$R_S(\tau) = R_X(\tau) \cos \omega_c \tau$$

(b) Also show that both the in-phase and quadrature channels are needed to produce stationary narrowband noise. (That is, if either of the $\sin \omega_c t$ or $\cos \omega_c t$ multiplying operations is omitted, the resultant output will not be strictly stationary.)



Problem 2.32

2.33 A sequence of discrete samples of a Gauss–Markov process can be generated using the following difference equation (see Section 5.4):

$$X_{k+1} = e^{-\beta\Delta t}X_k + W_k, \qquad k = 0, 1, 2, \ldots$$

W_k = white sequence, $N[0, \sigma^2(1 - e^{-2\beta\Delta t})]$

σ^2 = variance of the Markov process

β = reciprocal time constant of the process

Δt = time interval between samples

If the initial value of the process X_0 is chosen from a population that is $N(0, \sigma^2)$, then the sequence so generated will be a sample realization of a stationary Gauss–Markov process. Such a sample realization is easily generated with MATLAB's normal random number generator with appropriate scaling of the initial X_0 and the W_k sequence.

(a) Generate 1024 samples of a Gauss–Markov process with $\sigma^2 = 1$, $\beta = 1$, and $\Delta t = .05$ sec. As a matter of convenience, let the samples be a 1024-element row vector with a suitable variable name.

(b) Calculate the experimental autocorrelation function for the X_k sequence of part (a). That is, find $V_X(\tau)$ for $\tau = 0, .05, .10, \ldots, 3.0$ (i.e., 60 "lags"). You will find it convenient here to write a general MATLAB program for computing the autocorrelation function for a sequence of length s and for m lags. (This program can then be used in subsequent problems.) Compare your experimentally determined $V_X(\tau)$ with the true autocorrelation function $R_X(\tau)$ by plotting both $V_X(\tau)$ and $R_X(\tau)$ on the same graph. Note that for the relatively short 1024-point time sequence being used here, you should not expect to see a close match between $V_X(\tau)$ and $R_X(\tau)$ (see Example 2.14).

(c) The theory given in Section 2.15 states that the expectation of $V_X(\tau)$ is equal to $R_X(\tau)$ regardless of the length of the sequence. It is also shown that $V_X(\tau)$ converges in the mean for a Gauss–Markov process as T becomes large. One would also expect to see the same sort of convergence when we look at the average of an ensemble of $V_X(\tau)$'s that are generated from "statistically identical," but different, 1024-point sequences. This can be demonstrated (not proved) using a different seed in developing each $V_X(\tau)$. Say we use seeds 1, 2, . . . , 8. First plot the $V_X(\tau)$ obtained using seed 1. Next, average the two $V_X(\tau)$'s obtained from seeds 1 and 2, and plot the result. Then average the four $V_X(\tau)$'s for seeds 1, 2, 3, and 4, and plot the result. Finally, average all eight $V_X(\tau)$'s, and plot the result. You should see a general trend toward the true $R_X(\tau)$ as the number of $V_X(\tau)$'s used in the average increases.

2.34 A recursion equation for generating a Gauss–Markov process was given in Problem 2.33. We wish to use the same process here, but the sampling rate will be changed to make it more suitable for a demonstration of experimental determination of the process power spectral density function. Therefore, the parameters will be the same as in Problem 2.33, except that we will let $\Delta\tau = 1.0$ sec in this problem.

(a) Generate a 256-point (i.e., samples) realization of the process just described and store the samples as a row vector.

(b) Using the first 64 samples of the process, calculate the discrete periodogram of the process (see Eq. 2.15.7). To do this, you will need to do a discrete Fourier transform (DFT) of the 64-point time series. This can be done either by writing your own *m*-file implementing the DFT according to Eq. (2.17.6) or by using the built-in MATLAB function *fft*(*x*). The results should be the same (within a scale factor). Plot the resulting periodogram.

(c) Repeat part (b) using the first 128 samples of the process, and then repeat it again using all 256 samples of the process. Note that the "noisiness" of the successive periodograms does not diminish as more time data are included. This is consistent with Eq. (2.15.14). As we include more time data, the frequency samples crowd closer together, but the noisiness of the samples remains the same. (Note that this is in contrast to the autocorrelation function determination, where increasing the time span smooths out the experimental estimate of the true autocorrelation function.)

(d) One way of smoothing the noisiness of the discrete periodogram is to average the data in the frequency domain. To demonstrate this, reconsider the 256-point periodogram of part (c) and form an "averaged periodogram" by averaging the data in successive 8-point blocks in the frequency domain. This yields a smoother plot, but with coarser resolution, of course. Now each point represents the power in a bandwidth of $8\Delta f$ rather than Δf as before. Note that the average periodogram is beginning to approximate the true spectral density that is $2\sigma^2\beta/(\omega^2 + \beta^2)$. This approximation can be improved further by holding the sampling interval constant and increasing the number of samples in the time domain.

2.35 Discrete samples of a Wiener process are easily generated using MATLAB's normal random number generator and implementing the recursion equation:

$$X_{k+1} = X_k + W_k, \qquad k = 0, 1, 2, \ldots \tag{P2.35}$$

where the subscript k is the time index and the initial condition is set to

$$X_0 = 0$$

Consider a Wiener process where the white noise being integrated has a power spectral density of unity (see Section 2.13), and the sampling interval is 1 sec. The increment to be added with each step (i.e., W_k) is a $N(0, 1)$ random variable, and all the W_k's are independent. [That this will generate samples of a process whose variance is t (in sec) is easily verified by working out the variance of X_k for a few steps beginning at $k = 0$.]

(a) Using Eq. (P2.35), generate an ensemble of 50 sample realizations of the Wiener process described above for $k = 0, 1, 2, \ldots, 10$. For convenience, arrange the resulting realizations into a 50×11 matrix, where each row represents a sample realization beginning at $k = 0$.

(b) Plot any 8 sample realizations (i.e., rows) from part (a), and note the obvious nonstationary character of the process.

(c) Form the average squares of the 50 process realizations from part (a), and plot the result vs. time (i.e., k). (The resulting plot should be approximately linear with a slope of unity.)

REFERENCES CITED IN CHAPTER 2

1. E. Wong, *Stochastic Processes in Information and Dynamical Systems,* New York: McGraw-Hill, 1971.
2. A. H. Jazwinski, *Stochastic Processes and Filtering Theory,* New York: Academic Press, 1970.
3. N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series,* Cambridge, MA: MIT Press and New York: Wiley, 1949.
4. S. O. Rice, "Mathematical Analysis of Noise," *Bell System Tech. J., 23,* 282–332 (1944); *24,* 46–256 (1945).
5. J. S. Bendat and A. G. Piersol, *Random Data: Analysis and Measurement Procedures,* New York: Wiley-Interscience, 1971.
6. K. S. Shanmugan and A. M. Breipohl, *Random Signals: Detection, Estimation, and Data Analysis,* New York: Wiley, 1988.
7. W. B. Davenport, Jr. and W. L. Root, *An Introduction to the Theory of Random Signals and Noise,* New York: McGraw-Hill, 1958.
8. A. Papoulis, *Probability, Random Variables, and Stochastic Processes,* 2nd ed., New York: McGraw-Hill, 1984.
9. R. C. Dixon, *Spread Spectrum Systems,* New York: Wiley, 1976.
10. R. P. Denaro, "Navstar: The All-Purpose Satellite," *IEEE Spectrum, 18:*5, 35–40 (May 1981).
11. B. W. Parkinson and S. W. Gilbert, "NAVSTAR: Global Positioning System—Ten Years Later," *Proc. IEEE, 71:*10, 1177–1186 (Oct. 1983).
12. R. E. Ziemer and R. L. Peterson, *Digital Communications and Spread Spectrum Systems,* New York: Macmillan, 1985.
13. J. H. Laning, Jr. and R. H. Battin, *Random Processes in Automatic Control,* New York: McGraw-Hill, 1956.
14. C. E. Shannon, "The Mathematical Theory of Communication," *Bell System Tech. J.* (July and Oct. 1948). (Later reprinted in book form by the University of Illinois Press, 1949.)
15. C. E. Shannon, "Communication in the Presence of Noise," *Proc. Inst. Radio Engr., 37:*1, 10–21 (Jan. 1949).
16. H. Nyquist, "Certain Topics in Telegraph Transmission Theory," *Trans. Am. Inst. Elect. Engr., 47,* 617–644 (April 1928).

17. H. S. Black, *Modulation Theory,* New York: Van Nostrand Co., 1950.
18. S. Goldman, *Information Theory,* Englewood Cliffs, NJ: Prentice-Hall, 1953.
19. K. S. Shanmugam, *Digital and Analog Communication Systems,* New York: Wiley, 1979.
20. A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing,* Englewood Cliffs, NJ: Prentice-Hall, 1989.
21. D. Childers and A. Durling, *Digital Filtering and Signal Processing,* St. Paul, MN: West Publishing Co., 1975.

Additional References on Probability and Random Signals

22. P. Z. Peebles, Jr., *Probability, Random Variables, and Random Signal Principles,* 3rd ed., New York: McGraw-Hill, 1993.
23. H. J. Larson and B. O. Shubert, *Probabilistic Models in Engineering Sciences* (Vols. 1 and 2), New York: Wiley, 1979.
24. G. R. Cooper and C. D. McGillem, *Probabilistic Methods of Signal and System Analysis,* 2nd ed., New York: Holt, Rinehart, and Winston, 1986.
25. J. L. Melsa and A. P. Sage, *An Introduction to Probability and Stochastic Processes,* Englewood Cliffs, NJ: Prentice-Hall, 1973.
26. A. M. Breipohl, *Probabilistic Systems Analysis,* New York: Wiley, 1970.
27. J. V. Candy, *Signal Processing: The Model-Based Approach,* New York: McGraw-Hill, 1986.

3

Response of Linear Systems to Random Inputs

The central problem of linear systems analysis is: Given the input, what is the output? In the deterministic case, we usually seek an explicit expression for the response or output. In the random-input problem no such explicit expression is possible, except for the special case where the input is a so-called deterministic random process (and not always in this case). Usually, in random-input problems, we must settle for a considerably less complete description of the output than we get for corresponding deterministic problems. In the case of random processes the most convenient descriptors to work with are autocorrelation function, spectral density function, and mean-square value. We now examine the input–output relationships of linear systems in these terms.

3.1 INTRODUCTION: THE ANALYSIS PROBLEM

In any system satisfying a set of linear differential equations, the solution may be written as a superposition of an initial-condition part and another part due to the driving or forcing functions. Both the initial conditions and forcing functions may be random; and, if so, the resultant response is a random process. We direct our attention here to such situations, and it will be assumed that the reader has at least an elementary acquaintance with deterministic methods of linear system analysis (1, 2, 3).

With reference to Fig. 3.1, the analysis problem may be simply stated: Given the initial conditions and the input and the system's dynamical characteristics [i.e., $G(s)$ in Fig. 3.1], what is the output? Of course, in the stochastic problem, the input and output will have to be described in probabilistic terms.

We need to digress here for a moment and discuss a notational matter. In Chapters 1 and 2 we were careful to use uppercase symbols to denote random

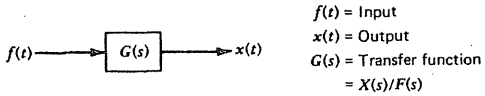


Figure 3.1 Block diagram for elementary analysis problem.

variables and lowercase symbols for the corresponding arguments of their probability density functions. This is the custom in most current books on probability. There is, however, a long tradition in engineering books on automatic control and linear systems analysis of using lowercase for time functions and uppercase for the corresponding Laplace or Fourier transforms. Hence, we are confronted with notational conflict. We will resolve this in favor of the traditional linear analysis notation, and from this point on we will use lowercase symbols for time signals—either random or deterministic—and uppercase for their transforms. This seems to be the lesser of the two evils. The reader will simply have to interpret the meaning of symbols such as $x(t)$, $f(t)$, and the like, within the context of the subject matter under discussion. This usually presents no problem. For example, with reference to Fig. 3.1, $g(t)$ would mean inverse transform of $G(s)$, and it clearly is a deterministic time function. On the other hand, the input and output, $f(t)$ and $x(t)$, will usually be random processes in the subsequent material.

Generally, analysis problems can be divided into two categories:

1. Stationary (steady-state) analysis. Here the input is assumed to be time stationary, and the system is assumed to have fixed parameters with a stable transfer function. This leads to a stationary output, provided the input has been present for a long period of time relative to the system time constants.
2. Nonstationary (transient) analysis. Here we usually consider the driving function as being applied at $t = 0$, and the system may be initially at rest or have nontrivial initial conditions. The response in this case is usually nonstationary. We note that analysis of unstable systems falls into this category, because no steady-state (stationary) condition will exist.

The similarity between these two categories and the corresponding ones in deterministic analysis should be apparent. Just as in circuit analysis, we would expect the transient solution to lead to the steady-state response as $t \to \infty$. However, if we are only interested in the stationary result, this is getting at the solution the "hard way." Much simpler methods are available for the stationary solution, and these will now be considered.

3.2 STATIONARY (STEADY-STATE) ANALYSIS

We assume in Fig. 3.1 that $G(s)$ represents a stable, fixed-parameter system and that the input is covariance (wide-sense) stationary with a known spectral func-

tion. In deterministic analysis, we know that if the input is Fourier transformable, the input spectrum is simply modified by $G(j\omega)$ in going through the filter. In the random process case, one interpretation of the spectral function is that it is proportional to the magnitude of the *square* of the Fourier transform. Thus the equation relating the input and output spectral functions is*

$$S_x(s) = G(s)G(-s)S_f(s) \tag{3.2.1}$$

Note that Eq. (3.2.1) is written in the s domain where the imaginary axis has the meaning of real angular frequency ω. If you prefer to write Eq. (3.2.1) in terms of ω, just replace s with $j\omega$. Equation (3.2.1) then becomes

$$\begin{aligned} S_x(j\omega) &= G(j\omega)G(-j\omega)S_f(j\omega) \\ &= |G(j\omega)|^2 S_f(j\omega) \end{aligned} \tag{3.2.2}$$

Because of the special properties of spectral functions, both sides of Eq. (3.2.2) work out to be real functions of ω. Also note that the autocorrelation function of the output can be obtained as the inverse Fourier transform of $S_x(s)$. Two examples will now illustrate the use of Eq. (3.2.1).

EXAMPLE 3.1

Consider a first-order low-pass filter with unity white noise as the input. With reference to Fig. 3.1, then

$$S_f(s) = 1$$

$$G(s) = \frac{1}{1 + Ts}$$

where T is the time constant of the filter. The output spectral function is then

$$\begin{aligned} S_x(s) &= \frac{1}{1 + Ts} \cdot \frac{1}{1 + T(-s)} \cdot 1 \\ &= \frac{(1/T)^2}{-s^2 + (1/T)^2} \end{aligned}$$

Or, in terms of real frequency ω,

$$S_x(j\omega) = \frac{(1/T)^2}{\omega^2 + (1/T)^2}$$

* See Problem 3.1 for a formal justification of Eq. (3.2.1).

This is sketched as a function of ω in Fig. 3.2. As would be expected, most of the spectral content is concentrated at low frequencies and then it gradually diminishes as $\omega \to \infty$.

It is also of interest to compute the mean-square value of the output. It is given by Eq. (2.7.17).

$$E(x^2) = \frac{1}{2\pi j} \int_{-j\infty}^{j\infty} \frac{1}{1 + Ts} \cdot \frac{1}{1 + T(-s)}\, ds \tag{3.2.3}$$

The integral of Eq. (3.2.3) is easily evaluated in this case by substituting $j\omega$ for s and using a standard table of integrals. This leads to

$$E(x^2) = \frac{1}{2T}$$

The "standard" table-of-integrals approach is of limited value, though, as will be seen in the next example. ■

EXAMPLE 3.2

Consider the input process to have an exponential autocorrelation function and the filter to be the same as in the previous example. Then

$$R_f(\tau) = \sigma^2 e^{-\beta|\tau|}$$

$$G(s) = \frac{1}{1 + Ts}$$

First, we transform R_f to obtain the input spectral function.

$$\mathfrak{F}[R_f(\tau)] = \frac{2\sigma^2\beta}{-s^2 + \beta^2}$$

The output spectral function is then

$$S_x(s) = \frac{2\sigma^2\beta}{-s^2 + \beta^2} \cdot \frac{(1/T)}{[s + (1/T)]} \cdot \frac{(1/T)}{[-s + (1/T)]} \tag{3.2.4}$$

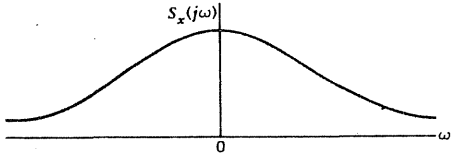


Figure 3.2 Spectral function for low-pass filter output with white-noise input.

Now, if we wish to find $E(x^2)$ in this case, it will involve integrating a function that is fourth-order in the denominator, and most tables of integrals will be of no help. We note, though, that the input spectral function can be factored and the terms of Eq. (3.2.4) can be rearranged as follows:

$$S_x(s) = \left[\frac{\sqrt{2\sigma^2\beta}}{(s+\beta)} \cdot \frac{(1/T)}{[s+(1/T)]}\right]\left[\frac{\sqrt{2\sigma^2\beta}}{(-s+\beta)} \cdot \frac{(1/T)}{[-s+(1/T)]}\right] \tag{3.2.5}$$

The first term has all its poles and zeros in the left half-plane, and the second term has mirror-image poles and zeros in the right half-plane. This regrouping of terms is known as *spectral factorization* and can always be done if the spectral function is rational in form (i.e., if it can be written as a ratio of polynomials in even powers of s).

Since special tables of integrals have been worked out for integrating complex functions of the type given by Eq. (3.2.5), we defer evaluating $E(x^2)$ until these have been presented in the next section. We note, however, that the concept of power spectral density presented in Section 2.7 is perfectly general, and its integral represents a mean-square value irrespective of whether or not the integral can be evaluated in closed form. There are many physical examples where one must resort to numerical integration to determine the "power" content of the signal (e.g., see Problem 3.28). ■

3.3 INTEGRAL TABLES FOR COMPUTING MEAN-SQUARE VALUE

In linear analysis problems, the spectral function can often be written as a ratio of polynomials in s^2. If this is the case, spectral factorization can be used to write the function in the form

$$S_x(s) = \frac{c(s)}{d(s)} \cdot \frac{c(-s)}{d(-s)} \tag{3.3.1}$$

where $c(s)/d(s)$ has all its poles and zeros in the left half-plane and $c(-s)/d(-s)$ has mirror-image poles and zeros in the right half-plane. No roots of $d(s)$ are permitted on the imaginary axis. The mean-square value of x can now be written as

$$E(x^2) = \frac{1}{2\pi j}\int_{-j\infty}^{j\infty} \frac{c(s)c(-s)}{d(s)d(-s)}\,ds \tag{3.3.2}$$

R. S. Phillips (4) was the first to prepare a table of integrals for definite integrals of the type given by Eq. (3.3.2). His table has since been repeated in many texts with a variety of minor modifications (5, 6, 7). An abbreviated table in terms of the complex s domain follows. An example will now illustrate the use of Table 3.1.

Table 3.1 Table of Integrals

$$I_n = \frac{1}{2\pi j}\int_{-j\infty}^{j\infty} \frac{c(s)c(-s)}{d(s)d(-s)}\,ds \tag{3.3.3}$$

$$c(s) = c_{n-1}s^{n-1} + c_{n-2}s^{n-2} + \cdots + c_0$$

$$d(s) = d_n s^n + d_{n-1}s^{n-1} + \cdots + d_0$$

$$I_1 = \frac{c_0^2}{2d_0d_1}$$

$$I_2 = \frac{c_1^2d_0 + c_0^2d_2}{2d_0d_1d_2}$$

$$I_3 = \frac{c_2^2d_0d_1 + (c_1^2 - 2c_0c_2)d_0d_3 + c_0^2d_2d_3}{2d_0d_3(d_1d_2 - d_0d_3)}$$

$$I_4 = \frac{c_3^2(-d_0^2d_3 + d_0d_1d_2) + (c_2^2 - 2c_1c_3)d_0d_1d_4 + (c_1^2 - 2c_0c_2)d_0d_3d_4 + c_0^2(-d_1d_4^2 + d_2d_3d_4)}{2d_0d_4(-d_0d_3^2 - d_1^2d_4 + d_1d_2d_3)}$$

EXAMPLE 3.3

The solution in Example 3.2 was brought to the point where the spectral function had been written in the form

$$S_x(s) = \left[\frac{\sqrt{2\sigma^2\beta}\cdot 1/T}{(s+\beta)(s+1/T)}\right]\left[\frac{\sqrt{2\sigma^2\beta}\cdot 1/T}{(-s+\beta)(-s+1/T)}\right] \tag{3.3.4}$$

Clearly, S_x has been factored properly with its poles separated into left and right half-plane parts. The mean-square value of x is given by

$$E(x^2) = \frac{1}{2\pi j}\int_{-j\infty}^{j\infty} S_x(s)\,ds \tag{3.3.5}$$

Comparing the form of $S_x(s)$ in Eq. (3.3.4) with the standard form given in Eq. (3.3.3), we see that

$$c(s) = \frac{\sqrt{2\sigma^2\beta}}{T}$$

$$d(s) = s^2 + (\beta + 1/T)s + \beta/T$$

Thus, we can use the I_2 integral of Table 3.1. The coefficients for this case are

$$c_1 = 0 \qquad d_2 = 1$$

$$c_0 = \frac{\sqrt{2\sigma^2\beta}}{T} \qquad d_1 = (\beta + 1/T)$$

$$d_0 = \beta/T$$

and $E(x^2)$ is then

$$E(x^2) = \frac{c_0^2}{2d_0d_1} = \frac{2\sigma^2\beta/T^2}{2(\beta/T)(\beta + 1/T)} = \frac{\sigma^2}{1 + \beta T}$$

3.4 PURE WHITE NOISE AND BANDLIMITED SYSTEMS

We are now in a position to demonstrate the validity of using the pure white-noise model in certain problems, even though white noise has infinite variance. This will be done by posing two hypothetical mean-square analysis problems:

1. Consider a simple first-order low-pass filter with bandlimited white noise as the input. Specifically, with reference to Fig. 3.1, let

$$S_f(j\omega) = \begin{cases} A, & |\omega| \le \omega_c \\ 0, & |\omega| > \omega_c \end{cases} \tag{3.4.1}$$

$$G(s) = \frac{1}{1 + Ts} \tag{3.4.2}$$

2. Consider the same low-pass filter as in problem 1, but with pure white noise as the input:

$$S_f(j\omega) = A, \qquad \text{for all } \omega \tag{3.4.3}$$

$$G(s) = \frac{1}{1 + Ts} \tag{3.4.4}$$

Certainly, problem 1 is physically plausible because bandlimited white noise has finite variance. Conversely, problem 2 is not because the input has infinite variance. The preceding theory enables us to evaluate the mean-square value of the output for both problems. As a matter of convenience, we do this in the real frequency domain rather than the complex s domain.

Problem 1:

$$S_x(j\omega) = \begin{cases} \dfrac{1}{1 + (T\omega)^2}, & |\omega| \le \omega_c \\ 0, & |\omega| > \omega_c \end{cases} \tag{3.4.5}$$

$$E(x^2) = \frac{1}{2\pi}\int_{-\omega_c}^{\omega_c} \frac{A}{1 + (T\omega)^2}\, d\omega = \frac{A}{\pi T}\tan^{-1}(\omega_c T) \tag{3.4.6}$$

Problem 2:

$$S_x(j\omega) = \frac{A}{1 + (T\omega)^2}, \qquad \text{for all } \omega \tag{3.4.7}$$

$$E(x^2) = \frac{1}{2\pi}\int_{-\infty}^{\infty} \frac{A}{1 + (T\omega)^2}\, d\omega = \frac{A}{\pi T}\tan^{-1}(\infty) = \frac{A}{2T} \tag{3.4.8}$$

Now, we see by comparing the results given by Eqs. (3.4.6) and (3.4.8) that the difference is just that between $\tan^{-1}(\omega_c T)$ and $\tan^{-1}(\infty)$. The bandwidth of the input is ω_c and the filter bandwidth is $1/T$. Thus, if their ratio is large, $\tan^{-1}(\omega_c T) \approx \tan^{-1}(\infty)$. For a ratio of 100:1, the error is less than 1 percent. Thus, if the input spectrum is flat considerably out beyond the point where the system response is decreasing at 20 db/decade (or faster), there is relatively little error introduced by assuming that the input is flat out to infinity. The resulting simplification in the analysis is significant.

3.5 NOISE EQUIVALENT BANDWIDTH

In filter theory, it is sometimes convenient to think of an idealized filter whose frequency response is unity over a prescribed bandwidth B (in hertz) and zero outside this band. This response is depicted in Fig. 3.3*a*. If this ideal filter is driven by white noise with amplitude A, its mean-square response is

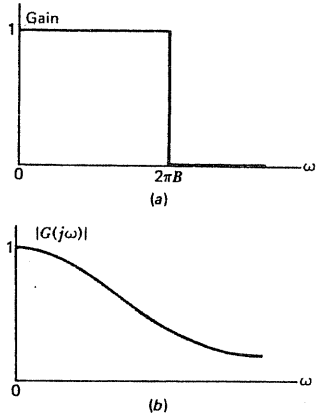


Figure 3.3 Ideal and actual filter responses. (*a*) Ideal. (*b*) Actual.

$$E(x^2) \quad \text{(ideal)} = \frac{1}{2\pi}\int_{-2\pi B}^{2\pi B} A\,d\omega = 2AB \tag{3.5.1}$$

Next, consider an actual filter $G(s)$ whose gain has been normalized to yield a peak response of unity. An example is shown in Fig. 3.3*b*. The mean-square response of the actual filter to white noise of amplitude A is given by

$$E(x^2) \quad \text{(actual)} = \frac{1}{2\pi j}\int_{-j\infty}^{j\infty} AG(s)G(-s)\,ds \tag{3.5.2}$$

Now, if we wish to find the idealized filter that will yield this same response, we simply equate $E(x^2)$ (ideal) and $E(x^2)$ (actual) and solve for the bandwidth that gives equality. The resultant bandwidth B is known as the *noise equivalent bandwidth*. It may, of course, be written explicitly as

$$B \quad \text{(in hertz)} = \frac{1}{2}\left[\frac{1}{2\pi j}\int_{-j\infty}^{j\infty} G(s)G(-s)\,ds\right] \tag{3.5.3}$$

EXAMPLE 3.4

Suppose we wish to find the noise equivalent bandwidth of the second-order low-pass filter

$$G(s) = \frac{1}{(1 + Ts)^2}$$

Since the peak response of $G(s)$ occurs at zero frequency and is unity, the gain scale factor is set properly. We must next evaluate the integral in brackets in Eq. (3.5.3). Clearly, $G(s)$ is second-order in the denominator, and therefore we use I_2 of the integral tables given in Section 3.3. The coefficients in this case are

$$c_1 = 0 \qquad d_2 = T^2$$
$$c_0 = 1 \qquad d_1 = 2T$$
$$d_0 = 1$$

and thus I_2 is

$$I_2 = \frac{c_0^2}{2d_0d_1} = \frac{1}{4T}$$

The filter's noise equivalent noise bandwidth is then

$$B = \frac{1}{8T}\,\text{Hz}$$

This says, in effect, that an idealized filter with a bandwidth of $1/8T$ Hz would pass the same amount of noise as the actual second-order filter. ■

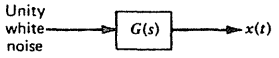


Figure 3.4 Shaping filter.

3.6 SHAPING FILTER

With reference to Fig. 3.4, we have seen that the output spectral function can be written as

$$S_x(s) = 1 \cdot G(s)G(-s) \tag{3.6.1}$$

If $G(s)$ is minimum phase and rational in form,* Eq. (3.6.1) immediately provides a factored form for $S_x(s)$ with poles and zeros automatically separated into left and right half-plane parts.

Clearly, we can reverse the analysis problem and pose the question: What minimum-phase transfer function will shape unity white noise into a given spectral function $S_x(s)$? The answer should be apparent. If we can use spectral factorization on $S_x(s)$, the part with poles and zeros in the left half-plane provides the appropriate shaping filter. This is a useful concept, both as a mathematical artifice and also as a physical means of obtaining a noise source with desired spectral characteristics from a wideband source.

EXAMPLE 3.5

Suppose we wish to find the shaping filter that will shape unity white noise into noise with a spectral function

$$S_x(j\omega) = \frac{\omega^2 + 1}{\omega^4 + 64} \tag{3.6.2}$$

First, we write S_x in the s domain as

$$S_x(s) = \frac{-s^2 + 1}{s^4 + 64} \tag{3.6.3}$$

Next, we find the poles and zeros of S_x.

$$\text{Zeros} = \pm 1$$

$$\text{Poles} = -2 \pm j2, \qquad 2 \pm j2$$

Finally, we group together left and right half-plane parts. $S_x(s)$ can then be written as

* This condition requires $G(s)$ to have a finite number of poles and zeros, all of which must be in the left half-plane.

$$S_x(s) = \frac{s+1}{s^2+4s+8} \cdot \frac{-s+1}{s^2-4s+8} \tag{3.6.4}$$

The desired shaping filter is then

$$G(s) = \frac{s+1}{s^2+4s+8} \tag{3.6.5}$$

■

3.7 NONSTATIONARY (TRANSIENT) ANALYSIS— INITIAL CONDITION RESPONSE

As mentioned previously, the response of a linear system may always be considered as a superposition of an initial-condition part and a driven part. The response due to the initial conditions is often ignored in tutorial discussions, but it should not be because there are many applications where the initial conditions are properly modeled as random variables. If this is the case, one simply solves the problem using standard deterministic methods leaving the initial conditions in the solution in general terms. An example will illustrate the procedure.

EXAMPLE 3.6

Suppose the circuit shown in Fig. 3.5 has been in operation for a long time with the switch open, and then the switch is closed at a random time, which we denote as $t = 0$. We are asked to describe the voltage across the capacitor after $t = 0$.

We first look at the steady-state condition just prior to closing the switch. The transfer function relating the capacitor voltage to the white-noise source is given by

$$G(s) = \frac{1}{1+2s} \tag{3.7.1}$$

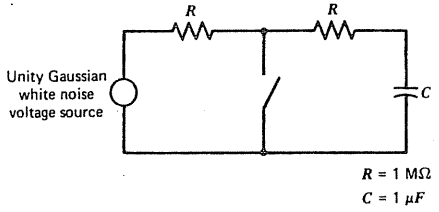


Figure 3.5 Circuit for Example 3.6.

Therefore, by using the methods of Sections 3.2 and 3.3, the mean-square value of the capacitor voltage v_c is found to be

$$E(v_c^2) = \frac{1}{2\pi j}\int_{-j\infty}^{j\infty} \frac{1}{1+2s} \cdot \frac{1}{1-2s}\, ds = \frac{1}{4} \tag{3.7.2}$$

We have now established that the initial condition is a normal random variable with zero mean and a variance of $\frac{1}{4}$.

After the switch is closed, the system differential equation is

$$C\dot{v}_c + \frac{1}{R} v_c = 0 \tag{3.7.3}$$

Taking the Laplace transform of both sides yields

$$C[sV_c(s) - v_c(0)] + \frac{1}{R} V_c(s) = 0$$

or

$$V_c(s) = \frac{v_c(0)}{s + (1/RC)} \tag{3.7.4}$$

The explicit expression for the time-domain waveform is now obtained by taking the inverse transform of Eq. (3.7.4). It is

$$v_c(t) = v_c(0)e^{-t/RC} \tag{3.7.5}$$

where $v_c(0)$ is a random variable characterized by $N(0, \frac{1}{4})$. Note that the solution of the initial-condition problem leads to a deterministic random process; that is, the process has deterministic structure and any particular realization of the process is exactly predictable once the initial condition is known.

The mean-square value of a random process is usually of prime interest. In this case, it is easily computed as

$$\begin{aligned} E(v_c^2) &= E[v_c(0)e^{-t/RC}]^2 \\ &= e^{-2t/RC} \cdot E[v_c^2(0)] \\ &= \tfrac{1}{4}e^{-2t/RC} \end{aligned} \tag{3.7.6}$$

It is, of course, a function of time. ■

The extension of the procedure of Example 3.6 to more complicated situations is fairly obvious, so this will not be pursued further.

3.8 NONSTATIONARY (TRANSIENT) ANALYSIS—FORCED RESPONSE

The block diagram of Fig. 3.1 is repeated as Fig. 3.6 with the addition of a switch in the input. Imagine the system to be initially at rest, and then close the switch at $t = 0$. A transient response takes place in the stochastic problem just as in the corresponding deterministic problem. If the input $f(t)$ is a nondeterministic random process, we would expect the response also to be nondeterministic, and its autocorrelation function may be computed in terms of the input autocorrelation function. This is done as follows.

The system response can be written as a convolution integral

$$x(t) = \int_0^t g(u)f(t - u)\,du \tag{3.8.1}$$

where $g(u)$ is the inverse Laplace transform of $G(s)$ and is usually referred to as the system weighting function. To find the autocorrelation function, we simply evaluate $E[x(t_1)x(t_2)]$.

$$\begin{aligned} R_x(t_1, t_2) &= E[x(t_1)x(t_2)] \\ &= E\left[\int_0^{t_1} g(u)f(t_1 - u)\,du \int_0^{t_2} g(v)f(t_2 - v)\,dv\right] \\ &= \int_0^{t_2}\int_0^{t_1} g(u)g(v)E[f(t_1 - u)f(t_2 - v)]\,du\,dv \end{aligned} \tag{3.8.2}$$

Now, if $f(t)$ is stationary, Eq. (3.8.2) can be written as

$$R_x(t_1, t_2) = \int_0^{t_2}\int_0^{t_1} g(u)g(v)R_f(u - v + t_2 - t_1)\,du\,dv \tag{3.8.3}$$

and we now have an expression for the output autocorrelation function in terms of the input autocorrelation function and system weighting function.

Equation (3.8.3) is difficult to evaluate except for relatively simple systems. Thus, we are often willing to settle for less information about the response and just compute its mean-square value. This is done by letting $t_2 = t_1 = t$ in Eq. (3.8.3) with the result

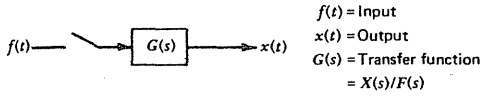


Figure 3.6 Block diagram for nonstationary analysis problem.

$$E[x^2(t)] = \int_0^t\int_0^t g(u)g(v)R_f(u - v)\,du\,dv \tag{3.8.4}$$

Three examples will now illustrate the use of Eqs. (3.8.3) and (3.8.4).

EXAMPLE 3.7

Let $G(s)$ be a first-order low-pass filter, and let $f(t)$ be white noise with amplitude A. Then

$$G(s) = \frac{1}{1 + Ts}$$

$$S_f(\omega) = A$$

Taking inverse transforms gives

$$g(u) = \frac{1}{T}e^{-u/T}$$

$$R_f(\tau) = A\delta(\tau)$$

Next, substituting in Eq. (3.8.4) yields

$$\begin{aligned} E[x^2(t)] &= \int_0^t\int_0^t \frac{A}{T^2}e^{-u/T}e^{-v/T}\delta(u - v)\,du\,dv \\ &= \frac{A}{T^2}\int_0^t e^{-2v/T}\,dv \\ &= \frac{A}{2T}[1 - e^{-2t/T}] \end{aligned} \tag{3.8.5}$$

Note that as $t \to \infty$, the mean-square value approaches $A/2T$, which is the same result obtained in Section 3.2 using spectral analysis methods. ■

(See p. 129/30.
See Example 3.1 on p. 13 and Section 3.4, problems on p. 135)

EXAMPLE 3.8

Let $G(s)$ be an integrator with zero initial conditions, and let $f(t)$ be a Gauss–Markov process with variance σ^2 and time constant $1/\beta$. We desire the mean-square value of the output x. The transfer function and input autocorrelation function are

$$G(s) = \frac{1}{s} \quad \text{or} \quad g(u) = 1$$

and

$$R_f(\tau) = \sigma^2 e^{-\beta|\tau|}$$

Next, we use Eq. (3.8.4) to obtain $E[x^2(t)]$.

$$E[x^2(t)] = \int_0^t \int_0^t 1 \cdot 1 \cdot \sigma^2 e^{-\beta|u-v|}\, du\, dv \tag{3.8.6}$$

Some care is required in evaluating Eq. (3.8.6) because one functional expression for $e^{-\beta|u-v|}$ applies for $u > v$, and a different one applies for $u < v$. This is shown in Fig. 3.7. Recognizing that the region of integration must be split into two parts, we have

$$E[x^2(t)] = \int_0^t \int_0^v \sigma^2 e^{-\beta(v-u)}\, du\, dv + \int_0^t \int_v^t \sigma^2 e^{-\beta(u-v)}\, du\, dv \tag{3.8.7}$$

Since there is symmetry in the two integrals of Eq. (3.8.7), we can simply evaluate the first one and multiply by 2. The mean-square value of x is then

$$\begin{aligned} E[x^2(t)] &= 2\int_0^t \int_0^v \sigma^2 e^{-\beta(v-u)}\, du\, dv = 2\int_0^t \sigma^2 e^{-\beta v} \int_0^v e^{\beta u}\, du\, dv \\ &= \frac{2\sigma^2}{\beta} \int_0^t e^{-\beta v}(e^{\beta v} - 1)\, dv \\ &= \frac{2\sigma^2}{\beta^2} [\beta t - (1 - e^{-\beta t})] \end{aligned} \tag{3.8.8}$$

Note that $E[x^2(t)]$ increases without bound as $t \to \infty$. This might be expected because an integrator is an unstable system. ■

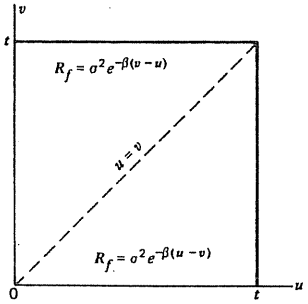


Figure 3.7 Regions of integration for Example 3.8.

EXAMPLE 3.9

As our final example, we find the autocorrelation function of the output of a simple integrator driven by unity-amplitude Gaussian white noise. The transfer function and input autocorrelation function are

$$G(s) = \frac{1}{s} \qquad \text{or} \qquad g(u) = 1$$

$$R_f(\tau) = \delta(\tau)$$

We obtain $R_x(t_1, t_2)$ from Eq. (3.8.3):

$$R_x(t_1, t_2) = \int_0^{t_2} \int_0^{t_1} 1 \cdot 1 \cdot \delta(u - v + t_2 - t_1)\, du\, dv \tag{3.8.9}$$

The region of integration for this double integral is shown in Fig. 3.8 for $t_2 > t_1$. The argument of the Dirac delta function in Eq. (3.8.9) is zero along the dashed line in the figure. It can be seen that it is convenient to integrate first with respect to v if $t_2 > t_1$ as shown in the figure. Considering this case first (i.e., $t_2 > t_1$), we have

$$R_x(t_1, t_2) = \int_0^{t_1} \int_0^{t_2} \delta(u - v + t_2 - t_1)\, dv\, du = \int_0^{t_1} 1 \cdot du = t_1$$

Similarly, when $t_2 < t_1$,

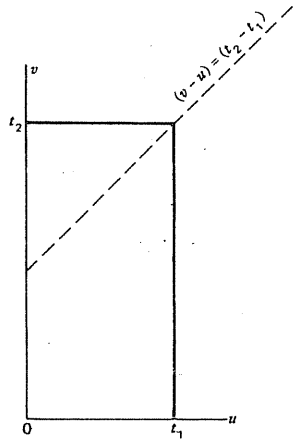


Figure 3.8 Region of integration for Example 3.9.

$$R_x(t_1, t_2) = t_2$$

The final result is then

$$R_x(t_1, t_2) = \begin{cases} t_1, & t_2 \geq t_1 \\ t_2, & t_2 < t_1 \end{cases} \tag{3.8.10}$$

Note that this is the same result obtained for the Wiener process in Chapter 2. ■

In concluding this section, we might comment that if the transient response includes both forced and initial-condition components, the total response is just the superposition of the two. The mean-square value must be evaluated with care, though, because the total mean-square value is the sum of the two only when the crosscorrelation is zero. If the crosscorrelation between the two responses is not zero, it must be properly accounted for in computing the mean-square value.

3.9 DISCRETE-TIME PROCESS MODELS AND ANALYSIS

Our emphasis thus far has been on continuous-time random signals and the associated response of linear systems to such signals. There is a discrete-time counterpart to all of this, and we shall now take a brief look at discrete-time random processes. In order to keep the discussion as brief as possible, we will confine our attention to single-input, single-output constant-parameter systems. Then later, in Chapter 5, we will consider multiple input–output systems.

It was mentioned in Section 3.6 that a continuous-time process with a rational spectral density can be thought of as the result of passing white noise through a linear filter. The filter, in turn, specifies an input–output differential equation relationship between the response and the input white noise. This equation has the general form

$$\begin{aligned}(D^n + a_{n-1}D^{n-1} + a_{n-2}D^{n-2} + \cdots + a_0)x(t) \\ = (b_mD^m + b_{m-1}D^{m-1} + \cdots + b_0)u(t), \qquad m = n - 1\end{aligned} \tag{3.9.1}$$

where D is derivative operator and $u(t)$ is unity white noise. (The scale factor on the input is absorbed in the b coefficients.) The system transfer function $G(s)$ is then

$$G(s) = \frac{b_ms^m + b_{m-1}s^{m-1} + \cdots + b_0}{s^n + a_{n-1}s^{n-1} + a_{n-2}s^{n-2} + \cdots + a_0} \tag{3.9.2}$$

It should be apparent that the continuous-time process $x(t)$ generated by Eq. (3.9.1) can be either stationary or nonstationary, depending on the stability characteristics of $G(s)$ and the initial conditions. [We assume here that the $x(t)$ process is initiated at $t = 0$.]

In the discrete-time world, the input–output relationship corresponding to Eq. (3.9.1) is a difference equation, and it has the general form

$$\begin{aligned}y(k + n) + \alpha_{n-1}y(k + n - 1) + \alpha_{n-2}y(k + n - 2) + \cdots + \alpha_0y(k) \\ = \beta_mw(k + m) + \beta_{m-1}w(k + m - 1) + \cdots + \beta_0w(k), \qquad m = n - 1\end{aligned} \tag{3.9.3}$$

The corresponding quantities in the continuous and discrete models can be summarized as follows:

Continuous-time	Discrete-time
Continuous time t	Integer index k, $k = 0, 1, 2, \ldots$
Response $x(t)$	Response $y(k)$
nth derivative	n units of advance
Input continuous unity white noise $u(t)$	Input discrete white sequence $w(k)$ with unit variance

Note that we have intentionally used different symbols in the continuous and discrete models in order to emphasize that there need be no direct connection between the two models. The difference equation, Eq. (3.9.3), is called the ARMA model for the $y(k)$ process; the left side of the equation is the AR part of the model (for autoregressive), and the right side is the MA part (for moving average).* Just as in the continuous problem, we can think of generating a sample response as the result of inputting a particular $w(k)$ sequence (chosen by chance, of course) and then solving for the resulting $y(k)$ sequence. We cannot write out an explicit solution, but we can conceptually think of the ARMA model as shaping the input white sequence into a corresponding colored sequence. Also, just as in the continuous case we can generate either stationary or nonstationary processes, depending on the stability characteristics of the ARMA difference equation. Z-transforms play the same role in difference equations that Laplace transforms do in differential equations, so they can be used in examining the stability of discrete-time systems. Two examples will illustrate this.

EXAMPLE 3.10

Consider the simple first-order ARMA model

$$y(k + 1) - y(k) = w(k), \qquad k = 0, 1, 2, \ldots \tag{3.9.4}$$

* We are being somewhat restrictive in our ARMA model in that k is allowed to increment only in a positive sense beginning at zero, and that the order of the MA part of the model is less than the order of the AR part. The resulting transfer function in the z-domain is then strictly proper. Also, the transform algebra used in the subsequent analysis can be single-sided, in contrast to two-sided, because k is nonnegative. All of this is in anticipation of our particular use of the ARMA model in Chapter 5. See references 8 and 9 for more on the ARMA model.

Note that the MA part of the model is trivial in this case because there is only the $w(k)$ term. We inquire as to the stability of the $y(k)$ process. All we need to do is look at the transfer function of the system in the z-domain. Toward this end we take the z-transform of both sides of Eq. (3.9.4) and form the output-to-input ratio.

$$zY(z) - Y(z) = W(z) \tag{3.9.5}$$

$$\frac{Y(z)}{W(z)} = \frac{1}{z - 1} \tag{3.9.6}$$

Immediately, we see that we have a pole on the unit circle at $z = 1$, so the system is unstable. In this situation we would expect the output $y(k)$ to be nonstationary. This example, of course, is the sampled form of a Wiener process, provided the initial $y(0)$ is zero and $w(k)$ is a white Gaussian sequence. We know that its variance increases linearly with the number of steps from the origin. ▩

EXAMPLE 3.11

We will now look at a slightly more complicated example where stability is not obvious at first glance. Let the ARMA model be

$$y(k + 2) - y(k + 1) + .5y(k) = .5w(k + 1) + .25w(k), \qquad k = 0, 1, 2, \ldots \tag{3.9.7}$$

Again, we inquire about the stability of $y(k)$; and, just as in the previous example, we can form the system transfer function in the z-domain by taking the z-transform of both sides of Eq. (3.9.7).

$$z^2Y(z) - zY(z) + .5Y(z) = .5zW(z) + .25W(z)$$

Or

$$\frac{Y(z)}{W(z)} = \frac{.5z + .25}{z^2 - z + .5} \tag{3.9.8}$$

Solving for the roots of the denominator tells us that we have a pair of complex poles at $-.5 \pm j.5$. They are within the unit circle (and the zero at $-.5$ does not affect the stability), so we would expect the output to reach a stationary condition after the transient dies out. Furthermore, the pole locations also provide information about how fast this should occur. We will not pursue this further here (see Example 5.6). ▩

We could go on and on exploiting the analogy between continuous- and discrete-time models, but much of this is better done in a state-space setting with vector models. Thus, further discussion of this will be deferred until Chapter 5. One further comment is in order here before closing, though. We have

referred to our discrete models here as discrete-*time* models, but time may not actually be important in many physical applications. For example, if one looks at a sequence of rolls of a pair of dice, it is only the sum of the dots that matters in many games, and the result of such an experiment is simply a sequence of discrete events. There is nothing "in between," so to speak, and time is not of the essence! On the other hand, there are many situations where the discrete-time sequence arises as a result of sampling a continuous-time process, and the sampling interval and other time-related matters are quite important. Both of these physical situations fit into the mathematical framework of what we have called discrete-time systems, and we append the word *time* just as a reminder that the sequences being considered are discrete in their argument and not in the random variable space.

3.10 SUMMARY

The mean-square value and spectral density function (or autocorrelation function) are the output process descriptors of prime interest. This is partially a matter of mathematical convenience, because they are usually the only descriptors that can be readily computed. If only the stationary solution is desired, the output spectral density is readily computed from Eq. (3.2.1) or (3.2.2). The mean-square value may then be obtained from the integral of the spectral function. Integral tables are available to assist in this task, provided the transfer and input spectral functions are both in rational form.

In transient problems, the portion of the system response due to random initial conditions is computed using standard deterministic methods. The resulting response is always a deterministic random process. The autocorrelation function of the driven response may be computed from Eq. (3.8.3). In many cases, the computation is quite involved; therefore, only the mean-square value is computed. This computation is relatively simple and is given by Eq. (3.8.4). The total response in the transient problem is, of course, the superposition of both the initial-condition and driven parts.

Linear multiple-input, multiple-output problems were not discussed in this chapter. Complicated problems of this type are best handled by state-variable methods, and discussion of such systems will be deferred to Chapter 5. Simple multiple-input, multiple-output problems are sometimes manageable using scalar methods, and in these cases one simply uses superposition in computing the various responses. One must, of course, be careful in computing spectral functions and mean-square values to account properly for any nontrivial crosscorrelations that may exist.

One further comment about process models for physical systems is in order. The models, be they continuous or discrete, must come from somewhere, of course. Most often they are derived from experimental data, but occasionally they come from purely theoretical considerations (perhaps backed up by experimental evidence). Analysis of experimental data is a separate subject in its own right, and much has been written about it (8, 9, 10). Usually, the end result of

the model formulation is either a spectral description of the process (or equivalently, its correlation structure) or a discrete time-series model, that is, an ARMA model. Often the experimentally derived models are themselves uncertain because of limited data or other experimental vagaries. Be that as it may, reasonable process models must be assumed before the engineer/analyst can go forward and proceed with the analysis and system design. We are primarily concerned here with the second half of the problem, that is, the analysis and design part. Thus, we will assume in subsequent chapters that somehow or other reasonable process models have been provided, and we simply pick up the problem at that point and proceed to look at analysis and design methods that are based on minimizing the mean-square error.

PROBLEMS

3.1 In section 3.1 the equation relating the input and output spectral densities $[S_x(j\omega) = |G(j\omega)|^2 S_f(j\omega)]$ was justified with heuristic arguments. This can be formalized by proceeding through the following steps:

(a) Write the output $x(t)$ as a convolution integral using Fourier rather than Laplace transforms.
(b) Do likewise for the shifted output $x(t + \tau)$.
(c) Multiply the expressions for $x(t)$ and $x(t + \tau)$ and (symbolically) form the expectation of the product.
(d) Now note that the Fourier transform of the autocorrelation function is the spectral density; transform both sides, interchange the order of integration and the desired result is apparent.

Proceed through the steps just described and formally justify the $S_x(j\omega) = |G(j\omega)|^2 S_f(j\omega)$ formula.

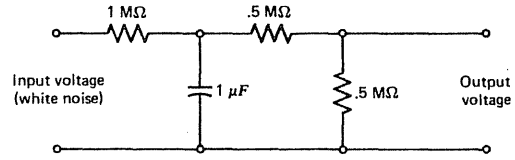
3.2 Find the steady-state mean-square value of the output for the following filters. The input is white noise with a spectral density amplitude A.

(a) $G(s) = \dfrac{Ts}{(1 + Ts)^2}$

(b) $G(s) = \dfrac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2}$

(c) $G(s) = \dfrac{s + 1}{(s + 2)^2}$

3.3 A white-noise process having a spectral density amplitude of A is applied to the circuit shown. The circuit has been in operation for a long time. Find the mean-square value of the output voltage.



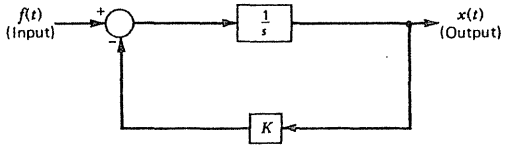
Problem 3.3

3.4 The input to the feedback system shown is a stationary Markov process with an autocorrelation function

$$R_f(\tau) = \sigma^2 e^{-\beta|\tau|}$$

The system is in a stationary condition.

(a) What is the spectral density function of the output?
(b) What is the mean-square value of the output?



Problem 3.4

3.5 Consider the nonminimum phase filter

$$G(s) = \frac{1 - T_1 s}{1 + T_2 s}$$

driven with a stationary Gauss–Markov process with an autocorrelation function $R(\tau) = \sigma^2 e^{-\beta|\tau|}$. Find:

(a) The spectral density function of the output.
(b) The mean-square value of the output.

3.6 Find the steady-state mean-square value of the output for a first-order low-pass filter [i.e., $G(s) = 1/(1 + Ts)$] if the input has an autocorrelation function of the form

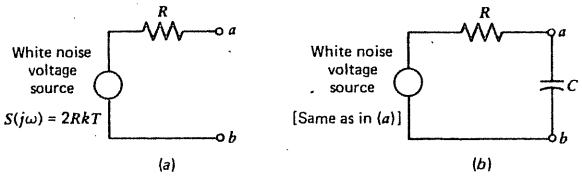
$$R(\tau) = \begin{cases} \sigma^2(1 - \beta|\tau|), & -\dfrac{1}{\beta} \le \tau \le \dfrac{1}{\beta} \\ 0, & |\tau| > \dfrac{1}{\beta} \end{cases}$$

[*Hint:* The input spectral function is irrational so the integrals given in Table 3.1 are of no help here. One approach is to write the integral expression for $E(X^2)$ in terms of real ω rather than s and then use conventional integral tables. Also, those familiar with residue theory will find that the integral can be evaluated by the method of residues.]

3.7 Thermal noise in a metallic resistor is sometimes modeled as a white-noise voltage source in series with the resistance R of the resistor (11, 12). This is shown in the figure on the next page, along with the parameters describing the spectral amplitude of the noise source. At room temperature the flat spectrum approximation is reasonably accurate from zero frequency out to the infrared range. Clearly, in the idealized model of part (*a*), the voltage from a to b would be infinity, which is physically impossible. The model is still useful, though, because there is always some shunt capacitance associated with the load connected from a to b. If nothing else, the parasitic capacitance of the resistor leads is sufficient to cause the spectral function to "roll off" at 20 db/decade and thus

cause the output to be bounded. This is shown in part (*b*) of the figure. Now, to demonstrate the validity of the white-noise model, consider an example where $R = 1 \times 10^6\ \Omega$ and $C = 1 \times 10^{-12}$ F (a plausible value for parasitic capacitance). Also, assume the temperature is room temperature, about 290 K.

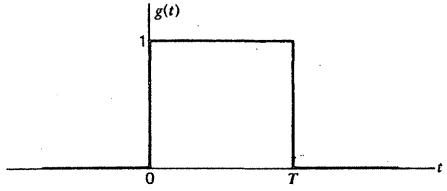
(a) Find the rms voltage across the capacitor C.
(b) Find the half-power frequency in hertz. (Defined as the frequency at which the spectral function is half its value at zero frequency.)
(c) Based on the result of part (b), would you think it reasonable to consider this noise source as being flat (i.e., pure white) in most electronic-circuit applications? Explain briefly.



R = Resistance of metallic resistor (ohms)
k = Boltzmann constant $\approx 1.38 \times 10^{-23}$ (joules/deg)
T = Temperature (degrees Kelvin)

Problem 3.7

3.8 Consider a linear filter whose weighting function is shown in the figure. (This filter is sometimes referred to as a finite-time integrator.) The input to the filter is white noise with a spectral density amplitude A, and the filter has been in operation a long time. What is the mean-square value of the output?



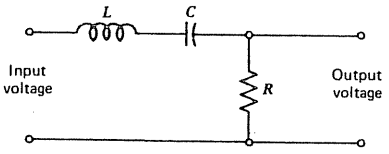
Problem 3.8 Filter weighting function.

3.9 Find the shaping filter that will shape unity white noise into noise with a spectral function

$$S(j\omega) = \frac{\omega^2 + 1}{\omega^4 + 8\omega^2 + 16}$$

3.10 A series resonant circuit is shown in the figure. Let the resistance R be small such that the circuit is sharply tuned (i.e., high Q or very low damping ratio). Find the noise equivalent bandwidth for this circuit and express it in terms of the damping ratio ζ and the natural undamped resonant frequency ω_r (i.e., $\omega_r = 1/\sqrt{LC}$). Note that the "ideal" response in this case is a unity-gain rectangular pass band centered about ω_r. Also find the usual half-power bandwidth and compare this with the noise equivalent bandwidth. (Half-power bandwidth

is defined to be the frequency difference between the two points on the response curve that are "down" by a factor of $1/\sqrt{2}$ from the peak value. It is useful to approximate the resonance curve as being symmetric about the peak for this part of the problem.)



Problem 3.10

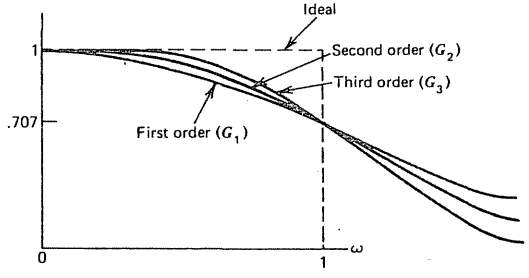
3.11 The transfer functions and corresponding bandpass characteristics for first-, second-, and third-order Butterworth filters are shown in the figure below. These filters are said to be "maximally flat" at zero frequency with each successive higher-order filter more nearly approaching the ideal curve than the previous one. All three filters have been normalized such that all responses intersect the −3-db point at 1 rad/sec (or $1/2\pi$ Hz).

(a) Find the noise equivalent bandwidth for each of the filters.
(b) Insofar as noise suppression is concerned, is there much to be gained by using anything higher-order than a third-order Butterworth filter?

3.12 Find the mean-square value of the output (averaged in an ensemble sense) for the following transfer functions. In both cases, the initial conditions are zero and the input $f(t)$ is applied at $t = 0$.

(a) $G(s) = \dfrac{1}{s^2}, \qquad R_f(\tau) = A\delta(\tau)$

(b) $G(s) = \dfrac{1}{s^2 + \omega_0^2}, \qquad R_f(\tau) = A\delta(\tau)$



(a)

$$G_1(s) = \frac{1}{s+1}$$

$$G_2(s) = \frac{1}{s^2 + \sqrt{2}s + 1}$$

$$G_3(s) = \frac{1}{(s+1)(s^2+s+1)}$$

(b)

Problem 3.11 (*a*) Responses of three Butterworth filters. (*b*) Transfer functions of Butterworth filters.

3.13 A certain linear system is known to satisfy the following differential equation:

$$\ddot{x} + a\dot{x} = f(t)$$

$$x(0) = \dot{x}(0) = 0$$

where $x(t)$ is the response and $f(t)$ is the input that is applied at $t = 0$. If $f(t)$ is white noise with spectral density amplitude A, what is the mean-square value of the response $x(t)$?

3.14 Consider a simple first-order low-pass filter whose transfer function is

$$G(s) = \frac{1}{1 + 10s}$$

The input to the filter is initiated at $t = 0$, and the filter's initial condition is zero. The input is given by

$$f(t) = Au(t) + n(t)$$

where

$u(t)$ = unit step function
A = random variable with uniform distribution from 0 to 1
$n(t)$ = unity Gaussian white noise

Find:

(a) The mean, mean square, and variance of the output evaluated at $t = .1$ sec.
(b) Repeat (a) for the steady-state condition (i.e., for $t = \infty$).

(*Hint:* Since the system is linear, superposition may be used in computing the output. Note that the deterministic component of the input is written explicitly in functional form. Therefore, deterministic methods may be used to compute the portion of the output due to this component. Also, remember that the mean-square value and the variance are not the same if the mean is nonzero.)

3.15 A signal is known to have the following form:

$$s(t) = a_0 + n(t)$$

where a_0 is an unknown constant and $n(t)$ is a stationary noise process with a known autocorrelation function

$$R_n(\tau) = \sigma^2 e^{-\beta|\tau|}$$

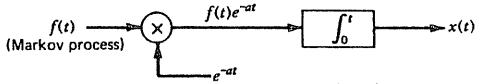
It is suggested that a_0 can be estimated by simply averaging $s(t)$ over a finite interval of time T. What would be the rms error in the determination of a_0 by this method?

(*Note:* The *root* mean square rather than mean square value is requested in this problem.)

3.16 In the figure shown, $f(t)$ is a time stationary random process whose autocorrelation function is

$$R_f(\tau) = \sigma^2 e^{-\beta|\tau|}$$

The input $f(t)$ is first multiplied by e^{-at} and then integrated, beginning at $t = 0$. The initial value of the integrator is zero. Find the mean-square value of $x(t)$.

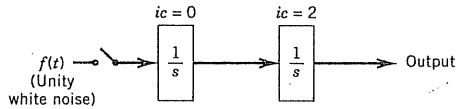


Problem 3.16

3.17 Consider an integrator whose initial output at $t = 0$ is a Gaussian random variable with zero mean and variance σ^2. A Gaussian white-noise input with spectral amplitude A is applied at $t = 0$. What is the mean-square value of the output as a function of time?

3.18 Unity Gaussian white noise $f(t)$ is applied to the cascaded combination of integrators shown in the figure. The switch is closed at $t = 0$. The initial condition for the first integrator is zero, and the second integrator has two units as its initial value.

(a) What is the mean-square value of the output at $t = 2$ sec?
(b) Sketch the probability density function for the output evaluated at $t = 2$ sec.

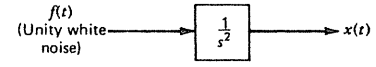


Problem 3.18

3.19 Consider the random process defined by the transfer function shown in the figure. The input $f(t)$ is Gaussian white noise with unity spectral amplitude, and the process is started at $t = 0$ with zero initial conditions. The autocorrelation function of the output $x(t)$ is defined as

$$R_x(t_1, t_2) = E[x(t_1)x(t_2)], \qquad t_1 \text{ and } t_2 > 0$$

Find $R_x(t_1, t_2)$.



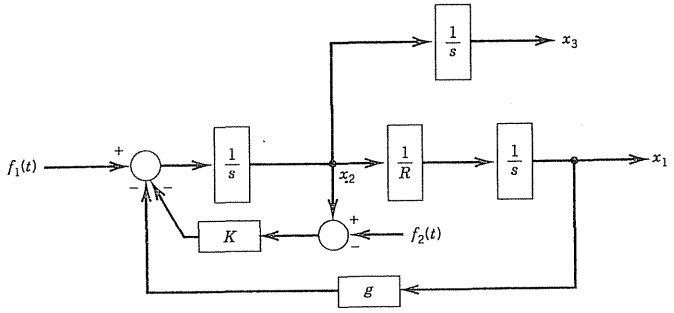
Problem 3.19

3.20 Consider again the filter with a rectangular weighting function discussed in Problem 3.8. Consider the filter to be driven with unity Gaussian white noise, which is initiated at $t = 0$ with zero initial conditions.

(a) Find the mean-square response in the interval from 0 to T.
(b) Find the mean-square response for $t \geq T$ and compare the result with that obtained in Problem 3.8.
(c) From the result of (b), would you say the filter's "memory" is finite or infinite?

3.21 The block diagram on the next page describes the error propagation in one channel of an inertial navigation system with external-velocity-reference damping (14). The inputs shown as $f_1(t)$ and $f_2(t)$ are random driving functions

due to the accelerometer and external-velocity-reference instrument errors. These will be assumed to be independent white-noise processes with spectral amplitudes A_1 and A_2, respectively. The outputs are labeled x_1, x_2, and x_3, and these physically represent the inertial system's platform tilt, velocity error, and position error. Find the steady-state mean-square value of each of the outputs.
(*Hint:* Since the system is linear, use superposition and note that the driving functions are independent.)



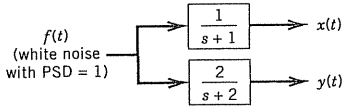
R = earth radius ≈ 2.09×10^7 ft
g = gravitational constant ≈ 32.2 ft/sec^2
K = feedback constant (adjustable to yield the desired damping ratio)

Problem 3.21

3.22 Random processes $x(t)$ and $y(t)$ are generated by passing unity Gaussian white noise through parallel filters as shown below.

(a) Find the autocorrelation functions for each of $x(t)$ and $y(t)$. You may assume that a stationary condition exists.
(b) Find the crosscorrelation function $R_{xy}(\tau)$.

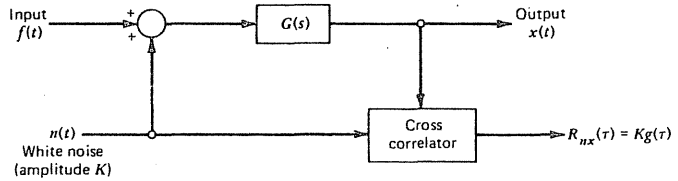
(*Hint:* Write x and y as convolution integrals. Then form the appropriate product and take the expectation. Finally, let $t \to \infty$ to get the stationary condition.)



Problem 3.22

3.23 The block diagram in the accompanying figure shows a means of determining the weighting function of a linear system (and thus, indirectly, the system's transfer function). The basic idea is to superimpose a small amount of white noise on the regular input and then crosscorrelate the resulting output with the intentionally added white noise. If the amplitude of the additive noise is relatively small, it is scarcely noticeable, if at all, and this provides a means of continuously monitoring the transfer characteristics of the system without disturbing its normal operation. Show that the output of the crosscorrelator $R_{nx}(\tau)$ is, in fact, proportional to the system weighting function $g(\tau)$. [If you need help on this one, see Truxal (15), p. 437. This idea has "been around" for a long

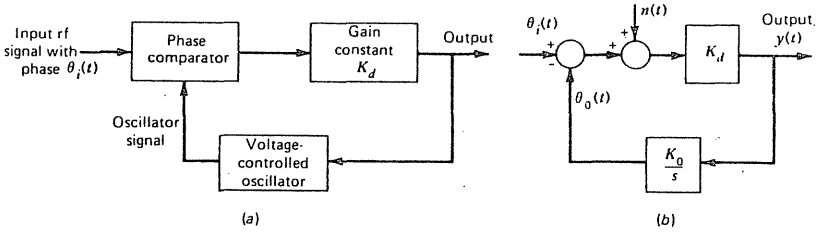
time but has seen only limited application. The reason, of course, lies in the computational effort in determining the crosscorrelation $R_{nx}(\tau)$ or its counterpart in the frequency domain, the cross-spectral density function. Either way, considerable computational effort is involved because the amount of data to be processed needs to be fairly large owing to the presence of a large signal component as well as the random noise. (See Problem 2.29 for more on determining the cross-spectral density function.)]



Problem 3.23

3.24 Part (*a*) of the accompanying figure shows a functional block diagram of an elementary phase-lock loop. Such circuits are used in a wide variety of communication applications (12, 16). Briefly, the phase comparator provides a signal proportional to the difference between the phase of the incoming rf signal and that of the local oscillator. The phase difference is modified by gain factor K_d and is then fed back as a voltage to the local voltage-controlled oscillator (VCO). This voltage causes the frequency of the VCO to shift up or down as necessary to lock onto the phase of the incoming rf signal.

Part (*b*) of the figure shows the linearized block diagram for a phase-lock loop with the addition of a phase noise input labeled $n(t)$. If the incoming rf signal is frequency-modulated (FM), it is the derivative of $\theta_i(t)$ that is proportional to the baseband signal, and thus $\dot{\theta}_i(t)$ is the signal to be recovered (detected) in this case.



Problem 3.24

(a) Show that the transfer function relating $\theta_i(t)$ to the output $y(t)$ is the equivalent of a differentiator in cascade with a first-order, low-pass filter.
(b) Consider the phase noise $n(t)$ to be flat with a spectral amplitude N_0. Also assume that the gain parameters of the phase-lock loop are chosen such that the baseband signal is passed with no distortion, that is, the loop response is flat over the signal frequency range, say, from 0 to W Hz. Find the spectral function of the output noise in the 0 to W Hz

range, and find its average power (i.e., mean-square value) in this frequency range. (It may help here to think of the noise as being limited to the 0 to W Hz range by a sharp-cutoff postdetector filter.)

3.25 For two stationary random processes $x(t)$ and $y(t)$, the coherence function is defined to be (see Chapter 2):

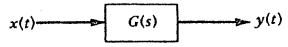
$$\gamma_{xy}^2(\omega) = \frac{|S_{xy}(j\omega)|^2}{S_x(j\omega)S_y(j\omega)} \tag{P3.25.1}$$

Consider a special case where $x(t)$ is the input to a linear system, and $y(t)$ is the output as shown in the accompanying figure.

(a) First show that

$$S_{xy}(j\omega) = G(j\omega)S_x(j\omega) \tag{P3.25.2}$$

(b) Then show that the coherence function is unity for all ω. (This shows that x and y do not have to be equal in order to have unity coherence. They only need to be intimately related, as, of course, they are in this situation.)



Problem 3.25

3.26 The recursion equation for generating a Markov process from problem 2.33 is repeated here for convenience:

$$x_{k+1} = e^{-\beta\Delta t}x_k + w_k, \quad k = 0, 1, 2, \ldots$$

w_k = white sequence $\sim N[0, \sigma^2(1 - e^{-2\beta\Delta t})]$

σ^2 = variance of the Markov process

β = reciprocal time constant of the Markov process

Δt = time interval between samples

(a) This will be recognized as a simple first-order ARMA model for the x_k process. Using the same parameter values as in Problem 2.33 (i.e., $\sigma^2 = 1$, $\beta = 1$, $\Delta t = .05$ sec), verify analytically that this model generates a stable process. This is easily done by locating the system characteristic pole in the z plane. What is the system time constant? Express this in terms of both seconds and number of discrete steps.

(b) In order to demonstrate the stability of the process of part (a), generate four sample realizations of the process for $k = 0, 1, 2, \ldots, 500$. The transient phenomenon that takes place and the evolution into a stationary condition will be more obvious if you initialize each realization to be zero at $k = 0$. View the plots of the four sample realizations. (The overall trend toward stationary behavior is more obvious when the four plots are superimposed on the same graph.)

(c) If we let $e^{-\beta\Delta t} = 1$ in the above recursion equation, the ARMA model becomes

$$x_{k+1} - x_k = w_k$$

This model was discussed in Example 3.10, and it was noted to be unstable. To demonstrate this, generate four sample realizations of this process for $k = 0, 1, 2, \ldots, 500$, and let the initial value of x_k be zero just as in part (b). Also, in order to make the initial rate of growth of x_k similar (statistically) to that of part (b), let the variance of w_k be .095163. View the plots of the four realizations and note the instability.

3.27 When it is difficult to integrate the power spectral density function in closed form, one should not overlook numerical integration. This is often the quickest way to get an answer to a specific numerical problem. In Problem 3.6, let $\sigma^2 = \beta = T = 1$, and repeat the problem using MATLAB's quad or quad8 numerical integration programs. Compare your result with the exact value of e^{-1}.

(*Note:* Beware of trying to integrate numerically through the origin where there is the indeterminate form 0/0. This can be avoided by staying an incremental distance away from the origin in the integration. Then approximate the omitted part as a narrow rectangular strip.)

3.28 One of the pseudorandom noise (PN) signals transmitted by the GPS satellites has a power spectral density that is approximated by the function

$$S(\omega) = \sigma^2 T\left[\frac{\sin\left(\frac{\omega T}{2}\right)}{\left(\frac{\omega T}{2}\right)}\right]^2 \tag{P3.28}$$

where σ^2 is the mean-square value of the signal, and T is the chip width of the spread spectrum signal. (See Chapter 11 for a brief discussion of the GPS satellite navigation system.)

(a) Find the fraction of the total power that is contained in the primary lobe of the spectrum (i.e., for $-2\pi/T < \omega < 2\pi/T$).

(b) Find the fraction of the total power that is contained in both the primary and first side lobes of the signal (i.e., for $-4\pi/T < \omega < 4\pi/T$).

(*Hint:* You will find MATLAB's numerical integration programs quad or quad8 useful for this problem. Also, the same precaution about integrating an indeterminate form that was mentioned in Problem 3.27 is applicable here.)

3.29 In the nonstationary mean-square-response problem, it is worth noting that the double integral in Eq. (3.8.4) reduces to a single integral when the input is white noise. That is, if $R_f(\tau) = A\delta(\tau)$, then

$$E[x^2(t)] = A\int_0^t g^2(v)\,dv \tag{P3.29}$$

where $g(v)$ is the inverse Laplace transform of the transfer function $G(s)$, and A is the power spectral density (PSD) of the white-noise input.

Evaluation of integrals analytically can be laborious (if not impossible), so one should not overlook the possibility of using numerical integration when $g^2(v)$

is either rather complicated in form, or when it is only available numerically. To demonstrate the effectiveness of the numerical approach, consider the following system driven by white noise whose PSD = 10 units:

$$G(s) = \frac{1}{s^2 + \omega_0^2}; \qquad \omega_0 = 20\pi$$

Let us say that we want to find $E[x^2(t)]$ for $0 \leq t \leq .1$ with a sample spacing of .001 sec (i.e., 101 samples including end points). Using MATLAB's numerical integration function quad8 or quad (or other suitable software), find the desired mean-square response numerically. Then plot the result along with the exact theoretical response for comparison [see part (b) of Problem 3.12].

REFERENCES CITED IN CHAPTER 3

1. J. J. D'Azzo and C. H. Houpis, *Linear Control System Analysis and Design,* 4th ed., New York: McGraw-Hill, 1995.
2. R. C. Dorf and R. H. Bishop, *Modern Control Systems,* 7th ed., Reading, MA: Addison-Wesley, 1995.
3. R. J. Smith and R. Dorf, *Circuits Devices and Systems,* 5th ed., New York: Wiley, 1992.
4. H. M. James, N. B. Nichols, and R. S. Phillips, *Theory of Servomechanisms,* Radiation Laboratory Series (Vol. 25), New York: McGraw-Hill, 1947.
5. G. C. Newton, L. A. Gould, and J. F. Kaiser, *Analytical Design of Linear Feedback Controls,* New York: Wiley, 1957.
6. G. R. Cooper and C. D. McGillem, *Probabilistic Methods of Signal and System Analysis,* 2nd ed., New York: Holt, Rinehart, and Winston, 1986.
7. K. S. Shanmugan and A. M. Breipohl, *Random Signals: Detection, Estimation, and Data Analysis,* New York: Wiley, 1988.
8. M. B. Priestley, *Spectral Analysis and Time Series,* New York: Academic Press, 1981.
9. J. M. Mendel, *Optimal Seismic Deconvolution,* New York: Academic Press, 1983.
10. J. S. Bendat and A. G. Piersol, *Random Data: Analysis and Measurement Procedures,* New York: Wiley-Interscience, 1971.
11. A. B. Carlson, *Communication Systems,* 2nd ed., New York: McGraw-Hill, 1975.
12. K. S. Shanmugan, *Digital and Analog Communication Systems,* New York: Wiley, 1979.
13. D. Childers and A. Durling, *Digital Filtering and Signal Processing,* St. Paul, MN: West Publishing, 1975.
14. G. R. Pitman (ed.), *Inertial Guidance,* New York: Wiley, 1962.
15. J. G. Truxal, *Automatic Feedback Control System Synthesis,* New York: McGraw-Hill, 1955.
16. F. M. Gardner, *Phaselock Techniques,* 2nd ed., New York: Wiley, 1979.

Additional References for General Reading

17. P. Z. Peebles, Jr., *Probability, Random Variables, and Random Signal Principles,* 3rd ed., New York: McGraw-Hill, 1993.
18. H. J. Larson and B. O. Shubert, *Probabilistic Models in Engineering Sciences* (Vols. 1 and 2), New York: Wiley, 1979.

4

Wiener Filtering

In this and subsequent chapters we will consider a particular branch of filter theory that is sometimes referred to as least-squares filtering. Actually, this is an oversimplification because it is the *average* squared error that is minimized and not just the squared error. "Linear minimum mean-square error filtering" is a more descriptive name for this type of filtering. This is a bit wordy, though, so the name is often shortened to MMSE filtering (for minimum mean-square error). Simply stated, the linear MMSE filter problem is this: Given the spectral characteristics of an additive combination of signal and noise, what linear operation on this input combination will yield the best separation of the signal from the noise? "Best" in this case means minimum mean-square error. This branch of filtering began with N. Wiener's work in the 1940s (1). R. E. Kalman then made an important contribution in the early 1960s by providing an alternative approach to the same problem using state-space methods (2, 3). Kalman's contribution has been especially significant in applied work, because his solution is readily implemented in time-variable, multiple-input/multiple-output applications.

We will consider the Wiener and Kalman theories in their historical order. It should be mentioned that neither is prerequisite material for the other; therefore, they may be studied in either order, or one to the exclusion of the other, for that matter.

4.1 THE WIENER FILTER PROBLEM

The purpose of any filter is to separate one thing from another. In the electric filter case, this usually refers to passing signals in a specified frequency range and rejecting those outside that range; and, historically, filter theory began with the problem of designing a circuit to yield the desired frequency response. This is still an important problem. In many applications in communication and control, one knows intuitively what the ideal frequency response should be. For

example, if we want to receive the signal from a particular AM radio station (and do it faithfully), we know that the appropriate filter is one that passes all frequencies within a few kilohertz on either side of the assigned station frequency and rejects all others. Certainly, no elaborate theory is needed to determine the desired frequency response in this case. The problem is simply one of circuit design. We will see, though, that this is not always the case. During World War II, Norbert Wiener considered a different sort of filter problem (1). Suppose the signal, as well as the noise, is noiselike in character, and suppose further that there is a significant overlap in the spectra of both the signal and noise. For example, say the signal is a Gauss–Markov process and the corrupting noise is white noise. Their spectral densities are shown in Fig. 4.1. Now, in this case, it should be apparent that no filter is going to yield perfect separation, and the filter that gives the best compromise of passing the signal and, at the same time, suppresses most of the noise is not at all obvious. Neither is it obvious how one should define "best compromise" in order to make the problem mathematically tractable. This is the problem Wiener examined in the 1940s. We note that he was not concerned with filter design in the sense of choosing appropriate resistors, capacitors, and so forth. Instead, his problem was more fundamental; namely, what should the filter's frequency response be in order to give the best possible separation of signal from noise?

The theory that is now loosely referred to as Wiener filter theory is characterized by:

1. The assumption that both signal and noise are random processes with known spectral characteristics or, equivalently, known auto- and cross-correlation functions.
2. The criterion for best performance is minimum mean-square error. This is partially to make the problem mathematically tractable, but it is also a good physical criterion in many applications.
3. A solution based on scalar methods that leads to the optimal filter weighting function (or transfer function in the stationary case).

We now proceed to the filter optimization problem.

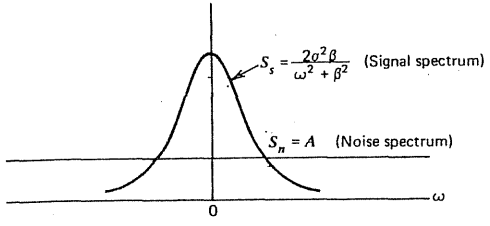


Figure 4.1 Spectral densities of signal and noise.

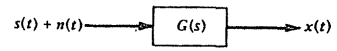


Figure 4.2 Filter optimization problem.

4.2 OPTIMIZATION WITH RESPECT TO A PARAMETER

One of the early methods of stochastic filter optimization was presented by R. S. Phillips in Volume 25 of the now famous Radiation Laboratory Series (4). His approach was less general than Wiener's but it is still useful in many applications. Basically, the method is to choose the form of the filter transfer function intuitively, leaving one or more parameters free to vary. One then minimizes the mean-square error with respect to these parameters.

Before looking at an example of Phillip's procedure, we need to derive an expression for the filter's mean-square error. In terms of its Laplace transform, the output of the filter shown in Fig. 4.2. is*

$$X(s) = G(s)[S(s) + N(s)] \tag{4.2.1}$$

We define the filter error as the difference between the actual output and what we would like it to be ideally—the signal. Therefore let†

$$e(t) = s(t) - x(t) \tag{4.2.2}$$

and

$$E(s) = S(s) - X(s) \tag{4.2.3}$$

Substituting Eq. (4.2.1) into Eq. (4.2.3) yields

$$E(s) = S(s) - G(s)[S(s) + N(s)] = [1 - G(s)]S(s) - [G(s)]N(s) \tag{4.2.4}$$

It can be seen that the error can be thought of as a superposition of two components, one due to the signal modified by the transfer function $[1 - G(s)]$ and another due to the noise modified by $-G(s)$. If the signal and noise have zero

* Where there might be confusion between the signal variable $s(t)$ and the complex variable s, we will show the time dependence explicitly, that is, $s(t)$ is the signal variable. Also, uppercase $S(s)$ without a subscript will denote the Laplace transform of $s(t)$; S with a subscript, for example, $S_s(s)$, denotes spectral density of the subscript variable. This overlap in notation should not be confusing when taken within the context of the subject under consideration.

† Some authors prefer to define the error with the opposite sign. In the subsequent optimization we will always be concerned with minimizing the mean-*square*-error. Thus, the sign of the error is of no consequence; the resulting optimal filter and its mean-square error are the same either way.

crosscorrelation, the mean-square error is obtained as simply the sum of two terms, that is,

$$E(e^2) = \frac{1}{2\pi j}\int_{-j\infty}^{j\infty} [1 - G(s)][1 - G(-s)]S_s(s)\,ds + \frac{1}{2\pi j}\int_{-j\infty}^{j\infty} G(s)G(-s)S_n(s)\,ds \tag{4.2.5}$$

We now have an explicit expression for the mean-square error in terms of the spectral functions of $s(t)$ and $n(t)$ (presumably known) and the filter transfer function. If $G(s)$ contains a parameter free to vary, we can now use ordinary differential calculus to minimize $E(e^2)$ with respect to the parameter. We now proceed with an example.

EXAMPLE 4.1

Consider the Gauss–Markov signal and white-noise situation shown in Fig. 4.1. It is apparent that some sort of low-pass filter is needed to separate signal from noise. Let us try a simple first-order filter of the form

$$G(s) = \frac{1}{1 + Ts}$$

We have now specified the functional form for $G(s)$, and hence we are ready to use Eq. (4.2.5). The needed quantities are

$$G(s) = \frac{1}{1 + Ts} = \frac{1/T}{s + (1/T)}$$

$$1 - G(s) = 1 - \frac{1}{1 + Ts} = \frac{s}{s + (1/T)}$$

$$S_s(s) = \frac{2\sigma^2\beta}{-s^2 + \beta^2} = \frac{\sqrt{2\sigma^2\beta}}{s + \beta} \cdot \frac{\sqrt{2\sigma^2\beta}}{-s + \beta}$$

$$S_n(s) = A = \sqrt{A} \cdot \sqrt{A}$$

Substituting the above quantities into Eq. (4.2.5) and evaluating $E(e^2)$ using the integral tables of Section 3.4 yield

$$E(e^2) = \frac{\sigma^2\beta T}{1 + \beta T} + \frac{A}{2T} \tag{4.2.6}$$

This can now be minimized with respect to T using differential calculus. The result is that $E(e^2)$ is a minimum for

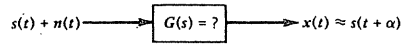


Figure 4.3 Wiener filter problem.

$$T = \frac{\sqrt{A}}{\sigma\sqrt{2\beta} - \beta\sqrt{A}} \tag{4.2.7}$$

It is interesting to note that this will yield a positive value of T only for certain values of the parameters. A negative solution for T simply means that no *relative* minimum exists within the interval from zero to infinity.

It should be remembered that the minimum obtained is not the absolute minimum possible (unless by coincidence), because the form of the filter transfer function was chosen intuitively. Other functional forms might have done better. ▩

4.3 THE STATIONARY OPTIMIZATION PROBLEM—WEIGHTING FUNCTION APPROACH*

We now consider the filter optimization problem that Wiener first solved in the 1940s (1). Referring to Fig. 4.3, we assume the following:

1. The filter input is an additive combination of signal and noise, both of which are covariance stationary with known auto- and crosscorrelation functions (or corresponding spectral functions).
2. The filter is linear and not time-varying. No further assumption is made as to its form.
3. The output is covariance stationary. (A long time has elapsed since any switching operation.)
4. The performance criterion is minimum mean-square error, where the error is defined as $e(t) = s(t + \alpha) - x(t)$.

In addition to the generalization relative to the form of the filter transfer function, we are also generalizing by saying the ideal filter output is to be $s(t + \alpha)$ rather than just $s(t)$. The following terminology has evolved relative to the choice of the α parameter:

1. ***α positive:*** This is called the *prediction* problem. (The filter is trying to predict the signal value α units ahead of the present time t.)
2. ***α = 0:*** This is called the *filter* problem. (The usual problem we have considered before.)

* Two-sided Laplace transform theory is used extensively in this section. See Appendix A for a brief review of two-sided transform theory.

3. **α *negative:*** This is called the *smoothing* problem. (The filter is trying to estimate the signal value α units in the past.)

This is an important generalization and there are numerous physical applications corresponding to all three cases. The α parameter is chosen to fit the particular application at hand, and it is fixed in the optimization process.

We begin by defining the filter error as

$$e(t) = s(t + \alpha) - x(t) \tag{4.3.1}$$

The squared error is then

$$e^2(t) = s^2(t + \alpha) - 2s(t + \alpha)x(t) + x^2(t) \tag{4.3.2}$$

We next write $x(t)$ as a convolution integral:

$$\int_{-\infty}^{\infty} g(u)[s(t - u) + n(t - u)]\, du \tag{4.3.3}$$

This can be substituted into Eq. (4.3.2) and both sides averaged to yield*

$$E(e^2) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(u)g(v)R_{s+n}(u - v)\, du\, dv - 2\int_{-\infty}^{\infty} g(u)R_{s+n,s}(\alpha + u)\, du + R_s(0) \tag{4.3.4}$$

where

R_s = autocorrelation function of $s(t)$

R_{s+n} = autocorrelation function of $s(t) + n(t)$

$R_{s+n,s}$ = crosscorrelation between $s(t) + n(t)$ and $s(t)$

Note that if signal and noise have zero crosscorrelation,

$$\left.\begin{aligned} R_{s+n} &= R_s + R_n \\ R_{s+n,s} &= R_s \end{aligned}\right\} \quad \text{(for zero correlation)} \tag{4.3.5}$$

We wish to find the function $g(u)$ in Eq. (4.3.4) that minimizes $E(e^2)$. This will be recognized as a problem in calculus of variations (5). Following the usual procedure, we replace $g(u)$ with a perturbed weighting function $g(u) + \varepsilon\eta(u)$

* We have chosen here to write the mean-square error in terms of the filter weighting function and input autocorrelation functions. In the stationary problem, one can also write $E(e^2)$ in terms of the filter transfer function and input spectral functions, and then proceed with the optimization on that basis (6, 7). We have chosen the time-domain approach because it is easily generalized to the nonstationary problem that is considered in Section 4.4. The frequency-domain approach is not readily generalized.

where

$g(u)$ = optimum weighting function [*Note*: From this point on in the solution, $g(u)$ will denote the *optimal* weighting function.]

$\eta(u)$ = an arbitrary perturbing function

ε = small perturbation factor such that the perturbed function approaches the optimum one as ε goes to zero

The optimum and perturbed weighting functions are sketched in Fig. 4.4. Replacing $g(u)$ with $g(u) + \varepsilon\eta(u)$ in Eq. (4.3.4) then leads to

$$E(e^2) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} [g(u) + \varepsilon\eta(u)][g(v) + \varepsilon\eta(v)]R_{s+n}(u - v)\, du\, dv - 2\int_{-\infty}^{\infty} [g(u) + \varepsilon\eta(u)]R_{s+n,s}(\alpha + u)\, du + R_s(0) \tag{4.3.6}$$

Note that $E(e^2)$ is now a function of ε, and it is to be a minimum when $\varepsilon = 0$. Now, using differential calculus methods, we differentiate $E(e^2)$ with respect to ε and set the result equal to zero for $\varepsilon = 0$. After interchanging dummy variables of integration freely, the result is

$$\int_{-\infty}^{\infty} \eta(\tau)\left[-R_{s+n,s}(\alpha + \tau) + \int_{-\infty}^{\infty} g(u)R_{s+n}(u - \tau)\, du\right] d\tau = 0 \tag{4.3.7}$$

A subtlety in the solution arises at this point; therefore, it is convenient to look at the causal and noncausal cases separately.

Noncausal Solution

If we put no constraint on the filter weighting function, we will very likely obtain a $g(u)$ that is nontrivial for negative as well as positive u. This weighting function is noncausal because it requires the filter to "look ahead" of real time and use data that are not yet available. This is, of course, not possible if the filter is operating on-line. However, in off-line applications, such as postflight analysis of recorded data, the noncausal solution is possible and very much of interest. Thus, it should not be ignored.

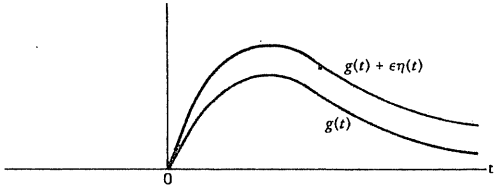


Figure 4.4 Optimal and perturbed weighting functions.

If there are no restrictions on $g(u)$, then, similarly, there are no constraints on the perturbation function $\eta(\tau)$. It is arbitrary for all values of its argument. Thus, if the integral with respect to τ in Eq. (4.3.7) is to be zero, the bracketed term must be zero for *all* τ. This leads to

$$\int_{-\infty}^{\infty} g(u)R_{s+n}(u-\tau)\,du = R_{s+n,s}(\alpha+\tau), \qquad -\infty < \tau < \infty \tag{4.3.8}$$

This is an integral equation of the first kind, and in this case it can be solved readily using Fourier transform methods. Since R_{s+n} is symmetric, the term on the left side of Eq. (4.3.8) has the exact form of a convolution integral. Therefore, transforming both sides yields

$$G(s)S_{s+n}(s) = S_{s+n,s}(s)e^{\alpha s} \tag{4.3.9}$$

or

$$G(s) = \frac{S_{s+n,s}(s)e^{\alpha s}}{S_{s+n}(s)} \tag{4.3.10}$$

Remember that the transforms indicated in Eq. (4.3.10) are *two-sided* transforms rather than the usual single-sided transforms. Of course, if we wish to find the weighting function $g(u)$, we simply take the inverse transform of the expression given by Eq. (4.3.10).

The filter mean-square error is given by Eq. (4.3.4). If $g(u)$ is the optimal weighting function satisfying Eq. (4.3.8), the mean-square error equation may be simplified as follows. First write the second term of Eq. (4.3.4) as the sum of two equal terms and combine one of these with the double integral term. After we rearrange terms, this leads to

$$E(e^2) = R_s(0) - \int_{-\infty}^{\infty} g(u)R_{s+n,s}(\alpha+u)\,du + \int_{-\infty}^{\infty} g(u)\left[-R_{s+n,s}(\alpha+u) + \int_{-\infty}^{\infty} g(v)R_{s+n}(v-u)\,dv\right]du \tag{4.3.11}$$

The bracketed quantity in Eq. (4.3.11) is zero for optimal $g(v)$ for all u. Therefore, the mean-square error is

$$E(e^2) = R_s(0) - \int_{-\infty}^{\infty} g(u)R_{s+n,s}(\alpha+u)\,du \tag{4.3.12}$$

EXAMPLE 4.2

Consider the same Markov signal and white-noise combination used in Example 4.1. We wish to find the optimal noncausal filter (i.e., $\alpha = 0$). In order to simplify the arithmetic, let $\sigma^2 = \beta = A = 1$. Since the signal and noise have zero crosscorrelation,

$$S_{s+n} = S_s + S_n = \frac{2}{-s^2+1} + 1 = \frac{-s^2+3}{-s^2+1} \tag{4.3.13}$$

$$S_{s+n,s} = S_s = \frac{2}{-s^2+1} \tag{4.3.14}$$

$$e^{\alpha s} = 1 \tag{4.3.15}$$

From Eq. (4.3.10) we have

$$G(s) = \frac{\dfrac{2}{-s^2+1}}{\dfrac{-s^2+3}{-s^2+1}} = \frac{2}{-s^2+3}$$

Expanding this with a partial fraction expansion yields

$$G(s) = \frac{1/\sqrt{3}}{s+\sqrt{3}} + \frac{1/\sqrt{3}}{-s+\sqrt{3}} \tag{4.3.16}$$

The positive- and negative-time parts of $g(u)$ are given by the first and second terms of Eq. (4.3.16). Thus, $g(u)$ is

$$g(u) = \begin{cases} \dfrac{1}{\sqrt{3}}e^{-\sqrt{3}u}, & u \geq 0 \\ \dfrac{1}{\sqrt{3}}e^{\sqrt{3}u}, & u < 0 \end{cases} \tag{4.3.17}$$

This is the optimal noncausal weighting function, and it is sketched in Fig. 4.5 along with the intuitive weighting function of Example 4.1 (evaluated for $\sigma^2 = \beta = A = 1$ and $T = 1 + \sqrt{2}$). Note that since the noncausal filter weights both past and future input data, it can afford to have a smaller time constant than the intuitive filter, which is allowed to weight only past input data.

It is also of interest to compare the mean-square errors for the noncausal optimal filter and the parameter-optimized filter of Example 4.1. These may be computed from Eqs. (4.2.6) and (4.3.12) with the result

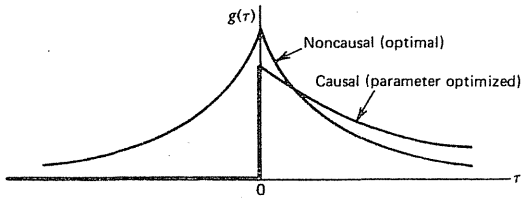


Figure 4.5 Filter weighting functions for optimal noncausal and parameter-optimized causal filters.

$$E(e^2) \quad \text{(parameter-optimized)} \approx 0.914$$

$$E(e^2) \quad \text{(noncausal optimal)} \approx 0.577$$

Notice that the noncausal filter has significantly less error than the causal one. Certainly, in off-line applications it would be worthwhile implementing the noncausal filter in preference to the intuitive causal one. ■

Causal Solution

The calculus of variations procedure led to Eq. (4.3.7), which is repeated here for convenient reference.

$$\int_{-\infty}^{\infty} \eta(\tau) \left[-R_{s+n,s}(\alpha + \tau) + \int_{-\infty}^{\infty} g(u) R_{s+n}(u - \tau)\, du \right] d\tau = 0 \qquad (4.3.7)$$

Recall that $\eta(\tau)$ is an arbitrary perturbing function. If we wish to constrain the filter weighting function to be causal, we must place a similar constraint on $\eta(\tau)$ in the variation. Otherwise, we get the unconstrained (noncausal) solution. Thus, for the causal case we require $\eta(\tau)$ to be zero for negative τ, and allow it to be arbitrary for positive τ. The bracketed quantity in Eq. (4.3.7) then needs to be zero only for positive τ. The zero criterion is satisfied for negative τ by virtue of our constraint on $\eta(\tau)$, that is, $\eta(\tau) = 0$ for $\tau < 0$. Therefore, the resulting integral equation is

$$\int_{-\infty}^{\infty} g(u) R_{s+n}(u - \tau)\, du - R_{s+n,s}(\alpha + \tau) = 0, \qquad \tau \geq 0 \qquad (4.3.18)$$

Equation (4.3.18) is known as the *Wiener-Hopf equation*, and the fact that it is valid only for $\tau \geq 0$ complicates the solution considerably.

One solution of Eq. (4.3.18) that is based on spectral factorization proceeds as follows. First, replace the right side with an unknown negative-time function $a(\tau)$ [i.e., $a(\tau)$ is unknown for negative time, but is known to be zero for positive time.] Equation (4.3.18) can be written as

$$\int_{-\infty}^{\infty} g(u) R_{s+n}(u - \tau)\, du - R_{s+n,s}(\alpha + \tau) = a(\tau), \qquad -\infty < \tau < \infty \qquad (4.3.19)$$

Transforming both sides of Eq. (4.3.19) then yields

$$G(s) S_{s+n}(s) - S_{s+n,s}(s) e^{\alpha s} = A(s) \qquad (4.3.20)$$

Next, use spectral factorization on S_{s+n} and group terms as follows:

$$[G(s) S^{+}_{s+n}(s)] S^{-}_{s+n}(s) - S_{s+n,s}(s) e^{\alpha s} = A(s)$$

or

$$G(s) S^{+}_{s+n}(s) = \frac{A(s)}{S^{-}_{s+n}(s)} + \frac{S_{s+n,s}(s) e^{\alpha s}}{S^{-}_{s+n}(s)} \qquad (4.3.21)$$

In Eq. (4.3.21), the "super +" indicates the factored part of the spectral function that has all its poles and zeros in the left half-plane. Similarly, the poles and zeros of S^{-}_{s+n} are mirror images of those of S^{+}_{s+n}. We note here that $g(u)$ is a stable positive-time function; therefore $G(s)$ will have its poles in the left half-plane. Thus, $G(s)S^{+}_{s+n}(s)$ will have all its poles in the left half-plane, and it will be the transform of a positive-time function. Similarly, $A(s)$ is the transform of a negative-time function, so its poles will be in the right half-plane. Also, both the zeros and poles of $S^{-}_{s+n}(s)$ are in the right half-plane. Hence, the three terms of Eq. (4.3.21) translate into words as

$$\begin{bmatrix}\text{Positive-time}\\ \text{function}\end{bmatrix} = \begin{bmatrix}\text{Negative-time}\\ \text{function}\end{bmatrix} + \begin{bmatrix}\text{Both positive- and}\\ \text{negative-time}\\ \text{function}\end{bmatrix}$$

Equating positive-time parts on both sides of Eq. (4.3.21) then leads to

$$G(s) S^{+}_{s+n}(s) = \text{Positive-time part of } \frac{S_{s+n,s}(s) e^{\alpha s}}{S^{-}_{s+n}(s)}$$

or

$$G(s) = \frac{1}{S^{+}_{s+n}(s)} \left[\text{Positive-time part of } \frac{S_{s+n,s}(s) e^{\alpha s}}{S^{-}_{s+n}(s)} \right] \qquad (4.3.22)$$

The bracketed term of Eq. (4.3.22) can be interpreted as follows. First, find the inverse transform of $S_{s+n,s}(s)/S^{-}_{s+n}(s)$. This will normally be nontrivial for both positive and negative time. Next, translate the time function an amount α. (This accounts for $e^{\alpha s}$.) Finally, take the ordinary *single-sided* Laplace transform of the shifted time function, and this will be the bracketed quantity in Eq. (4.3.22). Two examples should be helpful at this point.

EXAMPLE 4.3

Consider the same Markov signal and white-noise combination used in Examples 4.1 and 4.2. Again we let $\sigma^2 = \beta = A = 1$ to simplify the arithmetic and, in this example, we are looking for the optimal causal solution. Since the signal and noise are assumed to have zero crosscorrelation, the needed spectral functions are

$$S_{s+n} = S_s + S_n = \frac{2}{-s^2 + 1} + 1 = \frac{-s^2 + 3}{-s^2 + 1} \qquad (4.3.23)$$

$$S_{s+n,s} = S_s = \frac{2}{-s^2 + 1} \qquad (4.3.24)$$

Also, since the prediction time α is assumed to be zero,

$$e^{\alpha s} = 1 \tag{4.3.25}$$

First, we factor S_{s+n}:

$$S_{s+n} = S^+_{s+n}S^-_{s+n} = \left[\frac{s+\sqrt{3}}{s+1}\right]\left[\frac{-s+\sqrt{3}}{-s+1}\right] \tag{4.3.26}$$

Next, we form the $S_{s+n,s}/S^-_{s+n}$ function:

$$\frac{S_{s+n,s}}{S^-_{s+n}} = \frac{\dfrac{2}{-s^2+1}}{\dfrac{-s+\sqrt{3}}{-s+1}} = \frac{2}{(-s+\sqrt{3})(s+1)} \tag{4.3.27}$$

This, in turn, can be expanded in terms of a partial fraction expansion:

$$\frac{S_{s+n,s}}{S^-_{s+n}} = \frac{\sqrt{3}-1}{s+1} + \frac{\sqrt{3}-1}{-s+\sqrt{3}} \tag{4.3.28}$$

Clearly, the first term of Eq. (4.3.28) is the positive-time part. Therefore, $G(s)$ as given by Eq. (4.3.22) is

$$G(s) = \frac{1}{\dfrac{s+\sqrt{3}}{s+1}} \cdot \frac{\sqrt{3}-1}{s+1} = \frac{\sqrt{3}-1}{s+\sqrt{3}} \tag{4.3.29}$$

Or, in terms of the filter weighting function,

$$g(t) = \begin{cases} (\sqrt{3}-1)e^{-\sqrt{3}t}, & t \geq 0 \\ 0, & t < 0 \end{cases} \tag{4.3.30}$$

As before, the mean-square error of the filter can be computed using Eq. (4.3.12). The result is

$$E(e^2) = .732 \tag{4.3.31}$$

We have now examined three different optimization approaches for the same signal-plus-noise situation. A comparison of the results is shown in Table 4.1, with the most restrictive filter being listed first and the least restrictive one listed last. As should be expected, the mean-square error decreases with each successive relaxation of the constraints on the choice of transfer function. The linear constraint is, of course, present in all three solutions. We will see in later chapters that this is not as serious as one might think at first glance. The explanation of this, though, will be deferred until Chapter 5.

Table 4.1 Comparison of Results of Examples 4.1, 4.2, and 4.3

Type of Solution	Filter Transfer Function	Weighting Function	Mean-Square Error
Single parameter optimization	$G(s) = \dfrac{1}{1+(1+\sqrt{2})s}$	$g(t) = \begin{cases} \dfrac{1}{1+\sqrt{2}}e^{-(\sqrt{2}-1)t}, & t \geq 0 \\ 0, & t < 0 \end{cases}$	.914
Causal Wiener filter	$G(s) = \dfrac{\sqrt{3}-1}{s+\sqrt{3}}$	$g(t) = \begin{cases} (\sqrt{3}-1)e^{-\sqrt{3}t}, & t \geq 0 \\ 0, & t < 0 \end{cases}$	.732
Noncausal Wiener filter	$G(s) = \dfrac{2}{(s+\sqrt{3})(-s+\sqrt{3})}$	$g(t) = \begin{cases} \dfrac{1}{\sqrt{3}}e^{-\sqrt{3}t}, & t \geq 0 \\ \dfrac{1}{\sqrt{3}}e^{\sqrt{3}t}, & t < 0 \end{cases}$	.577

EXAMPLE 4.4

There is a classical problem in random process theory known as the *pure prediction* problem. Here we assume the additive noise corrupting the signal is zero, and we pose the problem of looking ahead and finding the best estimate of the signal α units ahead of the present time t. To demonstrate the applicability of Wiener filter theory to this problem, let the signal be Markov with a known autocorrelation function:

$$R_s(\tau) = \sigma^2 e^{-\beta|\tau|} \quad \text{or} \quad S_s(s) = \frac{2\sigma^2\beta}{-s^2+\beta^2} \tag{4.3.32}$$

Also,

$$R_n(\tau) = 0$$

We first factor $S_{s+n}(s)$.

$$S_{s+n} = S_s = S^+_s \cdot S^-_s = \frac{\sqrt{2\sigma^2\beta}}{s+\beta} \cdot \frac{\sqrt{2\sigma^2\beta}}{-s+\beta} \tag{4.3.33}$$

Next, we form the $S_{s+n,s}/S^-_{s+n}$ function

$$\frac{S_{s+n,s}}{S^-_{s+n}} = \frac{S_s}{S^-_s} = S^+_s = \frac{\sqrt{2\sigma^2\beta}}{s+\beta} \tag{4.3.34}$$

In this problem α is not zero, and hence we must first multiply Eq. (4.3.34) by $e^{\alpha s}$ and then find the positive-time part of the result. This is readily accomplished by appropriate shifting in the time domain as shown in Fig. 4.6. Finally substituting the proper positive-time part into Eq. (4.3.22) yields

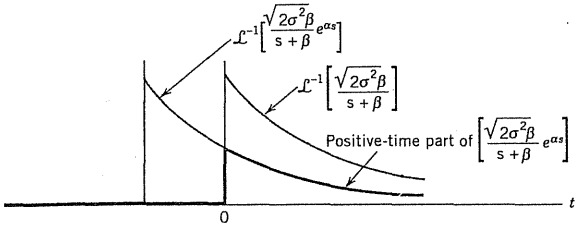


Figure 4.6 The shifted time functions for Example 4.4.

$$G(s) = \frac{1}{\frac{\sqrt{2\sigma^2\beta}}{s+\beta}} \cdot e^{-\alpha\beta}\frac{\sqrt{2\sigma^2\beta}}{s+\beta} = e^{-\alpha\beta} \tag{4.3.35}$$

Or the corresponding weighting function is

$$g(t) = e^{-\alpha\beta}\delta(t) \tag{4.3.36}$$

The Wiener solution says that the best one can hope to do (in the least-squares sense) is to multiply the present value of the input by an attenuation factor $e^{-\alpha\beta}$, and this will yield the best estimate of the process α units ahead of the present time. Observe that the predictive estimate is dependent only on the present value of the input signal and not on its past history. If the Markov signal under consideration is also Gaussian, then the predictive estimate given by Wiener theory is also identical to the conditional mean of the process at the predicted time, conditioned on all the prior input data. The use of the term Markov for a process with an exponential autocorrelation function should now be apparent. This will be expanded on later, in Chapter 5. ■

(See also p. 357)

4.4 THE NONSTATIONARY PROBLEM

In the preceding discussion it was assumed that the filter was "turned on" at $t = -\infty$, which made the entire past history of $s(t) + n(t)$ available for weighting. This led to the steady-state or stationary solution. In the transient or nonstationary problem, we consider the signal and noise to be covariance stationary processes with known spectral characteristics as before, but we consider the input to be applied at $t = 0$ rather than $-\infty$. This is indicated in Fig. 4.7 as a switching

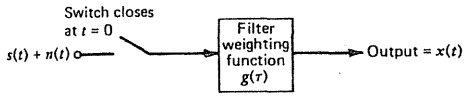


Figure 4.7 Block diagram for the nonstationary problem.

operation at the input. We assume zero initial conditions; hence the filter output can be written as

$$x(t) = \int_0^t g(\tau)[s(t-\tau) + n(t-\tau)]\,d\tau, \qquad t \geq 0 \tag{4.4.1}$$

As before, the ideal output would be $s(t + \alpha)$. Thus, the filter error is

$$e(t) = s(t+\alpha) - x(t) \tag{4.4.2}$$

Substituting Eq. (4.4.1) into Eq. (4.4.2), squaring, and then taking the expectation of both sides yield

$$E(e^2) = \int_0^t \int_0^t g(u)g(v)R_{s+n}(u-v)\,du\,dv - 2\int_0^t g(u)R_{s+n,s}(\alpha+u)\,du + R_s(0) \tag{4.4.3}$$

The problem is to choose $g(u)$ such as to minimize $E(e^2)$. The variational procedure to be followed is essentially the same as for the stationary case, so it will not be repeated. The only difference is in the limits on the integration, which, in turn, traces back to the expression for $x(t)$ given by Eq. (4.4.1). This is an important difference, though. We need not worry about placing a causality constraint on $g(\tau)$ or its perturbation, because the range of integration is limited to be from 0 to t. We can arbitrarily truncate $g(\tau)$ to zero outside this range.

As before, the variational procedure leads to an integral equation in $g(\tau)$. For the nonstationary problem, it is

$$\int_0^t g(u)R_{s+n}(\tau-u)\,du = R_{s+n,s}(\alpha+\tau), \qquad 0 \leq \tau \leq t \tag{4.4.4}$$

This equation is similar to the Wiener–Hopf equation except for the limits on the integral and the range of τ for which the equation is valid. These seemingly small differences complicate the solution even further, though. The spectral factorization method cannot be applied to Eq. (4.4.4) because the range of τ is finite rather than semifinite as in the Wiener–Hopf equation. Equation (4.4.4) is amenable to solution, though, by another method that involves transforming the integral equation to a differential equation and then solving the differential equation for $g(\tau)$ (8). This technique will work only for the case where the spectral function for $s(t) + n(t)$ is rational, that is, a ratio of polynomials in s^2. However, we note that this same restriction is necessary for spectral factorization, and therefore only limited types of spectral situations can be handled for either the stationary or nonstationary case.

The solution of Eq. (4.4.4) proceeds as follows. We first write R_{s+n} as a Fourier integral

$$\int_0^t g(u)\left[\frac{1}{2\pi j}\int_{-j\infty}^{j\infty} S_{s+n}(s)e^{s(\tau-u)}\,ds\right]du = R_{s+n,s}(\alpha+\tau) \tag{4.4.5}$$

Next, we write $S_{s+n}(s)$ as a ratio of polynomials in s^2

$$S_{s+n}(s) = \frac{N(s^2)}{D(s^2)} \tag{4.4.6}$$

Now we note that operating on $e^{s(\tau-u)}$ with the differential operator $D(d^2/d\tau^2)$ will generate $D(s^2)e^{s(\tau-u)}$.* Thus, the denominator of S_{s+n} in Eq. (4.4.5) can be canceled by operating on both sides of the equation with $D(d^2/d\tau^2)$. Similarly, algebraic multiplication within the integral by $N(s^2)$ is equivalent to operating in front of the first integral with $N(d^2/d\tau^2)$. Inserting these equivalent operations in Eq. (4.4.5) yields

$$N\left(\frac{d^2}{d\tau^2}\right)\int_0^t g(u)\left[\frac{1}{2\pi j}\int_{-j\infty}^{j\infty} e^{s(\tau-u)}\,ds\right]du = D\left(\frac{d^2}{d\tau^2}\right)R_{s+n,s}(\alpha+\tau) \tag{4.4.7}$$

We now note that the Fourier integral in brackets in Eq. (4.4.7) is just the Dirac delta function $\delta(\tau - u)$. Inserting this and using the shifting property of the impulse function lead to

$$N\left(\frac{d^2}{d\tau^2}\right)g(\tau) = D\left(\frac{d^2}{d\tau^2}\right)R_{s+n,s}(\alpha+\tau), \qquad 0 < \tau < t \tag{4.4.8}$$

We now have a differential equation in $g(\tau)$ rather than an integral equation. Furthermore, it is a linear differential equation with constant coefficients, and the solution of this type of equation is well known. Before proceeding, note that the interval on τ in Eq. (4.4.8) is the open interval $(0, t)$ rather than the closed interval $[0, t]$ associated with the integral equation, Eq. (4.4.4). This is intentional and arises because of the problem of continuity and differentiation at the end points. In other words, we may safely assume only that the differential equation is valid in the interior region of the interval.

The solution of Eq. (4.4.8) will, of course, contain arbitrary constants of integration. Furthermore, because of the end-point problem, impulse functions with undetermined amplitudes must be added at $\tau = 0+$ and $\tau = t-$. (The + and − indicate that the impulses are placed at the inside edges of the interval.) The rule for adding the impulses is as follows: If the order of $D(s^2)$ is:

1. The same as $N(s^2)$, add no impulses.
2. If it is two greater than $N(s^2)$, add simple impulses.

* Remember that D refers to the polynomial in the denominator of $S_{s+n}(s)$ and not the "D-operator" used in differential equation theory. That is, $D(d^2/d\tau^2)$ is $D(s^2)$ with s^2 replaced by $d^2/d\tau^2$. Similarly, $N(d^2/d\tau^2)$ is the numerator polynomial $N(s^2)$ with s^2 replaced with $d^2/d\tau^2$.

3. If it is four greater than $N(s^2)$, add simple impulses plus doublet impulses.
4. Etc.

The unknown coefficients in the general solution are evaluated by substituting the assumed solution into the original integral equation and demanding equality on both sides of the equation. This is much the same as using the initial conditions to evaluate the constants of integration in the usual initial-condition problem.

Remember that the procedure just described is highly specialized and applies only to the case where $S_{s+n}(s)$ can be written as a ratio of polynomials in s^2. The justification of the procedure in any particular case lies in the final substitution of the solution into the integral equation. We can ask no more of the solution than to satisfy the original integral equation.

EXAMPLE 4.5

We again look at the situation where the signal is Markov and the noise is white. The additive combination forms the input that is applied to the filter at $t = 0$. Let

$$R_s(\tau) = e^{-|\tau|} \quad \text{or} \quad S_s(s) = \frac{2}{-s^2+1} \tag{4.4.9}$$

$$R_n(\tau) = \delta(\tau) \quad \text{or} \quad S_n(s) = 1 \tag{4.4.10}$$

If we assume the signal and noise have zero crosscorrelation,

$$S_{s+n}(s) = \frac{2}{-s^2+1} + 1 = \frac{-s^2+3}{-s^2+1} = \frac{N(s^2)}{D(s^2)} \tag{4.4.11}$$

Also, if $\alpha = 0$,

$$R_{s+n,s}(\alpha+\tau) = R_s(\tau) = e^{-\tau}, \qquad \tau \geq 0 \tag{4.4.12}$$

Note that the absolute magnitude signs around τ may be dropped because τ is always positive. If we use the polynomials $N(s^2)$ and $D(s^2)$ as given by Eq. (4.4.11), the differential equation (4.4.8) becomes

$$\left(-\frac{d^2}{d\tau^2}+3\right)g(\tau) = \left(-\frac{d^2}{d\tau^2}+1\right)e^{-\tau} \tag{4.4.13}$$

and this reduces to

$$-\frac{d^2g(\tau)}{d\tau^2} + 3g(\tau) = 0 \tag{4.4.14}$$

The general solution of this equation is recognized to be

$$g(\tau) = ae^{-\sqrt{3}\tau} + be^{\sqrt{3}\tau} \tag{4.4.15}$$

We do not need to add impulses in this case because $N(s^2)$ and $D(s^2)$ are both second order. Thus, we know the filter weighting function is of the form given by Eq. (4.4.15) without additional impulses. The a and b coefficients may be evaluated by substituting the known form of solution (i.e., Eq. 4.4.15) into the original integral equation, Eq. (4.4.4), and then choosing a and b such that the resulting functions on left and right sides of the equation are identical functions of τ. This is straightforward, but considerable algebra is involved, which will be omitted. The end result is

$$a(t) = \frac{2(\sqrt{3}+1)e^{\sqrt{3}t}}{(\sqrt{3}+1)^2e^{\sqrt{3}t} - (-\sqrt{3}+1)^2e^{-\sqrt{3}t}} \tag{4.4.16}$$

$$b(t) = \frac{-2(-3\sqrt{3}+1)e^{-\sqrt{3}t}}{(\sqrt{3}+1)^2e^{\sqrt{3}t} - (-\sqrt{3}+1)^2e^{-\sqrt{3}t}} \tag{4.4.17}$$

Note that the "constants" are functions of the running time variable t. The final solution for weighting function is then

$$g(\tau;t) = a(t)e^{-\sqrt{3}\tau} + b(t)e^{\sqrt{3}\tau} \tag{4.4.18}$$

where $a(t)$ and $b(t)$ are given by Eqs. (3.4.16) and (3.4.17). The semicolon in $g(\tau;t)$ is used to emphasize the fact that τ is the usual age variable in the weighting function and t is just a parameter. The resulting filter is, of course, a time-variable filter.

It is readily verified that as t approaches ∞, the solution for $g(\tau)$ becomes

$$g(\tau) = (\sqrt{3}-1)e^{-\sqrt{3}\tau} \tag{4.4.19}$$

As should be expected, this is the same steady-state solution that was obtained previously using spectral factorization methods. It is of interest to note that the differential-equation approach provides an alternative method of solving the stationary problem. ■

In Example 4.5 it is worth noting that the running time variable t came into the weighting-function solution naturally (i.e., without any conscious effort) because we chose to write the superposition integral in the form

$$x(t) = \int_0^t g(\tau;t)f(t-\tau)\,d\tau \tag{4.4.20}$$

The other form, which is equally valid, and sometimes preferred in books on linear systems theory (9), is

$$x(t) = \int_0^t h(t,\tau)f(\tau)\,d\tau \tag{4.4.21}$$

In Eq. (4.4.21), $h(t,\tau)$ has the physical meaning of the system response to a unit impulse applied at time τ. The relationship between the impulsive response and weighting function is obtained by making the appropriate change of variable in either Eq. (4.4.20) or Eq. (4.4.21) and then comparing the two integrals. The result is

$$h(t,\tau) = g(t-\tau;t) \tag{4.4.22}$$

4.5 ORTHOGONALITY

It was shown in Section 4.4 that the filter weighting function that minimizes the mean-square error must satisfy the integral equation

$$\int_0^t g(u)R_{s+n}(\tau-u)\,du = R_{s+n,s}(\alpha+\tau), \qquad 0 \le \tau \le t \tag{4.5.1}$$

Also, the filter error is given by

$$e(t) = s(t+\alpha) - x(t) = s(t+\alpha) - \int_0^t g(u)[s(t-u)+n(t-u)]\,du \tag{4.5.2}$$

We wish to examine the expectation of the product of the filter error at time t and input at some time t_1 where $0 \le t_1 \le t$. Let the input be denoted as $z(t)$. Then

$$z(t_1) = s(t_1) + n(t_1) \tag{4.5.3}$$

and

$$E[z(t_1)e(t)] = E\left[\{s(t_1)+n(t_1)\} \times \left\{s(t+\alpha) - \int_0^t g(u)[s(t-u)+n(t-u)]\,du\right\}\right] \tag{4.5.4}$$

Moving $s(t_1) + n(t_1)$ inside the integration and carrying out the expectation operation yield

$$E[z(t_1)e(t)] = R_{s+n,s}(t-t_1+\alpha) - \int_0^t g(u)R_{s+n}(t-t_1-u)\,du \tag{4.5.5}$$

However, $g(u)$ must satisfy the integral equation, Eq. (4.5.1). Thus, the above must be zero for $0 \le (t-t_1) \le t$. Since the time t_1 was assumed to lie between 0 and t, this is equivalent to saying

$$E[z(t_1)e(t)] = 0, \qquad 0 \le t_1 \le t \tag{4.5.6}$$

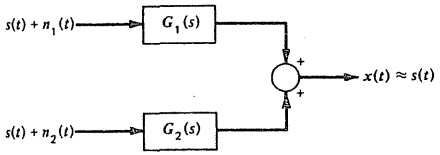


Figure 4.8 General two-input Wiener problem.

If the expectation of the product of two random variables is zero, the variables are said to be *orthogonal*. Equation (4.5.6) states that the filter error at the current time t is not only orthogonal to the input at the same time t, but it is also orthogonal to the input evaluated at any previous time during the past history of the filter operation. This is a consequence of minimizing the mean-square error. Furthermore, it should be apparent from the derivation that the argument can be reversed. That is, if we begin by assuming that $e(t)$ is orthogonal to $z(t_1)$, we can then conclude that the integral equation, Eq. (4.5.1), is satisfied. It is important to recognize this equivalence because some authors prefer to begin their optimality arguments with the orthogonality relationship rather than the minimization of the mean-square error (2, 3, 8).

4.6 COMPLEMENTARY FILTER*

Applications of Wiener filter theory are not as commonplace as one might expect. Perhaps one reason for this is that Wiener theory demands that the signal, as well as the noise, be noiselike in character. In the usual communication problem, this is not the case. The signal usually has at least some deterministic structure, and it is often not reasonable to assume it to be completely random. Thus, the typical filtering problem encountered in communication engineering simply does not fit the Wiener mold. There is an instrumentation application, though, where Wiener methods have been used extensively. In this application, redundant measurements of the same signal are available, and the problem is to combine all the information in such a way as to minimize the instrumentation errors. In order to keep the discussion as simple as possible, we will concentrate on the two-input case and simply mention that the technique is easily extended to more than two inputs.

Consider the general problem of combining two independent noisy measurements of the same signal as depicted in Fig. 4.8. In the context of instrumentation, the measurements might come from two completely different types of instruments, each with its own particular error characteristic. We wish to blend the two measurements together in such a way as to eliminate as much of the error as possible. If the signal $s(t)$ is noiselike, Wiener methods may, in principle,

* The term complementary filter appears to have originated in a paper published in 1953 by W. G. Anderson and E. H. Fritze (10).

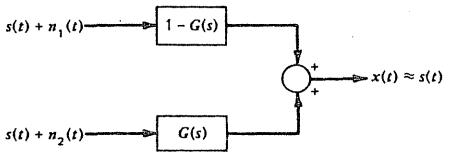


Figure 4.9 Complementary filter.

be used to determine the transfer functions $G_1(s)$ and $G_2(s)$ that minimize the mean-square error. However, more often than not the signal may not be properly modeled as a random process with known spectral characteristics. For example, if $s(t)$ represents the position of an airplane in flight along a prescribed air route, certainly the signal is not random. Furthermore, in this case we would not want to delay or distort the signal in any way in the process of filtering the measurement errors. Thus, we look for a way to filter the signal without paying the price of unwanted delay and distortion.

A method of filtering the noise without distorting the signal is shown in Fig. 4.9. From the block diagram, it should be apparent that the Laplace transform of the output may be written as

$$X(s) = \underbrace{S(s)}_{\text{Signal term}} + \underbrace{N_1(s)[1 - G(s)] + N_2(s)G(s)}_{\text{Noise term}} \tag{4.6.1}$$

Clearly, the signal term $S(s)$ is not affected by our choice of $G(s)$ in any way. On the other hand, the two noise inputs are modified by the complementary transfer functions $[1 - G(s)]$ and $G(s)$. If the two noises have complementary spectral characteristics, $G(s)$ may be chosen to mitigate the noise in both channels. For example, if n_1 is predominantly low-frequency noise and n_2 high-frequency, then choosing $G(s)$ to be a low-pass filter will automatically attenuate n_1 as well as n_2.

We now note that the noise term in Eq. (4.6.1) has the same form as seen before (Eq. 4.2.4) except for the sign on the $N_2(s)$ term. Thus, we would expect to be able to use Wiener methods in minimizing this term. This is perhaps even more evident from Fig. 4.10. It can be easily verified that the input–output relationships are identical for the systems of Figs. 4.9 and 4.10, and thus they are equivalent. From Fig. 4.10 we see that the purpose of the filter $G(s)$ is to

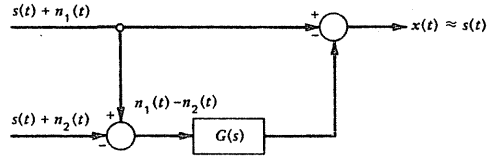


Figure 4.10 Differencing and feedforward configuration for a complementary filter.

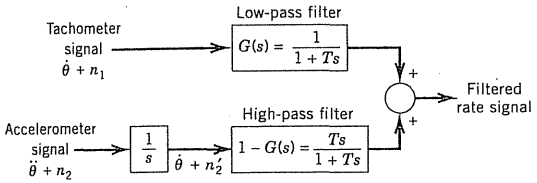


Figure 4.11 Conceptual complementary filter for combining tachometer and accelerometer signals.

give the best possible estimate of $n_1(t)$, and this, in turn, is subtracted from $s(t) + n_1(t)$ in order to give an improved estimate of $s(t)$. The input to $G(s)$ is $n_1(t) - n_2(t)$, and hence the filter must separate one *noiselike* signal from another. Clearly, if we let $n_1(t)$ play the role of signal and $-n_2(t)$ the role of the noise, this problem fits the single-input Wiener theory perfectly. An example will illustrate an engineering application of the complementary-filter method of combining redundant measurement data.

EXAMPLE 4.6

Let us say that in a particular closed-loop position servo, it is desirable to add rate feedback to the system to improve the system stability. The rate signal is to come from a permanent-magnet dc tachometer on the output shaft of the servo. However, it is noticed that the tachometer output is noisy due to the combined commutator/brush action. Low-pass filtering the tachometer signal is a possibility, but this introduces unwanted delay into the rate signal. Someone suggests that if we were to add an angular accelerometer (as well as the tachometer) on the output shaft, then we could use a complementary filter implementation to obtain a clean rate signal without the usual delay.

The suggested scheme is shown in Fig. 4.11. This is a conceptual block diagram, because clearly, we would not want to implement an integration in the acceleration path and then follow it directly with a differentiation. The overall transfer function for this path is just $T/(1 + Ts)$. This, in turn, can be combined with the upper path to yield the simplified block diagram shown in Fig. 4.12.

We have not attempted to optimize the low-pass filter in this example. Rather, we chose the simplest possible form for $G(s)$ that will give the desired low-pass filtering. For simplicity, we have assumed that both n_1 and n_2 are high-frequency noises. This being the case, n_2' will contain primarily low-frequency components because of the integration. The time constant T can be adjusted to

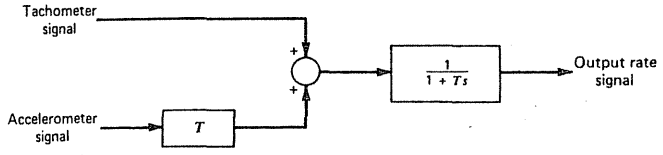


Figure 4.12 Simplified complementary filter for combining tachometer and accelerometer signals.

minimize the effects of the noise sources n_1 and n_2, subject to the constraint on the form that we chose for $G(s)$. It is left as an exercise to find the optimum T, given the power spectral densities of n_1 and n_2 (see Problem 4.14). ▩

The principal of complementary filtering is easily extended to the case of more than two signals. All one has to do is let one of the transfer characteristics be the complement of the sum of the others. For example, for the three-input case, let

$$G_1(s) = \text{transfer function for noisy measurement 1}$$

$$G_2(s) = \text{transfer function for noisy measurement 2}$$

$$1 - G_1(s) - G_2(s) = \text{transfer function for noisy measurement 3}$$

Then the signal component passes through the system undistorted, and one chooses $G_1(s)$ and $G_2(s)$ such as to give the best suppression of the noise. The problem of determining the optimal $G_1(s)$ and $G_2(s)$ is a two-input Wiener problem.

It is important to note that the choice of complementary filter transfer function does not depend on any prior assumptions about the *signal* structure. It can be either noiselike or deterministic, and the complementary feature assures that the signal will not be distorted in any way by the filtering action. For this reason, the complementary filter is also referred to as a *dynamically exact* mechanization. Philosophically, it is a "safe" design and is particularly applicable in situations where the designer wants a filter that will cope reasonably well with statistically unusual situations without giving disastrously large errors. A strict Wiener design with no complementary constraint is, of course, chosen to minimize the *average* squared error. So in the unusual situation, the error may be quite large. This can be disastrous in some applications.

4.7 THE DISCRETE WIENER FILTER

The Wiener approach to least-squares filtering is basically a weighting function approach. When viewed this way, the basic problem always reduces to: How should the past history of the input be weighted in order to yield the present best estimate of the variable of interest? It is instructive to see how this approach extends to the discrete-measurement situation. We could begin by discretizing the continuous weighting function $g(u)$ in Eq. (4.4.4), and then approximate the integral in the equation with a finite sum. This approximation is not necessary, though, because the minimum-mean-square-error problem can be re-posed in discrete-time terms and solved exactly in its own right. This will be our approach here.

Consider the filter input to be a sequence of discrete noisy measurements $z_1, z_2, \ldots, z_n$ as shown in Fig. 4.13. These are additive combinations of signal

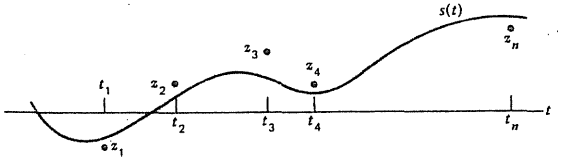


Figure 4.13 Discrete measurement situation.

and noise; hence, $z_1 = s_1 + n_1$, $z_2 = s_2 + n_2$, and so on. As before, we denote the filter output as x, and therefore the corresponding samples of the output are $x_1, x_2, \ldots, x_n$. We now write the output at time t_n as a general linear combination of the past measurements

$$x_n = k_1 z_1 + k_2 z_2 + \cdots + k_n z_n \tag{4.7.1}$$

The filter error may then be written as

$$\begin{aligned} e_n &= s_n - x_n \\ &= s_n - (k_1 z_1 + k_2 z_2 + \cdots + k_n z_n) \end{aligned} \tag{4.7.2}$$

The mean-square error is then

$$\begin{aligned} E(e_n^2) &= E[s_n - (k_1 z_1 + k_2 z_2 + \cdots + k_n z_n)]^2 \\ &= E(s_n^2) + [k_1^2 E(z_1^2) + k_2^2 E(z_2^2) + \cdots + k_n^2 E(z_n^2) \\ &\qquad + 2k_1 k_2 E(z_1 z_2) + 2k_1 k_3 E(z_1 z_3) + \cdots] \\ &\quad - [2k_1 E(z_1 s_n) + 2k_2 E(z_2 s_n) + \cdots + 2k_n E(z_n s_n)] \end{aligned} \tag{4.7.3}$$

We now wish to find $k_1, k_2, \ldots, k_n$ such as to minimize $E(e_n^2)$. The usual methods of differential calculus lead to the following set of linear equations:

$$\begin{bmatrix} E(z_1^2) & E(z_1 z_2) & \cdots & \cdots \\ E(z_2 z_1) & & & \\ \vdots & \ddots & \ddots & \\ E(z_n z_1) & & & E(z_n^2) \end{bmatrix} \begin{bmatrix} k_1 \\ k_2 \\ \vdots \\ k_n \end{bmatrix} = \begin{bmatrix} E(z_1 s_n) \\ E(z_2 s_n) \\ \vdots \\ E(z_n s_n) \end{bmatrix} \tag{4.7.4}$$

Just as in the continuous problem, we assume that the auto- and crosscorrelation functions of the signal and noise are known, so that all the expectations indicated in Eq. (4.7.4) are available and the equations may be solved for the weight factors $k_1, k_2, \ldots, k_n$. Note that the problem grows in size with each new measurement as n increments in time. Also, note that in our notation the ordering of the weight factors is opposite to that in the corresponding continuous problem. That is, the "end" weight factor k_n is the weight given to the current measurement at time t_n, whereas the corresponding weighting in the continuous case is the "beginning" value of $g(u)$, that is, $g(0)$ (see Eq. 4.4.1). It should be

clear that the size of the problem can easily get out of hand numerically as n becomes large. There is a special case that is quite manageable, though, and we will look at it in some detail.

Suppose we consider the stationary problem, and suppose further that the auto- and crosscorrelations for z and s become small as the "lag" between them becomes large. It is then reasonable to assume that the string of weight factors can be truncated at some suitably large value of n. This is to say (in our notation) that if we go backward and look at $k_n, k_{n-1}, \ldots, k_1$, the weight factor k_1 (and nearby weights) will be negligible. Now imagine a stationary real-time situation where we only store the n most recent measurements. As time evolves, every time we get a new measurement, we add it to our suite of measurements, and we discard the oldest one. Presumably, if a stationary condition exists, the truncated $k_1, k_2, \ldots, k_n$ sequence is a constant vector that can be precomputed off-line and stored. The numerical problem of summing n weighted measurements is then quite manageable, even if a few hundred terms are involved; in many applications, this provides sufficient accuracy for the problem at hand.

One nice feature of the truncated discrete-time approach that was just described is that no special mathematical form is required for the auto- and crosscorrelations that describe the signal and noise. This works fine for the stationary, single-input, single-output problem. However, the weight-factor approach becomes completely unwieldy in more complex time-variable, multiple-input, multiple-output applications. Such problems are much better solved using discrete Kalman filtering methods, and this is the subject of Chapter 5 and subsequent chapters.

4.8 PERSPECTIVE

A Wiener filter minimizes the mean-square estimation error subject to certain constraints and assumptions. It is important to remember that this optimization is only intended to apply to the problem of separating one *noiselike* signal from another, which is a very restricted class of filtering problems. Also the assumption of *linear* filtering was built into the derivation from the start. We will see later, in Chapter 5, that this is not a serious restriction if all the random processes involved are Gaussian. In this one case, the linear filter is optimum by almost any reasonable criterion of performance (11). However, the non-Gaussian case is another matter. A nonlinear filter may be better, and the Wiener filter is optimal only within the restricted class of linear filters.

Sometimes, minimization of the mean-square error can lead to seemingly strange physical results; therefore, the results of Wiener filtering should be viewed with a degree of caution. An example will illustrate this. It was mentioned in Section 2.11 that both the random telegraph wave and the Gauss–Markov process have exponential autocorrelation functions. Since the Wiener filter design depends only on the auto- and crosscorrelation functions, the solution for the pure prediction problem will be the same for both random-telegraph and Gauss–Markov signals with the same autocorrelation functions. It

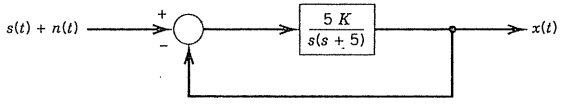
was found in Example 4.4 that the solution in this case is a simple attenuator and, for large prediction time, the predictor output is approximately zero. This makes good sense for the Gauss–Markov signal, because it is noiselike with a central tendency toward zero; in the absence of relevant (i.e., "recent") measurement information, one should just pick the process mean as the best estimate. This way the estimate is only rarely in error by a gross amount, and it is often close to the signal value. On the other hand, to estimate the random telegraph signal to be zero is pure nonsense. We know a priori that the signal is either $+1$ or -1, and it is *never* zero. That is, the Wiener predictor never predicts the correct answer, nor is it even close to the correct answer! We might better arbitrarily choose $+1$ as our estimate. Then, at least we would be correct half the time! Think of the many game (and more serious) situations where you would be better off to be exactly correct half the time (and grossly in error the other half) than to be significantly (but perhaps not grossly) in error *all* the time.

The reason for the strange result in the random-telegraph-wave predictor is that the optimization procedure did not account for the higher-order "statistics" of the process. The mean-square-error criterion is simplistic and only calls for knowledge of the correlation functions of the processes involved. This is, of course, convenient and it also happens to fit the Gaussian process case quite well (for reasons that may not be apparent yet). However, as has just been demonstrated, the mean-square-error criterion of performance can lead to strange results when dealing with non-Gaussian processes.

PROBLEMS

4.1 A closed-loop position control system has the form shown in the block diagram. The spectral density function of the *derivative* of the signal $s(t)$ is given by

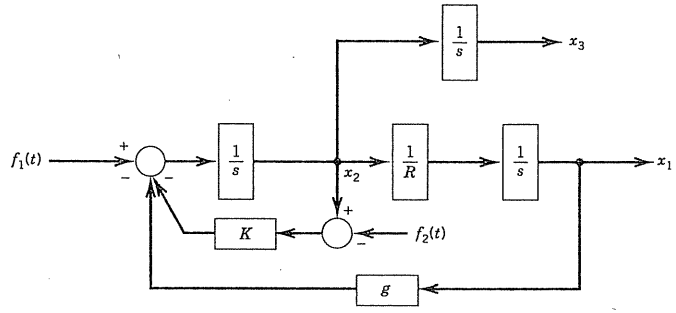
$$S_s(j\omega) = \frac{1000}{\omega^2 + .01}$$



Problem 4.1

and the noise is unity-amplitude Gaussian white noise. Find the value of the gain constant K that will minimize the mean-square error, and find the damping ratio corresponding to this gain.
(*Hint:* The error term that involves the signal spectrum is of the form $S(s)[1 - G(s)]$ (see Eq. 4.2.3). In this problem $[1 - G(s)]$ contains an s factor in the numerator that may be linked with $S(s)$. Since this has the interpretation of *derivative* of $s(t)$ in the time domain, the mean-square error term due to signal may be written in terms of the spectrum of the derivative of the signal, which is the spectral function given in the problem. This problem is worked out as an example in Truxal (12).

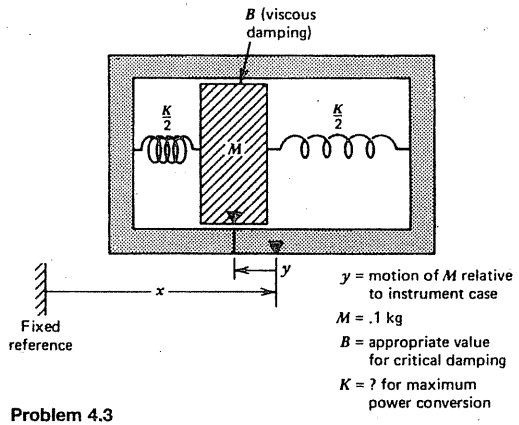
4.2 The figure for Problem 3.21 is repeated here for convenience. Let $f_1(t)$ and $f_2(t)$ be independent white-noise inputs with spectral amplitudes A_1 and A_2, just as in Problem 3.21. Consider x_3 (the inertial system position error) as the output and find the value of K that minimizes the mean-square value of x_3.



R = earth radius = 2.09×10^7 ft
g = gravitational constant = 32.2 ft/sec^2
K = feedback constant (adjustable to yield the desired damping ratio)

Problem 4.2

4.3 One facet of biomedical electronic instrumentation work is that of implanting miniature, telemetering transducers in live animals. This is done in order that various body functions may be observed under normal conditions of activity and environment. When considering such an implant, one is immediately confronted with the problem of supplying energy to the transducer. Batteries are an obvious possibility, but they are bulky and have finite life. Another possibility is to take advantage of the random motion of the animal, and a device for accomplishing this is shown in the simplified diagram of the accompanying figure. The energy conversion takes place in the damping mechanism. This is shown as simple viscous damping, but, in fact, would have to be some sort of electromechanical conversion device. Assuming that all the power indicated as being dissipated in damping can be converted to electrical form, what is the optimum value for the spring constant K and how much power is converted for this optimum condition? The spring–mass arrrangement is to be critically damped, the mass is .1 kg, and the autocorrelation function for the velocity $\dot{x}$ is estimated to be



Problem 4.3

$$R_x(\tau) = 1e^{-2\pi|\tau|} \text{ (ft/sec)}^2$$

[For more details, see Long (13).]

4.4 Find the optimal causal transfer function for the case where the autocorrelation functions of the signal and noise are

$$R_s(\tau) = 2e^{-2|\tau|}$$

$$R_n(\tau) = e^{-|\tau|}$$

Also find the mean-square error. The signal $s(t)$ and the noise $n(t)$ may be assumed to be independent of each other and both are time stationary. Also let the prediction time be zero. In brief, this is the classical Wiener filter problem with zero prediction time.

4.5 Find the optimal noncausal $G(s)$ for the signal and noise situation given in Problem 4.4. Also, find the corresponding weighting function and the resultant mean-square error.

4.6 Consider a stationary Gaussian signal whose spectral density function is

$$S_s(j\omega) = \frac{\omega^2 + 1}{\omega^4 + 8\omega^2 + 16}$$

Find the optimal predictor for the stationary case and $\alpha = 1$.
(*Note*: The causal solution is desired, and the predictor may be specified in terms of either a transfer function or a weighting function.)

4.7 Consider an additive combination of signal and noise where the spectral densities are given by

$$S_s(s) = \frac{2}{-s^2 + 1}$$

$$S_n(s) = \frac{4}{-s^2 + 4}$$

Both the signal and noise are stationary Gaussian processes, and they are statistically independent.

(a) Without regard to causality, what is the optimal linear filter for the stationary case? The answer may be specified in terms of either a transfer function or a weighting function.

(b) Does the noncausal result of (a) have any physical significance? That is, is there a corresponding estimation problem where the computed theoretical mean-square error would have the significance of actual mean-square estimation error? Explain briefly.

4.8 Consider the following autocorrelation functions of the signal and noise in a stationary Wiener prediction problem

$$R_s(\tau) = 4e^{-4|\tau|}$$

$$R_n(\tau) = e^{-|\tau|}$$

The signal and noise are independent and the prediction time is .25 sec. Find the optimal causal weighting function.

4.9 Show that the Wiener–Hopf equation (Eq. 4.3.18) is a sufficient as well as a necessary condition for minimizing the mean-square error.
[*Hint:* Differentiate Eq. (4.3.6) twice with respect to ε and then examine $d^2E(e^2)/d\varepsilon^2$. Recall from elementary calculus that the extremum is a relative minimum if this quantity is positive at the extremum point, which in this case is $\varepsilon = 0$.]

4.10 The orthogonality principle states that the filter error at the current time t is orthogonal to the filter input evaluated at any previous time since initiation of the input. In Section 4.5 this principle was derived from the nonstationary version of the Wiener–Hopf equation (Eq. 4.4.4). Reverse the arguments and show that the integral equation (i.e., Eq. 4.4.4) may be derived from the orthogonality principle. [If you need help on this one, see Davenport and Root (8), p. 240.]

4.11 Consider a noisy measurement of the form

$$z(t) = a_0 + n(t)$$

where a_0 is an unknown random constant with zero-mean normal distribution and a variance of σ^2. The additive noise may be assumed to be white with spectral amplitude A. Find the optimal time-variable filter for estimating a_0. The filter is turned on at $t = 0$.
(*Hint:* A random constant may be thought of as a limiting case of a Markov process with a very large time constant.)

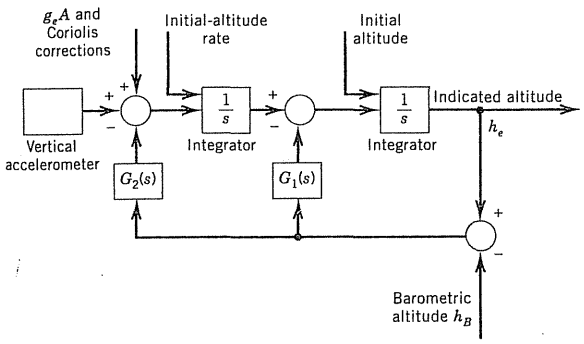
4.12 Verify that the $a(t)$ and $b(t)$ coefficients given in Example 4.5 are correct.

4.13 The figure on the next page was taken from Kayton and Fried (14), p. 318. It describes a means of blending together barometrically derived and inertially derived altitude signals in such a way as to take advantage of the best properties of both signals. The barometric signal, when taken by itself, has large inherent time lag that is undesirable. It is, however, stable and reasonably accurate in the steady-state. On the other hand, integrated vertical acceleration has an unbounded steady-state error because of accelerometer bias error. In effect, the high-frequency response of the accelerometer is good, but its low-frequency response is poor; just the reverse is true for the barometric instrument. Thus this is an ideal setting for a complementary filter application.

Let $G_1(s)$ and $G_2(s)$ in the figure be constant gains G_1 and G_2 and show that:

(a) The system fits the form of a complementary filter as discussed in Section 4.6.

(b) The system error has second-order characteristics with the natural frequency and damping ratio being controlled by the designer's choice of G_1 and G_2.



Problem 4.13

[*Note:* In order to show the complementary-filter property, you must conceptually think of directly integrating the accelerometer signal twice ahead of the summer to obtain an inertially derived altitude signal contaminated with error. However, direct integration of the accelerometer output is not required in the final implementation because of cancellation of s's in numerator and denominator of the transfer function. This will be apparent after carrying through the details of the problem. Also note that the g_e, A, and Coriolis corrections indicated in the figure are simply the gravity, accelerometer-bias, and Coriolis corrections required in order that the accelerometer output be $\ddot{h}$ (as best possible). Also, the initial conditions indicated in the figure may be ignored because the system is stable and we are interested in only the steady-state condition here.]

4.14 This problem is a continuation of Example 4.6. Assume that the tachometer and accelerometer noises are independent, white Gaussian processes with spectral density amplitudes S_T and S_A, respectively. Find the value of the time constant T that minimizes the mean-square error. Express your answer as a function of S_T and S_A.

4.15 The stationary discrete Wiener filter was discussed in Section 4.7. Consider the special case where the signal and noise are uncorrelated and have identical autocorrelation functions:

$$R_s(\tau) = R_n(\tau) = e^{-|\tau|}$$

The sampling interval is .1 sec and the autocorrelation functions are to be truncated with 32 samples (i.e., beginning at $\tau = 0$ and ending at $\tau = 3.1$ sec). Calculate the weight factors $k_1, k_2, \ldots, k_{32}$ for this situation.
[*Hint:* The 32×32 square coefficient matrix of the $\mathbf{k}$ vector in Eq. (4.7.4) is a symmetric Toepliz matrix. This is especially easy to form using the MATLAB function toepliz (**x**), where **x** is a 32×1 vector whose elements are samples of the autocorrelation function of $s + n$.]

Are your results consistent with the causal solution for the corresponding continuous problem? (See Section 4.3.)

4.16 In Problem 4.15 the spectral characteristics of the signal and noise are identical. Thus, the resulting filter is trivial; there can be no effective separation

of signal from noise other than what one gets from a single sample at the current time (multiplied by .5 in this case). Consider now a nontrivial case where the signal and noise are uncorrelated, and their autocorrelation functions are

$$R_s(\tau) = e^{-|\tau|}$$

$$R_n(\tau) = e^{-4|\tau|}$$

Note that the noise in this example has considerably more high-frequency content than the signal. Use 32 samples spaced .1 sec apart just as in Problem 4.15 and calculate the weight factors $k_1, k_2, \ldots, k_{32}$. Compare your result with the solution for the corresponding continuous problem.
(*Note:* The two solutions are not identical, nor should they be. That is, the weight factors in the discrete problem are *not* simply samples of the weighting function in the continuous case.)

REFERENCES CITED IN CHAPTER 4

1. N. Wiener, *Extrapolation, Interpolation, and Smoothing of Stationary Time Series,* New York: Wiley, 1949.
2. R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Trans. ASME—J. Basic Eng.,* 35–45 (March 1960).
3. R. E. Kalman and R. Bucy, "New Results in Linear Filtering and Prediction," *Trans. ASME—J. Basic Eng., 83,* 95 (1961).
4. H. M. James, N. B. Nichols, and R. S. Phillips, *Theory of Servomechanisms* (Vol. 25), Radiation Laboratory Series, New York: McGraw-Hill, 1947.
5. R. Weinstock, *Calculus of Variations,* New York: McGraw-Hill, 1947.
6. H. W. Bode and C. E. Shannon, "A Simplified Derivation of Linear Least Squares Smoothing and Prediction Theory," *Proc. I.R.E., 38,* 417–424 (April 1950).
7. G. R. Cooper and C. D. McGillem, *Probabilistic Methods of Signal and System Analysis,* 2nd ed., New York: Holt, Rinehart, and Winston, 1986.
8. W. B. Davenport, Jr. and W. L. Root, *Introduction to Random Signals and Noise,* New York: McGraw-Hill, 1958.
9. C. T. Chen, *Linear System Theory and Design,* New York: Holt, Rinehart, and Winston, 1984.
10. W. G. Anderson and E. H. Fritze, "Instrument Approach System Steering Computer," *Proc. I.R.E., 41* (Feb. 1953).
11. J. S. Meditch, *Stochastic Optimal Linear Estimation and Control,* New York: McGraw-Hill, 1969.
12. J. G. Truxal, *Automatic Feedback Control System Synthesis,* New York: McGraw-Hill, 1955, pp. 472–477.
13. F. M. Long, *Biological Power Sources,* Ph.D. Dissertation, Ames, IA: Iowa State University, 1961.
14. M. Kayton and W. R. Fried (eds.), *Avionics Navigation Systems,* New York: Wiley, 1969, p. 318.

5

The Discrete Kalman Filter, State-Space Modeling, and Simulation

The end result of the Wiener solution of the optimal filter problem is a filter weighting function in the continuous case, or a set of weight factors in the corresponding discrete problem.* In effect, this tells how the past values of the input should be weighted in order to determine the present value of the output, that is, the optimal estimate. Unfortunately, the Wiener solution does not lend itself very well to the more complicated time-variable, multiple-input/output problem. In 1960, R. E. Kalman provided an alternative way of formulating the minimum mean-square error (MMSE) filtering problem using state-space methods (1). Engineers, especially in the field of navigation, were quick to see the Kalman technique as a practical solution to a number of problems that were previously considered intractable using Wiener methods. The two main features of the Kalman formulation and solution of the problem are (1) vector modeling of the random processes under consideration and (2) recursive processing of the noisy measurement (input) data. We now proceed to the details of the recursive solution of the discrete-data linear filter problem.

5.1 A SIMPLE RECURSIVE EXAMPLE

When working with practical problems involving discrete data, it is important that our methods be computationally feasible as well as mathematically correct.

* Even though the details of Wiener filter theory are not needed for this and subsequent chapters, the reader should at least browse through Chapter 4 and especially Section 4.7 on the discrete Wiener filter.

A simple example will illustrate this. Consider the problem of estimating the mean of some random constant based on a sequence of noisy measurements. Let us assume that our estimate is to be the sample mean and that we wish to refine our estimate with each new measurement as it becomes available. That is, think of processing the data on-line. Let the measurement sequence be denoted as $z_1, z_2, \ldots, z_n$, where the subscript denotes the time at which the measurement is taken. One method of processing the data would be to store each measurement as it becomes available and then compute the sample mean in accordance with the following algorithm (in words):

1. ***First measurement z_1:*** Store z_1 and estimate the mean as

$$\hat{m}_1 = z_1$$

2. ***Second measurement z_2:*** Store z_2 along with z_1 and estimate the mean as

$$\hat{m}_2 = \frac{z_1 + z_2}{2}$$

3. ***Third measurement z_3:*** Store z_3 along with z_1 and z_2 and estimate the mean as

$$\hat{m}_3 = \frac{z_1 + z_2 + z_3}{3}$$

4. And so forth.

Clearly, this would yield the correct sequence of sample means as the experiment progresses. It should also be clear that the amount of memory needed to store the measurements keeps increasing with time, and also the number of arithmetic operations needed to form the estimate increases correspondingly. This would lead to obvious problems when the total amount of data is large. Thus, consider a simple variation in the computational procedure in which each new estimate is formed as a blend of the old estimate and the current measurement. To be specific, consider the following algorithm:

1. ***First measurement z_1:*** Compute the estimate as

$$\hat{m}_1 = z_1$$

   Store $\hat{m}_1$ and discard z_1.

2. ***Second measurement z_2:*** Compute the estimate as a weighted sum of the previous estimate $\hat{m}_1$ and the current measurement z_2:

$$\hat{m}_2 = \tfrac{1}{2}\hat{m}_1 + \tfrac{1}{2}z_2$$

   Store $\hat{m}_2$ and discard z_2 and $\hat{m}_1$.

3. ***Third measurement z_3:*** Compute the estimate as a weighted sum of $\hat{m}_2$ and z_3:

$$\hat{m}_3 = \tfrac{2}{3}\hat{m}_2 + \tfrac{1}{3}z_3$$

Store $\hat{m}_3$ and discard z_3 and $\hat{m}_2$.

4. And so forth. It should be obvious that at the nth stage the weighted sum is

$$\hat{m}_n = \left(\frac{n-1}{n}\right)\hat{m}_{n-1} + \left(\frac{1}{n}\right)z_n$$

Clearly, the above procedure yields the same identical sequence of estimates as before, but without the need to store all the previous measurements. We simply use the result of the previous step to help obtain the estimate at the current step of the process. In this way, the previous computational effort is used to good advantage and not wasted. The second algorithm can proceed on ad infinitum without a growing memory problem. Eventually, of course, as n becomes extremely large, a round-off problem might be encountered. However, this is to be expected with either of the two algorithms.

The second algorithm is a simple example of a *recursive* mode of operation. The key element in any recursive procedure is the use of the results of the previous step to aid in obtaining the desired result for the current step. This is one of the main features of Kalman filtering, and one that clearly distinguishes it from the weight-factor (Wiener) approach.

In order to apply the recursive philosophy to estimation of a random process, it is first necessary that both the process and the measurement noise be modeled in vector form. Thus, we digress for a moment and look at the vector description of a random process. We will then return to the recursive filtering problem in Section 5.5.

5.2 VECTOR DESCRIPTION OF A CONTINUOUS-TIME RANDOM PROCESS

In the material to follow, it is desirable to have the random process description written in vector form. We begin with a scalar process $y(t)$ for which we have a description as discussed in Chapter 2. We now wish to rewrite the description in the following format:

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{u} \tag{5.2.1}$$

$$\mathbf{y} = \mathbf{B}\mathbf{x} \tag{5.2.2}$$

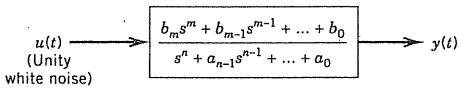


Figure 5.1 Shaping filter ($m = n - 1$).

The vector process under consideration is denoted as $\mathbf{x}$ and is an $(n \times 1)$ column vector. $\mathbf{F}$, $\mathbf{G}$, and $\mathbf{B}$ are rectangular matrices whose elements may be time-varying, and $\mathbf{u}$ is a $(p \times 1)$ column vector whose elements are unity white-noise processes. This is called the *state model* of the process, and, in words, it simply says that we want to think of our process as the result of passing white noise through some system with linear dynamics.* The original process $y(t)$ is required to be a linear combination of the system state variables via Eq. (5.2.2) and, thus, the old process can be recovered from the new model with the appropriate additive combination of state variables.

In general, it is not possible to model all random processes in the form of Eqs. (5.2.1) and (5.2.2). However, many of the processes encountered in physical applications may be so modeled; in particular, all processes with rational spectral density functions have state models with finite dimensionality. A procedure for forming the state model for such processes follows.

State Model for a Process with a Rational Spectral Function

Assume we know the spectral density function $S_y(s)$ for the process $y(t)$. We first factor the spectral function using the same spectral factorization methods used in Chapters 3 and 4. This leads to

$$S_y(s) = S_y^+(s)S_y^-(s) \tag{5.2.3}$$

where we are assured that both the poles and zeros of S_y^+ lie in the left half-plane. Now, it is clear from Section 3.6 that $S_y^+(s)$ is just the shaping filter required to shape unity white noise into the process $y(t)$. The assumed form of the shaping-filter transfer function [i.e., $S_y^+(s)$] is shown in Fig. 5.1, and it is assumed that the order of the denominator is at least one greater than the numerator. This is necessary in order for the $y(t)$ process to have finite variance.

It is now convenient to decompose the block diagram of Fig. 5.1 into the equivalent form shown in Fig. 5.2. With the state variables chosen as the usual phase variables, we are assured that each will have bounded variance. The final state model for the process $y(t)$ is then

* Only an elementary knowledge of state-space methods is needed for our discussion of Kalman filtering. The treatment of state-space methods given in most introductory books on linear control theory (e.g., refs. 2 and 3) is quite adequate for our purposes here.

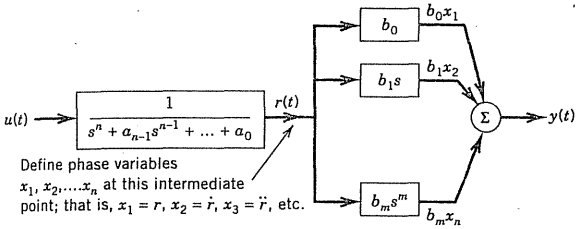


Figure 5.2 Block diagram showing state variables ($m = n - 1$).

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \vdots \\ \dot{x}_n \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots \\ 0 & 0 & 1 & \cdots \\ & & & \\ -a_0 & -a_1 & \cdots & -a_{n-1} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} u(t) \tag{5.2.4}$$

$$y = [b_0 b_1 \cdots b_{n-1}] \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \tag{5.2.5}$$

The special form of the state equations given by Eqs. (5.2.4) and (5.2.5) is known as the controllable canonical form in control theory. Note that the process $y(t)$ is not one of the system state variables directly. Rather, it is reconstructed as a linear combination of state variables via Eq. (5.2.5). Two examples illustrate this modeling procedure.

EXAMPLE 5.1

Consider a random process with a spectral function given by

$$S_y(j\omega) = \frac{16}{\omega^4 - 4\omega^2 + 16} \tag{5.2.6}$$

A sketch of this spectral function is shown in Fig. 5.3 and it shows a mild

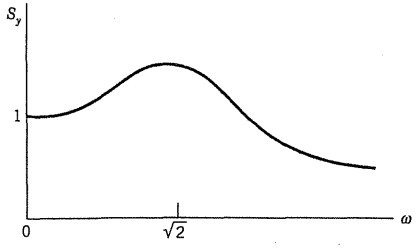


Figure 5.3 Spectral function for Example 5.1.

resonance phenomenon occurring at $\sqrt{2}$ rad/sec. We first need to convert $S_y(j\omega)$ to the s domain by replacing $j\omega$ with s. The result is

$$S_y(s) = \frac{16}{s^4 + 4s^2 + 16} \tag{5.2.7}$$

Since the poles of $S_y(s)$ are at $s = -1 \pm j\sqrt{3}$ and $s = 1 \pm j\sqrt{3}$, $S_y(s)$ can be factored in the form

$$S_y(s) = S_y^+(s) \cdot S_y^-(s) = \frac{4}{s^2 + 2s + 4} \cdot \frac{4}{(-s)^2 + 2(-s) + 4} \tag{5.2.8}$$

A block diagram showing the state and output relationships can now be formed as shown in Fig. 5.4.

The complete state model is then

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -4 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(t) \tag{5.2.9}$$

$$y = [4 \quad 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{5.2.10}$$

EXAMPLE 5.2

The Gauss–Markov process model is frequently used in engineering applications. This is partly because of its simplicity and partly because the analyst often lacks the detailed knowledge of the process under consideration that would be needed to devise a more complicated model. The Gauss–Markov process has an autocorrelation function given by

$$R_y(\tau) = \sigma^2 e^{-\beta|\tau|}$$

The corresponding spectral function is

$$S_y(s) = \frac{2\sigma^2\beta}{-s^2 + \beta^2} = \frac{\sqrt{2\sigma^2\beta}}{s + \beta} \frac{\sqrt{2\sigma^2\beta}}{-s + \beta}$$

Clearly, the shaping filter is

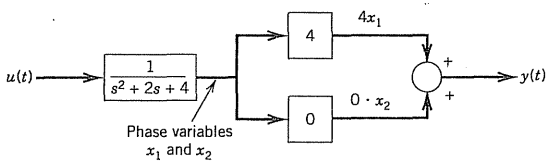


Figure 5.4 Block diagram for state model for Example 5.1.

$$S_y^+(s) = \frac{\sqrt{2\sigma^2\beta}}{s+\beta}$$

and the corresponding differential equation for y is

$$\dot{y} = -\beta y + \sqrt{2\sigma^2\beta}\, u(t) \tag{5.2.11}$$

where $u(t)$ is unity white noise. Equation (5.2.11) is first order, so this is the state model for the process. ■

Nonstationary and Deterministic Processes

The state model of a random process is reasonably general in form and will accommodate a wide variety of situations. All that is required is that the process under consideration be related to white noise via a linear differential equation. A few examples of processes that do not have rational spectral functions will illustrate this further.

1. ***Wiener process (Brownian motion):*** The Wiener process is defined as the integral of Gaussian white noise with zero initial condition. Thus, the appropriate state model is first order and is

$$\dot{y} = ku(t), \qquad y(0) = 0 \tag{5.2.12}$$

where $u(t)$ is unity Gaussian white noise and k is a scale factor.

2. ***Random bias:*** A random constant satisfies the differential equation

$$\dot{y} = 0, \qquad y(0) = a_0 \tag{5.2.13}$$

The initial condition a_0 is a random variable whose distribution is presumed to be known. (See Problems 5.10 and 5.11 for more on modeling random bias and Wiener processes.)

3. ***Random ramp:*** A random ramp process with random initial value and slope may be written as

$$y(t) = a_0 + a_1 t \tag{5.2.14}$$

where a_0 and a_1 are random variables. The differential equation corresponding to Eq. (5.2.14) is

$$\ddot{y} = 0, \qquad y(0) = a_0, \qquad \dot{y}(0) = a_1 \tag{5.2.15}$$

This is a second-order differential equation, so the state vector for the process y must be a two-tuple. Using phase variables in the vector model leads to

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ 0\end{bmatrix}u, \quad \begin{bmatrix}x_1(0)\\ x_2(0)\end{bmatrix} = \begin{bmatrix}a_0\\ a_1\end{bmatrix} \tag{5.2.16}$$

$$y = [1 \quad 0]\begin{bmatrix}x_1\\ x_2\end{bmatrix} \tag{5.2.17}$$

4. ***Harmonic motion with random amplitude and phase:*** A process having pure harmonic motion with known frequency ω_0 rad/sec satisfies the differential equation

$$\ddot{y} + \omega_0^2 y = 0 \tag{5.2.18}$$

The solution of this equation is

$$y(t) = y(0)\cos\omega_0 t + \frac{\dot{y}(0)}{\omega_0}\sin\omega_0 t \tag{5.2.19}$$

The initial conditions $y(0)$ and $\dot{y}(0)$ are random variables.

Model 1:

If we choose phase variables as our state variables, the state model becomes

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ -\omega_0^2 & 0\end{bmatrix}\begin{bmatrix}x_1\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ 0\end{bmatrix}u \tag{5.2.20}$$

$$y = [1 \quad 0]\begin{bmatrix}x_1\\ x_2\end{bmatrix} \tag{5.2.21}$$

This might be called the standard phase-variable model.

Model 2:

There is another model that is also used to describe random harmonic motion. In Eq. (5.2.19) suppose we let the coefficients of the sine and cosine terms be the state variables. Define x_1' and x_2' as follows:

$$x_1' = y(0), \qquad x_2' = -\frac{\dot{y}(0)}{\omega_0} \tag{5.2.22}$$

Then the state equations are

$$\begin{bmatrix}\dot{x}_1'\\ \dot{x}_2'\end{bmatrix} = \begin{bmatrix}0 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}x_1'\\ x_2'\end{bmatrix} + \begin{bmatrix}0\\ 0\end{bmatrix}u \tag{5.2.23}$$

$$y = [\cos\omega_0 t - \sin\omega_0 t]\begin{bmatrix}x_1'\\ x_2'\end{bmatrix} \tag{5.2.24}$$

Note that the $\mathbb{F}$ matrix describing the dynamics in Model 2 is simpler than the $\mathbb{F}$ matrix in Model 1. However, this comes at the expense of a more complicated output matrix. Note that it contains time-variable elements. This is permissible; the model is still linear.

There is a special case of random harmonic motion that leads to Rayleigh amplitude and uniform phase distributions. This occurs when x_1' and x_2' are independent, zero-mean Gaussian random variables with equal variances. This should be apparent from the discussion of narrow-band processes in Section 2.12 and noting that the output equation relating y to x_1' and x_2' can be written as

$$\begin{aligned} y &= x_1' \cos \omega_0 t - x_2' \sin \omega_0 t \\ &= R \cos(\omega_0 t + \theta) \end{aligned} \tag{5.2.25}$$

The preceding examples are just a few of the many processes that can be modeled in vector form. Stationarity is not necessary. The only requirement is that the process must somehow or other be related to white noise through a linear differential equation. It is especially important to note that one does not choose the number of state variables in the model at will; rather, the number of elements in the state vector is fixed by the order of the differential equations relating the various processes in the model to the white-noise driving functions. This is a basic fact of life and must be adhered to if the state model is to faithfully represent the situation at hand. Engineering literature abounds with examples of sloppy modeling and the reader is cautioned accordingly.

Also, remember that the vector model of a random process is not unique. You can always perform any nonsingular linear transformation on a set of state variables and obtain another valid set. Of course, since the $\mathbf{F}$, $\mathbf{G}$, and $\mathbf{B}$ matrices (see Eqs. 5.2.1 and 5.2.2) for the transformed set will be different from those of the original set, the new model will look different. It will, however, be a perfectly proper model if the transformation is done correctly. (See Problem 5.13.)

5.3 DISCRETE-TIME MODEL

Discrete-time processes may arise in either of two ways. First, there is the situation where a sequence of events takes place naturally in discrete steps. This might be the result of a sequence of chance experiments such as the discrete random-walk problems of statistics. Recall in this problem that we imagine the walker taking a number of steps at random, either forward or backward. The length of each step may be either a fixed or a random variable. In either case, the random variable of interest is the distance from the origin after taking n steps. In this problem, there is no such thing as fractional steps. The time variable moves in discrete jumps!

Discrete-time processes may also arise from sampling a continuous process at discrete times. The sampling may be intentional and may be under the control

of the designer, as is the case when analog data are converted to digital form. Or sometimes the sampling may be unintentional and forced on us by a measurement constraint that allows observation of the process only at discrete points in time. For example, the TRANSIT satellite navigation system provides the user with only a single position fix with each satellite pass (4). Thus, the user is allowed to observe his or her position only at discrete points in time. Furthermore, in this case the observation times are not equally spaced because the various satellite orbits do not have perfect symmetry.

Irrespective of how the discretization arises in the physical problem, we wish to fit all situations into the following format:

$$\mathbf{x}_{k+1} = \boldsymbol{\phi}_k \mathbf{x}_k + \mathbf{w}_k \tag{5.3.1}$$

$$\mathbf{y}_k = \mathbf{B}_k \mathbf{x}_k \tag{5.3.2}$$

where

$\mathbf{x}_k$ = vector state of the process at time t_k, that is, $\mathbf{x}_k = \mathbf{x}(t_k)$

$\boldsymbol{\phi}_k$ = matrix that relates $\mathbf{x}_k$ to $\mathbf{x}_{k+1}$ in the absence of a forcing function (in the sampled version of a continuous process, this is the state transition matrix)

$\mathbf{w}_k$ = vector whose elements are white sequences

$\mathbf{B}_k$ = linear connection matrix between output $\mathbf{y}_k$ and state $\mathbf{x}_k$

Recall from Chapter 3 that a white sequence is a sequence of zero-mean random variables that are uncorrelated timewise. Note, however, that the elements of $\mathbf{w}_k$ may have a mutual nontrivial correlation at any point in time t_k. The covariance matrix associated with $\mathbf{w}_k$ is assumed to be known, and it will be denoted as $\mathbf{Q}_k$. Thus, we have

$$E[\mathbf{w}_k \mathbf{w}_i^T] = \begin{cases} \mathbf{Q}_k, & i = k \\ 0, & i \neq k \end{cases} \tag{5.3.3}$$

where "super T" denotes transpose. (We will reserve "prime" for other uses.)

Sampled Continuous-Time Systems

As mentioned previously, the discrete model of Eqs. (5.3.1) and (5.3.2) need not come from a sampled continuous situation, but this is so common in applied work that it warrants amplification. Let us say we begin with a continuous process described by

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{u} \tag{5.3.4}$$

where $\mathbf{u}$ is a vector forcing function whose elements are white noise. Let us consider samples of this process at discrete times $t_0, t_1, \ldots, t_k, \ldots$. State-

space methods may be used to obtain the difference equation relating the samples of $\mathbf{x}$. Specifically, the solution of Eq. (5.3.4) at time t_{k+1} may be written as

$$\mathbf{x}(t_{k+1}) = \boldsymbol{\phi}(t_{k+1}, t_k)\mathbf{x}(t_k) + \int_{t_k}^{t_{k+1}} \boldsymbol{\phi}(t_{k+1}, \tau)\mathbf{G}(\tau)\mathbf{u}(\tau)\, d\tau \tag{5.3.5}$$

Or, in our abbreviated notation,

$$\mathbf{x}_{k+1} = \boldsymbol{\phi}_k\mathbf{x}_k + \mathbf{w}_k$$

Clearly, $\boldsymbol{\phi}_k$ is the state transition matrix for the step from t_k to t_{k+1}, and $\mathbf{w}_k$ is the driven response at t_{k+1} due to the presence of the white-noise input during the (t_k, t_{k+1}) interval. Note that the white-noise input requirement in the continuous model automatically assures that $\mathbf{w}_k$ will be a white sequence in the discrete model.

Analytical methods for finding the state transition matrix are well known, and these may be used in systems with low dimensionality. However, evaluation of the $\mathbf{Q}_k$ matrix that describes $\mathbf{w}_k$ may not be so obvious. Formally, we can write $\mathbf{Q}_k$ in integral form as

$$\begin{aligned}\mathbf{Q}_k &= E[\mathbf{w}_k\mathbf{w}_k^T] \\ &= E\left\{\left[\int_{t_k}^{t_{k+1}} \boldsymbol{\phi}(t_{k+1}, \xi)\mathbf{G}(\xi)\mathbf{u}(\xi)\, d\xi\right]\left[\int_{t_k}^{t_{k+1}} \boldsymbol{\phi}(t_{k+1}, \eta)\mathbf{G}(\eta)\mathbf{u}(\eta)\, d\eta\right]^T\right\} \\ &= \int_{t_k}^{t_{k+1}}\int_{t_k}^{t_{k+1}} \boldsymbol{\phi}(t_{k+1}, \xi)\mathbf{G}(\xi)E[\mathbf{u}(\xi)\mathbf{u}^T(\eta)]\mathbf{G}^T(\eta)\boldsymbol{\phi}^T(t_{k+1}, \eta)\, d\xi\, d\eta\end{aligned} \tag{5.3.6}$$

The matrix $E[\mathbf{u}(\xi)\mathbf{u}^T(\eta)]$ is a matrix of Dirac delta functions that, presumably, is known from the continuous model. Thus, in principle, $\mathbf{Q}_k$ may be evaluated from Eq. (5.3.6). This is not a trivial task, though, even for low-order systems. If the continuous system given rise to the discrete situation has constant parameters and if the various white-noise inputs have zero crosscorrelation, some simplification is possible and the transfer function methods of Chapter 3 may be applied. This is best illustrated with an example rather than in general terms.

EXAMPLE 5.3

The integrated Gauss–Markov process shown in Fig. 5.5 is frequently encountered in engineering applications. The continuous model in this case is

$$\begin{bmatrix}\dot{x}_1 \\ \dot{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1 \\ 0 & -\beta\end{bmatrix}\begin{bmatrix}x_1 \\ x_2\end{bmatrix} + \begin{bmatrix}0 \\ \sqrt{2\sigma^2\beta}\end{bmatrix}u(t) \tag{5.3.7}$$

$$y = [1 \quad 0]\begin{bmatrix}x_1 \\ x_2\end{bmatrix} \tag{5.3.8}$$

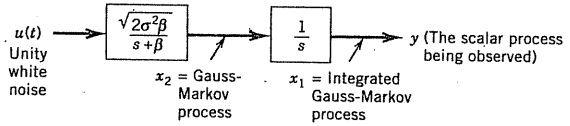


Figure 5.5 Integrated Gauss–Markov process.

Let us say the sampling interval is Δt and we wish to find the corresponding discrete model. This amounts to the determination of $\boldsymbol{\phi}_k$, $\mathbf{Q}_k$, and $\mathbf{B}_k$. The transition matrix is easily determined as*

$$\begin{aligned}\boldsymbol{\phi}_k &= [\mathcal{L}^{-1}[(s\mathbf{I} - \mathbf{F})^{-1}]]_{t=\Delta t} \\ &= \mathcal{L}^{-1}\begin{bmatrix}s & -1 \\ 0 & s+\beta\end{bmatrix}^{-1} = \mathcal{L}^{-1}\begin{bmatrix}\frac{1}{s} & \frac{1}{s(s+\beta)} \\ 0 & \frac{1}{s+\beta}\end{bmatrix} \\ &= \begin{bmatrix}1 & \frac{1}{\beta}(1 - e^{-\beta\Delta t}) \\ 0 & e^{-\beta\Delta t}\end{bmatrix}\end{aligned} \tag{5.3.9}$$

Next, rather than using Eq. (5.3.6) directly to determine $\mathbf{Q}_k$, we use the transfer function approach. From the block diagram of Fig. 5.5, we observe the following transfer functions:

$$G(u \text{ to } x_1) = G_1 = \frac{\sqrt{2\sigma^2\beta}}{s(s+\beta)} \tag{5.3.10}$$

$$G(u \text{ to } x_2) = G_2 = \frac{\sqrt{2\sigma^2\beta}}{s+\beta} \tag{5.3.11}$$

The corresponding weighting functions are

$$g_1(t) = \sqrt{\frac{2\sigma^2}{\beta}}(1 - e^{-\beta t}) \tag{5.3.12}$$

$$g_2(t) = \sqrt{2\sigma^2\beta}\, e^{-\beta t} \tag{5.3.13}$$

* When evaluating $\boldsymbol{\phi}_k$ analytically in higher-order systems, the matrix-inverse and inverse-Laplace-transform operations can become quite laborious. A similar situation applies to the integration that has to be done to obtain a closed-form expression for $\mathbf{Q}_k$. Even so, in analysis work it is often useful to have these closed-form expressions as a function of the step size. Here is a place where the symbolic mathematics capability of MATLAB Version 4 can be helpful, and this is illustrated in M-file ex05_3.m. This M-file is a re-work of Example 5.3 using symbolic mathematics.

We can now use the methods of Chapter 3 to find the needed mean-square responses:

$$\begin{aligned} E[x_1x_1] &= \int_0^{\Delta t}\int_0^{\Delta t} g_1(\xi)g_1(\eta)E[u(\xi)u(\eta)]\,d\xi\,d\eta \\ &= \int_0^{\Delta t}\int_0^{\Delta t} \frac{2\sigma^2}{\beta}(1-e^{\beta\xi})(1-e^{-\beta\eta})\delta(\xi-\eta)\,d\xi\,d\eta \\ &= \frac{2\sigma^2}{\beta}\left[\Delta t - \frac{2}{\beta}(1-e^{-\beta\Delta t}) + \frac{1}{2\beta}(1-e^{-2\beta\Delta t})\right] \end{aligned} \tag{5.3.14}$$

$$\begin{aligned} E[x_1x_2] &= \int_0^{\Delta t}\int_0^{\Delta t} g_1(\xi)g_2(\eta)E[u(\xi)u(\eta)]\,d\xi\,d\eta \\ &= \int_0^{\Delta t}\int_0^{\Delta t} 2\sigma^2 e^{-\beta\xi}(1-e^{-\beta\eta})\delta(\xi-\eta)\,d\xi\,d\eta \\ &= 2\sigma^2\left[\frac{1}{\beta}(1-e^{-\beta\Delta t}) - \frac{1}{2\beta}(1-e^{-2\beta\Delta t})\right] \end{aligned} \tag{5.3.15}$$

$$\begin{aligned} E[x_2x_2] &= \int_0^{\Delta t}\int_0^{\Delta t} g_2(\xi)g_2(\eta)E[u(\xi)u(\eta)]\,d\xi\,d\eta \\ &= \int_0^{\Delta t}\int_0^{\Delta t} 2\sigma^2\beta e^{-\beta\xi}e^{-\beta\eta}\delta(\xi-\eta)\,d\xi\,d\eta \\ &= \sigma^2(1-e^{-2\beta\Delta t}) \end{aligned} \tag{5.3.16}$$

Thus, the $\mathbf{Q}_k$ matrix is

$$\mathbf{Q}_k = \begin{bmatrix} E[x_1x_1] & E[x_1x_2] \\ E[x_1x_2] & E[x_2x_2] \end{bmatrix} = \begin{bmatrix} \text{Eq. (5.3.14)} & \text{Eq. (5.3.15)} \\ \text{Eq. (5.3.15)} & \text{Eq. (5.3.16)} \end{bmatrix} \tag{5.3.17}$$

The $\mathbf{B}_k$ matrix is the same for both the continuous and discrete models and is

$$\mathbf{B}_k = [1 \quad 0] \tag{5.3.18}$$

The discrete model is now complete with the specification of $\boldsymbol{\phi}_k$, $\mathbf{Q}_k$, and $\mathbf{B}_k$ as given by Eqs. (5.3.9), (5.3.17), and (5.3.18). Note that the k subscript could have been dropped in this example because the sampling interval is constant. ■

Numerical Evaluation of $\boldsymbol{\phi}_k$ and $\mathbf{Q}_k$

Analytical methods for finding $\boldsymbol{\phi}_k$ and $\mathbf{Q}_k$ work quite well for constant parameter systems with just a few elements in the state vector. However, the dimensionality does not have to be very large before it becomes virtually impossible to work out explicit expressions for $\boldsymbol{\phi}_k$ and $\mathbf{Q}_k$. This is especially true if the system $\mathbf{F}$ matrix contains time-varying terms. Thus, we often need to resort to numerical methods.

The state transition matrix tells us how the dynamical system naturally relaxes from one state to a subsequent state in the absence of a forcing function. Stated in mathematical terms,

$$\mathbf{x}(t) = \boldsymbol{\phi}(t, t_k)\,\mathbf{x}(t_k), \qquad t \geq t_k \tag{5.3.19}$$

It is easily verified that if $\boldsymbol{\phi}$ satisfies the matrix differential equation

$$\dot{\boldsymbol{\phi}}(t, t_k) = \mathbf{F}(t)\boldsymbol{\phi}(t, t_k), \qquad \boldsymbol{\phi}(t_k, t_k) = \mathbf{I} \tag{5.3.20}$$

then $\mathbf{x}(t)$ as given by Eq. (5.3.19) will satisfy Eq. (5.2.1) with $\mathbf{u} = 0$, which describes the natural dynamics of the system. Clearly, once we have established the differential equation that $\boldsymbol{\phi}$ must satisfy (and the initial condition), then the entire theory of numerical solutions of ordinary differential equations can be brought to bear on the problem (5). Runga–Kutta methods are especially attractive for solving the initial-condition problem, and they carry over nicely from scalar problems to matrix differential equations (6). If $\mathbf{F}(t)$ varies appreciably over the Δt interval of interest, there appears to be no way to avoid solving Eq. (5.3.20) by some means or other. Some simplification does occur, though, for the special case where $\mathbf{F}$ is constant, and that will be considered next.

Assume that $\mathbf{F}$ is constant over the (t_k, t_{k+1}) interval of interest. Then the state transition matrix is simply the matrix exponential of $\mathbf{F}\Delta t$, that is,

$$\boldsymbol{\phi}_k = e^{\mathbf{F}\Delta t} = \mathbf{I} + \mathbf{F}\Delta t + \frac{(\mathbf{F}\Delta t)^2}{2!} + \cdots \tag{5.3.21}$$

where Δt is the step size. This is especially easy to evaluate using MATLAB's built-in expm function. An example will illustrate this.

EXAMPLE 5.4

Consider the harmonic motion process that was discussed in Section 5.2 (see Eq. 5.2.20). For this numerical example we will let $\omega_0 = 1$ and $\Delta t = .1$. $\mathbf{F}\Delta t$ is then

$$\mathbf{F}\Delta t = \begin{bmatrix} 0 & .1 \\ -.1 & 0 \end{bmatrix}$$

Say that we give $\mathbf{F}\Delta t$ the variable name fdt. Then executing MATLAB's expm function yields (with rounding)

$$\text{expm(fdt)} = \begin{bmatrix} .9950 & .0998 \\ -.0998 & .9950 \end{bmatrix}$$

■

The numerical evaluation of $\mathbf{Q}_k$ is not as easy as evaluating $\boldsymbol{\phi}_k$. One method is to use the defining equation, Eq. (5.3.6), and evaluate $\mathbf{Q}_k$ for a small interval using a first-order approximation in Δt. This can then be propagated through a sequence of small steps to get the $\mathbf{Q}_k$ for the whole interval. This is a workable way of evaluating $\mathbf{Q}_k$ if done carefully with very small steps (7). However, it is not the most efficient or convenient method. We will look first at the general time-variable case and then consider the fixed-parameter case later.

It will be shown later in Chapter 7 that $\mathbf{Q}_k$ must satisfy the matrix differential equation:

$$\mathbf{Q}_k(t, t_k) = \mathbf{F}(t)\mathbf{Q}_k(t, t_k) + \mathbf{Q}_k(t, t_k)\mathbf{F}^T(t) + \mathbf{G}(t)\mathbf{W}\mathbf{G}(t)^T, \qquad \mathbf{Q}_k(t_k, t_k) = 0 \tag{5.3.22}$$

where $\mathbf{W}$ is the power spectral density matrix associated with the forcing function $\mathbf{u}$ in Eq. (5.3.4) and $\mathbf{F}(t)$ in the same equation describes the system dynamics. The notation in Eq. (5.3.22) is similar to that used in the transition matrix problem, that is, t is the "running" time variable, t_k is fixed, and $t \geq t_k$. Presumably, $\mathbf{F}$, $\mathbf{G}$, and $\mathbf{W}$ are known, so, in principle, Eq. (5.3.22) can be solved numerically for any step size. This is not a trivial matter, though, and it involves the use of Runga–Kutta (or other) numerical methods (6).

The $\mathbf{Q}_k$ evaluation problem simplifies considerably for the fixed-parameter case. One method due to van Loan (8) is especially convenient when using MATLAB. It proceeds as follows:

1. First, form a $2n \times 2n$ matrix that we will call $\mathbf{A}$ (n is the dimension of $\mathbf{x}$).

$$\mathbf{A} = \left[\begin{array}{c|c} -\mathbf{F} & \mathbf{G}\mathbf{W}\mathbf{G}^T \\ \hline \mathbf{0} & \mathbf{F}^T \end{array}\right] \Delta t \tag{5.3.23}$$

2. Using MATLAB (or other software), form e^A and call it $\mathbf{B}$.

$$\mathbf{B} = \text{expm}(\mathbf{A}) = \left[\begin{array}{c|c} \cdots & \boldsymbol{\phi}^{-1}\mathbf{Q}_k \\ \hline 0 & \boldsymbol{\phi}^T \end{array}\right] \tag{5.3.24}$$

(The upper-left partition of $\mathbf{B}$ is of no concern here.)

3. Transpose the lower-right partition of $\mathbf{B}$ to get $\boldsymbol{\phi}$.

$$\boldsymbol{\phi} = \text{transpose of lower-right partition of } \mathbf{B} \tag{5.3.25}$$

4. Finally, $\mathbf{Q}_k$ is obtained from the upper-right partition of $\mathbf{B}$ as follows:

$$\mathbf{Q}_k = \boldsymbol{\phi} * (\text{upper-right part of } \mathbf{B}) \tag{5.3.26}$$

The method will now be illustrated with an example.

EXAMPLE 5.5

Consider the nonstationary harmonic-motion process described by the differential equation

$$\ddot{y} + y = 2u(t) \tag{5.3.27}$$

where $u(t)$ is unity white noise (see Section 5.2). The continuous state model for this process is then

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}}_{\mathbf{F}} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 2 \end{bmatrix}}_{\mathbf{G}} \begin{bmatrix} 0 \\ u(t) \end{bmatrix} \tag{5.3.28}$$

where x_1 and x_2 are the usual phase variables. In this case, $\mathbf{W}$ is

$$\mathbf{W} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{5.3.29}$$

(The scale factor is accounted for in $\mathbf{G}$.) $\mathbf{G}\mathbf{W}\mathbf{G}^T$ is then

$$\mathbf{G}\mathbf{W}\mathbf{G}^T = \begin{bmatrix} 0 & 0 \\ 0 & 4 \end{bmatrix} \tag{5.3.30}$$

Now form the partitioned $\mathbf{A}$ matrix. Let $\Delta t = .1$ just as in Example 5.4.

$$\mathbf{A} = \begin{bmatrix} -\mathbf{F}\Delta t & \mathbf{G}\mathbf{W}\mathbf{G}^T\Delta t \\ 0 & \mathbf{F}^T\Delta t \end{bmatrix} = \left[\begin{array}{cc|cc} 0 & -.1 & 0 & 0 \\ .1 & 0 & 0 & .4 \\ \hline 0 & 0 & 0 & -.1 \\ 0 & 0 & .1 & 0 \end{array}\right] \tag{5.3.31}$$

The next step is to compute $\mathbf{B} = e^A$. The result is (with numerical rounding)

$$\mathbf{B} = \text{expm}(\mathbf{A}) = \left[\begin{array}{cc|c|c} .9950 & -.0998 & -.0007 & -.0200 \\ .0998 & .9950 & .0200 & .3987 \\ \hline 0 & 0 & .9950 & -.0998 \\ 0 & 0 & .0998 & .9850 \end{array}\right] \tag{5.3.32}$$

Finally, we get both $\boldsymbol{\phi}$ and $\mathbf{Q}_k$ from

$$\boldsymbol{\phi} = \text{transpose of lower-right partition of } \mathbf{B} = \begin{bmatrix} .9950 & .0998 \\ -.0998 & .9950 \end{bmatrix} \text{(same result as in Example 5.4)} \tag{5.3.33}$$

$$\mathbf{Q}_k = \boldsymbol{\phi} * \text{(upper-right partition of } \mathbf{B})$$
$$= \begin{bmatrix} .0013 & .0199 \\ .0199 & .3987 \end{bmatrix} \tag{5.3.34}$$

■

State Model from the ARMA Model

It was mentioned previously that sometimes processes are intrinsically discrete and have nothing to do with a continuous evolution of time. Such processes are often described in terms of an ARMA model, which was touched on briefly in Section 3.9 (9, 10, 11). We will now continue that discussion and show how the ARMA model can be converted to state-space form. We begin by repeating the ARMA model equation given in Section 3.9:

$$y(k+n) + \alpha_{n-1}y(k+n-1) + \alpha_{n-2}y(k+n-2) + \cdots + \alpha_0 y(k)$$
$$= \beta_m w(k+m) + \beta_{m-1}w(k+m-1) + \cdots + \beta_0 w(k),$$
$$m = n-1 \quad \text{and} \quad k = 0, 1, 2, \ldots \tag{5.3.35}$$

Note that the model is single-sided in k and the order of the MA part is at least one less than the AR part. The reasons for these restrictions will be apparent shortly.

Conversion of the ARMA model to the controllable canonical form is effected by following a procedure analogous to that used in the continuous differential equation case. First, we define an intermediate variable $r(k)$, which is the solution of

$$r(k+n) + \alpha_{n-1}r(k+n-1)$$
$$+ \alpha_{n-2}r(k+n-2) + \cdots + \alpha_0 r(k) = w(k) \tag{5.3.36}$$

Next, we define state variables that are analogous to the phase variables in the continuous case:

$$x_1(k) = r(k)$$
$$x_2(k) = r(k+1)$$
$$\vdots$$
$$x_n(k) = r(k+n-1) \tag{5.3.37}$$

Using state variables as defined by Eq. (5.3.37) and the difference equation, Eq. (5.3.36), leads to the matrix equation

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ \vdots \\ x_n(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & \cdots \\ 0 & 0 & 1 & 0 & \cdots \\ \vdots & & & & \\ -\alpha_0 & -\alpha_1 & -\alpha_2 & \cdots & -\alpha_{n-1} \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} w(k) \tag{5.3.38}$$

We now need to reconstruct the original $y(k)$ variable as a linear combination of the elements of the state vector. This can be done by considering a superposition of equations of the type given by Eq. (5.3.36) with the index k advanced appropriately. However, it is a bit easier to do this with a block diagram by using z transforms as shown in Fig. 5.6. The analogy between the block diagrams of Figs. 5.2 and 5.6 should be apparent. It should be clear now that the output equation that takes us from the state space back to $y(k)$ is

$$y(k) = [\beta_0 \quad \beta_1 \quad \beta_2 \quad \cdots \quad \beta_n] \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix} \tag{5.3.39}$$

We will illustrate the procedure with an example.

EXAMPLE 5.6

We will use the same ARMA model here that was introduced in Example 3.11 in Chapter 3. The difference equation is

$$y(k+2) - y(k+1) + .5y(k) = .5w(k+1) + .25w(k), \qquad k = 0, 1, 2, \ldots \tag{5.3.40}$$

The block diagram that leads to the controllable canonical form is shown in Fig. 5.7. The state model is then

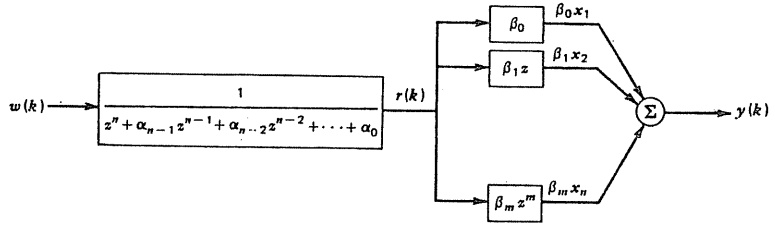


Figure 5.6 Block diagram for constructing state model from ARMA model.

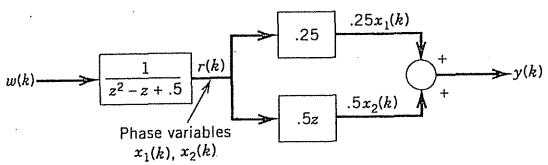


Figure 5.7 Block diagram for state model.

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -.5 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w(k) \tag{5.3.41}$$

$$y(k) = \begin{bmatrix} .25 & .5 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \tag{5.3.42}$$

Eqs. (5.3.41) and (5.3.42) now compose the model.

We can use this same example to demonstrate why we need to put certain restrictions on our ARMA model. First, we restrict k to integer increments beginning at $k = 0$, that is, $k = 0, 1, 2, \ldots$. This is done simply because in the filter theory that follows in Section 5.5 we wish to consider measurement sequences that begin at some finite real time, and it is a matter of convenience to let the point of initiation be $k = 0$ (equivalent to $t = 0$ in the continuous case). We have no problem with letting k (or t) proceed on indefinitely in a positive sense, but time prior to starting the process will be of no consequence in the material that follows. The second restriction put on our ARMA model has to do with the order of the right side of the ARMA equation. Suppose we were to let the right side be of the same order as the left side. This would lead to a transfer function in the z-domain that has the same order polynomial in the numerator as in the denominator. For example, we might then have something like

$$\text{Transfer function} = \frac{z^2 + 3z + 1}{z^2 - z + .5}$$

It can be seen by long division that there will be a direct feedthrough of the input $w(k)$ into the output. We do not wish to allow this in the Kalman filter model that follows in Section 5.5 (at least in its most elementary form), so we will require the order of the MA part to be at least one less than the AR part. There is also a problem of causality when the order of the right side is equal to or greater than the left side, and this will be elaborated on further in Section 5.5.

To summarize, we put the restrictions on our ARMA model because we only wish to consider processes that begin at a finite time, and ones that are causal with no direct feedthrough of the input into the output.

To demonstrate further that the model specified by Eqs. (5.3.41) and (5.3.42) is, in fact, causal and has no direct feedthrough, we will go through three steps of a simple simulation. Suppose that the initial conditions on the two state variables are

$$x_1(0) = x_2(0) = 0$$

These can be specified arbitrarily, because the system is second order. Now suppose that the first three values of the input sequence are $w(0) = 1.0$, $w(1) = -.5$, and $w(2) = .2$ (by chance, of course). We begin at $k = 0$.

At k = 1:

$$\begin{bmatrix} x_1(0) \\ x_2(0) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \text{(this is the given initial condition)}$$

$$y(0) = \begin{bmatrix} .25 & .5 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} = 0$$

Note that the state and output at $k = 0$ do not depend on $w(0)$.

At k = 1:

$$\begin{bmatrix} x_1(1) \\ x_2(1) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -.5 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} (1.0) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$y(1) = \begin{bmatrix} .25 & .5 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = .5$$

Note that the state and output at $k = 1$ depend on $w(0)$ but not on $w(1)$.

At k = 2:

$$\begin{bmatrix} x_1(2) \\ x_2(2) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -.5 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} (-.5) = \begin{bmatrix} 1 \\ .5 \end{bmatrix}$$

$$y(2) = \begin{bmatrix} .25 & .5 \end{bmatrix} \begin{bmatrix} 1 \\ .5 \end{bmatrix} = .5$$

As before, note that the state and output at $k = 2$ do not depend on the "current" value of the input, $w(2)$. This is largely a matter of notation, but it is important in the material that follows in Section 5.5. Conceptually, we can think of the input white sequence unfolding with time one step back of the state and output sequences. That is, when $y(1)$ unfolds, only $w(0)$ is known; when $y(2)$ unfolds, only $w(0)$ and $w(1)$ are known; when $y(3)$ unfolds, only $w(0)$, $w(1)$, and $w(2)$ are known; and so forth. Thinking in terms of real time, at time k nothing is known about $w(k)$ other than the fact that it has zero mean and a known variance. It does not unfold and reveal itself until step $k + 1$.

Before we leave this example, it is of interest to check to see if the original ARMA model, Eq. (5.3.40), is satisfied. We have worked through enough steps to check this at $k = 0$. Substituting the computed values for $y(0)$, $y(1)$, $y(2)$ and the given values of $w(0)$ and $w(1)$ yields

$$(.5) - (.5) + (.5)(0) = (.5)(-.5) + (.25)(1.0)$$

which checks. It might also be noted that the "initial conditions" on the scalar difference equation in y, that is, $y(0)$ and $y(1)$, are not the same as those of the state variables $x_1(0)$ and $x_2(0)$, nor should they be. They are algebraically related, but there is not a one-to-one correspondence, because the state variables are intermediate phase variables and not just $y(k)$ and $y(k + 1)$. ■

In closing it should be mentioned that the state-space model for a given system is not unique. This should be obvious from the fact that we can make any nonsingular transformation on any set of state variables and get another equally legitimate set. The choice of state variables is largely a matter of convenience and may vary from one application to another. The controllable canonical form that was used here is especially convenient in making the transition from a differential (or difference) equation model to a vector model, because the coefficients in the scalar equation transfer directly without any algebraic manipulation. It might also be mentioned that one can also go the other direction in modeling, that is, go from the state model to the scalar model. The conversion is relatively easy and the details are discussed in Problem 5.18.

5.4 MONTE CARLO SIMULATION OF DISCRETE-TIME PROCESSES

Monte Carlo simulation refers to system simulation using random sequences as inputs. Such methods are often helpful in understanding the behavior of stochastic systems that are not amenable to analysis by usual direct mathematical methods. This is especially true of nonlinear filtering problems (considered later in Chapter 9), but there are also many other applications where Monte Carlo methods are useful. Briefly, these methods involve setting up a statistical experiment that matches the physical problem of interest, then repeating the experiment over and over with typical sequences of random numbers, and finally, analyzing the results of the experiment statistically. We are concerned here primarily with experiments where the random processes are Gaussian and sampled in time. We will begin with processes that originate conceptually as continuous processes, and then conclude with a brief discussion of simulating ARMA processes where there need not be any corresponding continuous-time process in the background.

Simulation of Sampled Continuous-Time Random Processes

The usual description of a stationary continuous-time random process is its power spectral density (PSD) function or the corresponding autocorrelation function. It was mentioned in Chapter 2 that the autocorrelation function provides a complete statistical description of the process when it is Gaussian. This is important in Monte Carlo simulation (even though somewhat restrictive), because Gaussian processes have a firm theoretical foundation and this adds credibility to the resulting analysis. In Section 5.3 a general method was given for obtaining a discrete state-space model for a random process, given its power spectral density. Thus, we will begin with a state model of the form given by Eqs. (5.3.1) and (5.3.2) (repeated here for convenience).

$$\mathbf{x}_{k+1} = \boldsymbol{\phi}_k \mathbf{x}_k + \mathbf{w}_k \tag{5.4.1}$$

$$\mathbf{y}_k = \mathbf{B}_k \mathbf{x}_k \tag{5.4.2}$$

Presumably, $\boldsymbol{\phi}_k$ and $\mathbf{B}_k$ are known, and $\mathbf{w}_k$ is a Gaussian white sequence with known covariance $\mathbf{Q}_k$. The problem is to generate an ensemble of random trials of $\mathbf{x}_k$ and $\mathbf{y}_k$ (i.e., sample realizations of the processes) for $k = 0, 1, 2, \ldots, m$.

Equations (5.4.1) and (5.4.2) are explicit. Thus, once methods are established for generating $\mathbf{w}_k$ for $k = 0, 1, 2, \ldots, (m - 1)$ and setting the initial condition for $\mathbf{x}$ at $k = 0$, then programming the few lines of code needed to implement Eqs. (5.4.1) and (5.4.2) is routine. MATLAB is especially useful here because of its "user friendliness" in performing matrix calculations. If $\boldsymbol{\phi}_k$ and $\mathbf{B}_k$ are constants, they are simply assigned variable names and given numerical values in the MATLAB workspace. If $\boldsymbol{\phi}_k$ and $\mathbf{B}_k$ are time-variable, it is relatively easy to reevaluate the parameters with each step as the simulation proceeds in time. Generating the $\mathbf{w}_k$ sequence is a bit more difficult, though, because $\mathbf{Q}_k$ is usually not diagonal. Proceeding on this basis, we begin with a vector $\mathbf{u}_k$ whose components are independent samples from an $N(0, 1)$ population (which is readily obtained in MATLAB), and then operate on this vector with a linear transformation $\mathbf{C}_k$ that is chosen so as to yield a $\mathbf{w}_k$ vector with the desired covariance structure. The desired $\mathbf{C}_k$ is not unique, but a simple way of forming a suitable $\mathbf{C}_k$ is to let it be lower triangular and then solve for the unknown elements. Stated mathematically, we have (temporarily omitting the k subscripts)

$$\mathbf{w} = \mathbf{C}\mathbf{u} \tag{5.4.3}$$

and we demand that

$$E[(\mathbf{C}\mathbf{u})(\mathbf{C}\mathbf{u})^T] = E[\mathbf{w}\mathbf{w}^T] = \mathbf{Q} \tag{5.4.4}$$

Now, $E[\mathbf{u}\mathbf{u}^T]$ is the unitary matrix because of the way we obtain the elements of $\mathbf{u}$ as independent $N(0, 1)$ samples. Therefore,

$$\mathbf{C}\mathbf{C}^T = \mathbf{Q} \tag{5.4.5}$$

We will now proceed to show that the algebra for solving for the elements of $\mathbf{C}$ is simple, provided that the steps are done in the proper order. This will

be demonstrated for a 2×2 $\mathbf{Q}$ matrix. Recall that $\mathbf{Q}$ is symmetric and positive definite. For the 2×2 case, we have (with the usual matrix subscript notation)

$$\begin{bmatrix} c_{11} & 0 \\ c_{21} & c_{22} \end{bmatrix} \begin{bmatrix} c_{11} & c_{21} \\ 0 & c_{22} \end{bmatrix} = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix}$$

or

$$\begin{bmatrix} c_{11}^2 & c_{11}c_{21} \\ c_{11}c_{21} & c_{21}^2 + c_{22}^2 \end{bmatrix} = \begin{bmatrix} q_{11} & q_{12} \\ q_{21} & q_{22} \end{bmatrix} \tag{5.4.6}$$

We start first with the 11 term.

$$c_{11} = \sqrt{q_{11}} \tag{5.4.7}$$

Next, we solve for the 21 term.

$$c_{21} = \frac{q_{12}}{c_{11}} \tag{5.4.8}$$

Finally, c_{22} is obtained as

$$c_{22} = \sqrt{q_{22} - c_{21}^2} \tag{5.4.9}$$

The preceding 2×2 example is a special case of what is known as Cholesky factorization, and it is easily generalized to higher-order cases (12). This procedure factors a symmetric, positive definite matrix into upper- and lower-triangular parts, and MATLAB has a built-in function chol to perform this operation. The user defines a matrix variable, say, QUE, with the numerical values of $\mathbf{Q}$, and then chol(QUE) returns the transpose of the desired $\mathbf{C}$ in the notation used here. This is a very nice feature of MATLAB, and it is a valuable timesaver when dealing with higher-order systems.

It should also be clear that if the transformation $\mathbf{C}$ takes a vector of uncorrelated, unit-variance random variables into a corresponding set of correlated random variables, then $\mathbf{C}^{-1}$ will do just the opposite. If we start with a set of random variables with covariance $\mathbf{Q}$, then $\mathbf{C}^{-1}\mathbf{Q}\mathbf{C}^{-1T}$ will be the covariance of the transformed set. This covariance is, of course, just the identity matrix.

Specifying an appropriate initial condition on $\mathbf{x}$ in the simulation can also be troublesome, and each case has to be considered on its own merits. If the process being simulated is nonstationary, there is no "typical" starting point. This will depend on the definition of the process. For example, a Wiener process is defined to have a zero initial condition. All sample realizations must be initialized at zero in this case. On the other hand, a simple one-state random-walk process can be defined to start with any specified $\mathbf{x}_0$, be it deterministic or random.

If the process being considered is stationary, one usually wants to generate an ensemble of realizations that are stationary throughout the time span of the runs. The initial condition on $\mathbf{x}$ must be chosen carefully to assure this. There is one special case where specification of the initial components of $\mathbf{x}$ is relatively easy. If the process is stationary and the state variables are chosen to be phase variables, (as is the case in the model development shown in Fig. 5.2), it works out that the covariance matrix of the state variables is diagonal in the steady-state condition (see Problem 5.9). Thus, one simply chooses the components of $\mathbf{x}$ as independent samples from an $N(0, 1)$ population appropriately scaled in accordance with the rms values of the process "position," "velocity," "acceleration," and so forth. If the state variables are not phase variables, however, then they will be correlated (in general), and this complicates matters considerably. Sometimes, the most expeditious way of circumventing the problem is to start the simulation with zero initial conditions, then let the process run until the steady-state condition is reached (or nearly so), and finally use just the latter portion of the realization for "serious" analysis. This may not be an elegant solution to the initial-condition problem, but it is effective.

Simulation of ARMA Models

Simulating a Gaussian random process that is described by an ARMA model is simpler than the corresponding discretized continuous-time problem. This is because we have the luxury of starting with a model that is already in discrete form. When doing the simulation with MATLAB, it is convenient to first convert the scalar ARMA equation into vector form. This can be done by inspection without any laborious calculations and was discussed in detail in Section 5.3.

The steps involved in simulating a process defined by a scalar ARMA equation will now be illustrated using the second-order example given in Section 5.3 (Example 5.6). Both the scalar and vector models are repeated here for convenience.

Scalar ARMA Equation:

$$y(k+2) - y(k+1) + .5y(k) = .5w(k+1) + .25w(k), \qquad k = 0, 1, 2, \ldots \tag{5.4.10}$$

Corresponding Vector ARMA Model:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -.5 & 1 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w(k) \tag{5.4.11}$$

$$y(k) = \begin{bmatrix} .25 & .5 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} \tag{5.4.12}$$

The steps in a typical simulation can be summarized as follows:

1. Convert the scalar equation, Eq. (5.4.10), to vector form using the "phase variable" method given in Section 5.3. The result is given by Eqs. (5.4.11) and (5.4.12).
2. A white Gaussian sequence with unity variance is needed for the simulation. This is easily obtained using the MATLAB random number generator.
3. An initial condition on $\mathbf{x}(k)$ must be specified. In this example it is a 2-tuple. The choice of initial condition will govern the initial transient in much the same way as in the continuous-time problem.
4. Programming the recursive equation for $\mathbf{x}(k)$ is routine, once the initial vector $\mathbf{x}(0)$ is specified. Program Eq. (5.4.11).
5. Finally, the original $y(k)$ process is reconstructed from $\mathbf{x}(k)$ at each step using the output equation, Eq. (5.4.12).

In summary, simulation is an important tool in analyzing systems with random inputs. With modern computer technology, simulation is relatively easy to do, and it is often the best way to get answers to otherwise intractable problems.

5.5 THE DISCRETE KALMAN FILTER

R. E. Kalman's paper describing a recursive solution of the discrete-data linear filtering problem was published in 1960 (1). About this same time, advances in digital computer technology made it possible to consider implementing his recursive solution in a number of real-time applications. This was a fortuitous circumstance, and Kalman filtering caught hold almost immediately. We will consider some examples shortly, but we must first develop the Kalman filter recursive equations, which are, in effect, the "filter."

We begin by assuming the random process to be estimated can be modeled in the form

$$\mathbf{x}_{k+1} = \boldsymbol{\phi}_k \mathbf{x}_k + \mathbf{w}_k \tag{5.5.1}$$

The observation (measurement) of the process is assumed to occur at discrete points in time in accordance with the linear relationship

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{v}_k \tag{5.5.2}$$

Some elaboration on notation and the various terms of Eqs. (5.5.1) and (5.5.2) is in order:

$\mathbf{x}_k$ = $(n \times 1)$ process state vector at time t_k

$\boldsymbol{\phi}_k$ = $(n \times n)$ matrix relating $\mathbf{x}_k$ to $\mathbf{x}_{k+1}$ in the absence of a forcing function (if $\mathbf{x}_k$ is a sample of continuous process, $\boldsymbol{\phi}_k$ is the usual state transition matrix)

$\mathbf{w}_k$ = $(n \times 1)$ vector—assumed to be a white sequence with known covariance structure

$\mathbf{z}_k$ = $(m \times 1)$ vector measurement at time t_k

$\mathbf{H}_k$ = $(m \times n)$ matrix giving the ideal (noiseless) connection between the measurement and the state vector at time t_k

$\mathbf{v}_k$ = $(m \times 1)$ measurement error—assumed to be a white sequence with known covariance structure and having zero crosscorrelation with the $\mathbf{w}_k$ sequence

The covariance matrices for the $\mathbf{w}_k$ and $\mathbf{v}_k$ vectors are given by

$$E[\mathbf{w}_k \mathbf{w}_i^T] = \begin{cases} \mathbf{Q}_k, & i = k \\ 0, & i \neq k \end{cases} \tag{5.5.3}$$

$$E[\mathbf{v}_k \mathbf{v}_i^T] = \begin{cases} \mathbf{R}_k, & i = k \\ 0, & i \neq k \end{cases} \tag{5.5.4}$$

$$E[\mathbf{w}_k \mathbf{v}_i^T] = 0, \qquad \text{for all } k \text{ and } i \tag{5.5.5}$$

We assume at this point that we have an initial estimate of the process at some point in time t_k, and that this estimate is based on all our knowledge about the process prior to t_k. This prior (or *a priori*) estimate will be denoted as $\hat{\mathbf{x}}_k^-$ where the "hat" denotes estimate, and the "super minus" is a reminder that this is our best estimate prior to assimilating the measurement at t_k. (Note that super minus as used here is not related in any way to the super minus notation used in spectral factorization.) We also assume that we know the error covariance matrix associated with $\hat{\mathbf{x}}_k^-$. That is, we define the estimation error to be

$$\mathbf{e}_k^- = \mathbf{x}_k - \hat{\mathbf{x}}_k^- \tag{5.5.6}$$

and the associated error covariance matrix is*

$$\mathbf{P}_k^- = E[\mathbf{e}_k^- \mathbf{e}_k^{-T}] = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)^T] \tag{5.5.7}$$

* We tacitly assume here that the estimation error has zero mean, and thus it is proper to refer to $E[\mathbf{e}_k^- \mathbf{e}_k^{-T}]$ as a covariance matrix. It is also, of course, a moment matrix, but it is usually not referred to as such.

In many cases, we begin the estimation problem with no prior measurements. Thus, in this case, if the process mean is zero, the initial estimate is zero, and the associated error covariance matrix is just the covariance matrix of $\mathbf{x}$ itself.

With the assumption of a prior estimate $\hat{\mathbf{x}}_k^-$, we now seek to use the measurement $\mathbf{z}_k$ to improve the prior estimate. We choose a linear blending of the noisy measurement and the prior estimate in accordance with the equation

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-) \tag{5.5.8}$$

where

$\hat{\mathbf{x}}_k$ = updated estimate

$\mathbf{K}_k$ = blending factor (yet to be determined)

The justification of the special form of Eq. (5.5.8) will be deferred until Section 5.8. The problem now is to find the particular blending factor $\mathbf{K}_k$ that yields an updated estimate that is optimal in some sense. Just as in the Wiener solution, we use minimum mean-square error as the performance criterion. Toward this end, we first form the expression for the error covariance matrix associated with the updated (*a posteriori*) estimate.

$$\mathbf{P}_k = E[\mathbf{e}_k\mathbf{e}_k^T] = E[(\mathbf{x}_k - \hat{\mathbf{x}}_k)(\mathbf{x}_k - \hat{\mathbf{x}}_k)^T] \tag{5.5.9}$$

Next, we substitute Eq. (5.5.2) into Eq. (5.5.8) and then substitute the resulting expression for $\hat{\mathbf{x}}_k$ into Eq. (5.5.9). The result is

$$\begin{aligned}\mathbf{P}_k = E\{&[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) - \mathbf{K}_k(\mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-)]\\ &[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-) - \mathbf{K}_k(\mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-)]^T\}\end{aligned} \tag{5.5.10}$$

Now, performing the indicated expectation and noting the $(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)$ is the a priori estimation error that is uncorrelated with the measurement error $\mathbf{v}_k$, we have

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T \tag{5.5.11}$$

Notice here that Eq. (5.5.11) is a perfectly general expression for the updated error covariance matrix, and it applies for any gain $\mathbf{K}_k$, suboptimal or otherwise.

Returning to the optimization problem, we wish to find the particular $\mathbf{K}_k$ that minimizes the individual terms along the major diagonal of $\mathbf{P}_k$, because these terms represent the estimation error variances for the elements of the state vector being estimated. The optimization can be done in a number of ways. We will do this using a straightforward differential calculus approach, and to do so we need two matrix differentiation formulas. They are

$$\frac{d[\text{trace}(\mathbf{AB})]}{d\mathbf{A}} = \mathbf{B}^T \quad (\mathbf{AB} \text{ must be square}) \tag{5.5.12}$$

$$\frac{d[\text{trace}(\mathbf{ACA}^T)]}{d\mathbf{A}} = 2\mathbf{AC} \quad (\mathbf{C} \text{ must be symmetric}) \tag{5.5.13}$$

where the derivative of a scalar with respect to a matrix is defined as

$$\frac{ds}{d\mathbf{A}} = \begin{bmatrix} \dfrac{ds}{da_{11}} & \dfrac{ds}{da_{12}} & \cdots \\ \dfrac{ds}{da_{21}} & \dfrac{ds}{da_{22}} & \cdots \\ \vdots & & \end{bmatrix} \tag{5.5.14}$$

Proof of these two differentiation formulas will be left as an exercise (see Problem 5.16). We will now expand the general form for $\mathbf{P}_k$, Eq. (5.5.11), and rewrite it in the form:

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_k^- - \mathbf{P}_k^-\mathbf{H}_k^T\mathbf{K}_k^T + \mathbf{K}_k(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)\mathbf{K}_k^T \tag{5.5.15}$$

Notice that the second and third terms are linear in $\mathbf{K}_k$ and that the fourth term is quadratic in $\mathbf{K}$. The two matrix differentiation formulas may now be applied to Eq. (5.5.15). We wish to minimize the trace of $\mathbf{P}$ because it is the sum of the mean-square errors in the estimates of all the elements of the state vector. We can use the argument here that the individual mean-square errors are also minimized when the total is minimized, provided that we have enough degrees of freedom in the variation of $\mathbf{K}_k$, which we do in this case. We proceed now to differentiate the trace of $\mathbf{P}_k$ with respect to $\mathbf{K}_k$, and we note that the trace of $\mathbf{P}_k^-\mathbf{H}_k^T\mathbf{K}_k^T$ is equal to the trace of its transpose $\mathbf{K}_k\mathbf{H}_k\mathbf{P}_k^-$. The result is

$$\frac{d(\text{trace } \mathbf{P}_k)}{d\mathbf{K}_k} = -2(\mathbf{H}_k\mathbf{P}_k^-)^T + 2\mathbf{K}_k(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k) \tag{5.5.16}$$

We now set the derivative equal to zero and solve for the optimal gain. The result is

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)^{-1} \tag{5.5.17}$$

This particular $\mathbf{K}_k$, namely, the one that minimizes the mean-square estimation error, is called the *Kalman gain*.

The covariance matrix associated with the optimal estimate may now be computed. Referring to Eq. (5.5.11), we have

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T \tag{5.5.18}$$

$$= \mathbf{P}_k^- - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_k^- - \mathbf{P}_k^-\mathbf{H}_k^T\mathbf{K}_k^T + \mathbf{K}_k(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)\mathbf{K}_k^T \tag{5.5.19}$$

Routine substitution of the optimal gain expression, Eq. (5.5.17), into Eq. (5.5.19) leads to

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{P}_k^-\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)^{-1}\mathbf{H}_k\mathbf{P}_k^- \tag{5.5.20}$$

or

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)\mathbf{K}_k^T \tag{5.5.21}$$

or

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^- \tag{5.5.22}$$

Note that we have four expressions for computing the updated $\mathbf{P}_k$ from the prior $\mathbf{P}_k^-$. Three of these, Eqs. (5.5.20), (5.5.21), and (5.5.22), are only valid for the optimal gain condition. However, Eq. (5.5.18) is valid for any gain, optimal or suboptimal. All four equations yield identical results for optimal gain with perfect arithmetic. We note, though, that in the real engineering world Kalman filtering is a numerical procedure, and some of the *P*-update equations may perform better numerically than others under unusual conditions. More will be said of this later in Chapter 6. For now, we will list the simplest update equation, that is, Eq. (5.5.22), as the usual way to update the error covariance. One should remember, though, that there are alternative equations for implementing the error covariance update.

We now have a means of assimilating the measurement at t_k by the use of Eq. (5.5.8) with $\mathbf{K}_k$ set equal to the Kalman gain as given by Eq. (5.5.17). Note that we need $\hat{\mathbf{x}}_k^-$ and $\mathbf{P}_k^-$ to accomplish this, and we can anticipate a similar need at the next step in order to make optimal use of the measurement $\mathbf{z}_{k+1}$. The updated estimated $\hat{\mathbf{x}}_k$ is easily projected ahead via the transition matrix. We are justified in ignoring the contribution of $\mathbf{w}_k$ in Eq. (5.5.1) because it has zero mean and is not correlated with any of the previous $\mathbf{w}$'s.* Thus, we have

$$\hat{\mathbf{x}}_{k+1}^- = \boldsymbol{\phi}_k\hat{\mathbf{x}}_k \tag{5.5.23}$$

* Recall that in our notation $\mathbf{w}_k$ is the process noise that accumulates during the step ahead from t_k to t_{k+1}. This is purely a matter of notation (but an important one), and in some books it is denoted as $\mathbf{w}_{k+1}$ rather than $\mathbf{w}_k$ (6, 9). Consistency in notation is the important thing here. Conceptually, we are thinking of doing real-time filtering in contrast to smoothing, which we usually think of doing offline (see Chapter 8). Therefore, if we begin with a discrete ARMA model, the white forcing sequence $w(k)$ must conform to this same step-ahead notation, and we need to restrict the order of the MA part of the model to be at least one less than the AR part (see Section 5.3). This then assures us that $w(k)$ contributes only to the state vector *after* time t_k and not before.

Auto Regressive Moving Average

The error covariance matrix associated with $\hat{\mathbf{x}}_{k+1}^-$ is obtained by first forming the expression for the a priori error

$$\begin{aligned}\mathbf{e}_{k+1}^- &= \mathbf{x}_{k+1} - \hat{\mathbf{x}}_{k+1}^- \\ &= (\boldsymbol{\phi}_k\mathbf{x}_k + \mathbf{w}_k) - \boldsymbol{\phi}_k\hat{\mathbf{x}}_k \\ &= \boldsymbol{\phi}_k\mathbf{e}_k + \mathbf{w}_k\end{aligned} \tag{5.5.24}$$

We now note that $\mathbf{w}_k$ and $\mathbf{e}_k$ have zero crosscorrelation, because $\mathbf{w}_k$ is the process noise for the step ahead of t_k. Thus, we can write the expression for $\mathbf{P}_{k+1}^-$ as

$$\begin{aligned}\mathbf{P}_{k+1}^- &= E[\mathbf{e}_{k+1}^-\mathbf{e}_{k+1}^{-T}] = E[(\boldsymbol{\phi}_k\mathbf{e}_k + \mathbf{w}_k)(\boldsymbol{\phi}_k\mathbf{e}_k + \mathbf{w}_k)^T] \\ &= \boldsymbol{\phi}_k\mathbf{P}_k\boldsymbol{\phi}_k^T + \mathbf{Q}_k\end{aligned} \tag{5.5.25}$$

We now have the needed quantities at time t_{k+1}, and the measurement $\mathbf{z}_{k+1}$ can be assimilated just as in the previous step.

Equations (5.5.8), (5.5.17), (5.5.22), (5.5.23), and (5.5.25) comprise the Kalman filter recursive equations. It should be clear that once the loop is entered, it can be continued ad infinitum. The pertinent equations and the sequence of computational steps are shown pictorially in Fig. 5.8. This summarizes what is now known as the *Kalman filter*.

Before we proceed to some examples, it is interesting to reflect on the Kalman filter in perspective. If you were to stumble onto the recursive process of Fig. 5.8 without benefit of previous history, you might logically ask, "Why in the world did somebody call that a filter? It looks more like a computer algorithm." You would, of course, be quite right in your observation. The Kalman filter is just a computer algorithm for processing discrete measurements (the input) into optimal estimates (the output). Its roots, though, go back to the

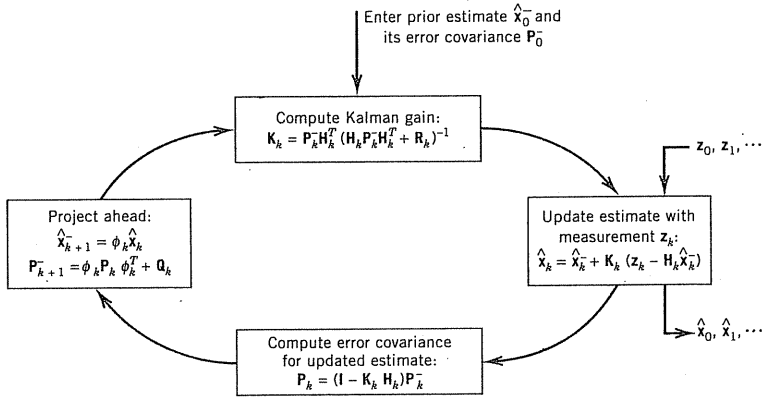


Figure 5.8 Kalman filter loop.

days when filters were made of electrical elements wired together in such a way as to yield the desired frequency response. The design was often heuristic. Wiener then came on the scene in the 1940s and added a more sophisticated type of filter problem. The end result of his solution was a filter weighting function or a corresponding transfer function in the complex domain. Implementation in terms of electrical elements was left as a further exercise for the designer. The discrete-time version of the Wiener problem remained unsolved (in a practical sense, at least) until Kalman's paper of 1960. Even though his presentation appeared to be quite abstract at first glance, engineers soon realized that this work provided a practical solution to a number of unsolved filtering problems, especially in the field of navigation. More than 35 years have elapsed since Kalman's original paper, and there are still numerous current papers dealing with new applications and variations on the basic Kalman filter. It has withstood the test of time!

5.6 SCALAR KALMAN FILTER EXAMPLES

The basic recursive equations for the Kalman filter were presented in Section 5.5. Two simple scalar examples will illustrate the use of these equations.

EXAMPLE 5.7

Wiener (Brownian Motion) Process Consider the Wiener process described in Fig. 5.9. Recall that this process has Gaussian statistics and is assumed to be zero at $t = 0$. Assume that we have a sequence of independent noisy measurements taken at unit intervals as shown in Fig. 5.9 and let the standard deviation of the measurement error be one-half. We wish to determine the optimal estimate of the process at the sample times 0, 1, 2, 3, . . . , etc.

The model parameters for the Kalman filter may be computed as follows: The process differential equation for this case is

$$\dot{x} = u(t) \tag{5.6.1}$$

Therefore, the transition matrix is

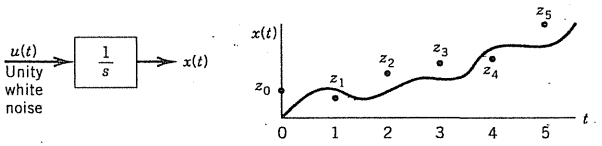


Figure 5.9 Block diagram of Wiener process and typical sample function of the process.

$$\phi_k = 1 \tag{5.6.2}$$

The Q_k matrix is computed as

$$Q_k = E[w_k, w_k] = E\left[\int_0^1 1 \cdot u(\xi)\, d\xi \int_0^1 1 \cdot u(\eta)\, d\eta\right]$$

$$= \int_0^1 \int_0^1 E[u(\xi)u(\eta)]\, d\xi\, d\eta = \int_0^1 \int_0^1 \delta(\xi - \eta)\, d\xi\, d\eta = 1 \tag{5.6.3}$$

Since the measurement has a direct one-to-one correspondence to the process $x(t)$, the H_k matrix is

$$H_k = 1 \tag{5.6.4}$$

The variance of the measurement error is

$$R_k = (\text{Standard deviation})^2 = (\tfrac{1}{2})^2 = \tfrac{1}{4} \tag{5.6.5}$$

Our initial estimate at $t = 0$ is $\hat{x}_0^-$ and is zero because the Wiener process is defined to begin at zero. Furthermore, because of this prior knowledge about the process, the error associated with our initial estimate is zero; that is, the a priori estimate, $\hat{x}_0^- = 0$, is perfect by definition of the process. Thus,

$$P_0^- = 0 \tag{5.6.6}$$

We now have all the parameters needed for the Kalman filter and are ready to begin the recursive process. Referring to Fig. 5.8, we enter the loop at $t = 0$ and process the first measurement.

Step 1: $t = 0$ (Subscripts will be dropped for constant parameters.)
Compute gain:

$$K_0 = P_0^- H^T (H P_0^- H^T + R)^{-1}$$

$$= \frac{0}{.25} = 0 \tag{5.6.7}$$

Update estimate:

$$\hat{x}_0 = \hat{x}_0^- + K_0(z_0 - H\hat{x}_0^-)$$

$$= \hat{x}_0^- = 0 \tag{5.6.8}$$

Update P:

$$P_0 = (I - K_0H)P_0^-$$
$$= 1 \cdot P_0^- = 0 \tag{5.6.9}$$

Project ahead:

$$\hat{x}_1 = \phi\hat{x}_0 = 1 \cdot 0 = 0$$
$$P_1^- = \phi P_0\phi^T + Q \tag{5.6.10}$$
$$= 1 \cdot 0 \cdot 1 + 1 = 1 \tag{5.6.11}$$

Note that the measurement at $t = 0$ is given zero weight relative to the prior estimate. This makes good sense because the initial estimate is known to be perfect, whereas the measurement is known to be noisy.

Step 2: $t = 1$
Compute gain:

$$K_1 = P_1^-H^T(HP_1^-H^T + R)^{-1}$$
$$= 1 \cdot 1(1 \cdot 1 \cdot 1 + \tfrac{1}{4})^{-1} = \tfrac{4}{5} \tag{5.6.12}$$

Update estimate:

$$\hat{x} = \hat{x}_1^- + K_1(z_1 - H\hat{x}_1^-)$$
$$= 0 + \tfrac{4}{5}(z_1) \tag{5.6.13}$$

Update P:

$$P_1 = (I - K_1H)P_1^-$$
$$= (1 - \tfrac{4}{5} \cdot 1)1 = \tfrac{1}{5} \tag{5.6.14}$$

Project ahead:

$$\hat{x}_2^- = \phi\hat{x}_1 = 1 \cdot \hat{x}_1$$
$$P_2^- = \phi P_1\phi_T + Q \tag{5.6.15}$$
$$= 1 \cdot \tfrac{1}{5} \cdot 1 + 1 = \tfrac{6}{5} \tag{5.6.16}$$

This could now be carried on ad infinitum. Note that with step 2 the filter begins to give the measurement some weight in determining the optimal estimate. As times goes on, the filter depends more and more on the measurements and less on the initial assumptions. ■

EXAMPLE 5.8

Scalar Gauss–Markov Process Consider a stationary Gauss–Markov process whose autocorrelation function is

$$R_x(\tau) = 1 \cdot e^{-|\tau|} \tag{5.6.17}$$

Clearly, the correlation time and variance for this process are both unity.

Therefore, the spectral function for the process is

$$S_x(s) = \frac{2}{-s^2 + 1} = \frac{\sqrt{2}}{s + 1} \cdot \frac{\sqrt{2}}{-s + 1} \tag{5.6.18}$$

and the shaping filter that shapes white noise into the process is shown in Fig. 5.10. Thus, the state equation for this process is

$$\dot{x} = -x + \sqrt{2}u(t) \tag{5.6.19}$$

Suppose we have a sequence of noisy measurements of this process taken .02 sec apart beginning at $t = 0$. The measurement error will be assumed to have a variance of unity. We wish to process these via a Kalman filter and obtain an optimal estimate of $x(t)$. First, we need to determine the filter parameters ϕ_k, H_k, Q_k, and R_k.

The state transition matrix (scalar in this case) is

$$\phi_k = e^{-.02} \approx .9802 \tag{5.6.20}$$

The measurement relationship to x is

$$H_k = 1 \tag{5.6.21}$$

The input noise sequence is

$$Q_k = E[w_k^2] = E\left[\int_0^{.02} \sqrt{2}e^{-\xi}u(\xi)\,d\xi \int_0^{.02} \sqrt{2}e^{-\eta}u(\eta)\,d\eta\right]$$
$$= \int_0^{.02} (\sqrt{2}e^{-\eta})^2\,d\eta = 1 - e^{-2(.02)} \approx .03921 \tag{5.6.22}$$

The measurement error variance is

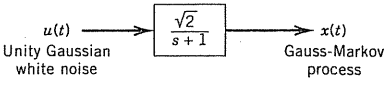


Figure 5.10 Shaping filter for Gauss–Markov process.

$$R_k = 1 \tag{5.6.23}$$

We also need the initial conditions $\hat{x}_0^-$ and P_0^- to enter the recursive loop. In this case, we have assumed that the process is Gauss–Markov and stationary with a known autocorrelation function. We have no measurements prior to $t = 0$, but the assumed knowledge of the process autocorrelation function tells us the process has zero mean and a variance of unity. This is important information and enables us to start the recursive process at $t = 0$ with initial conditions

$$\hat{x}_0^- = 0 \tag{5.6.24}$$

$$P_0^- = 1 \tag{5.6.25}$$

Now that the four filter parameters and initial conditions are determined, it is a routine matter to cycle through the Kalman filter loop shown in Fig. 5.8 as many times as desired.

In order to add a note of realism to this example, the Markov process and noisy measurement situation just described were simulated using the methods described in Section 5.4. The results for the first 51 steps of the simulation are shown in Fig. 5.11. The discrete measurements z_k are shown as triangles, and it should be obvious that we have postulated a very noisy measurement situation for this example. In spite of this, though, the filter does a reasonably good job of tracking $x(t)$. After about the first 20 steps, the filter settles down to a steady-state condition where the Kalman gain is about .165 and the standard deviation of the filter error is about .4. That is, $\sqrt{P_k}$ in the steady state is about .4 units.

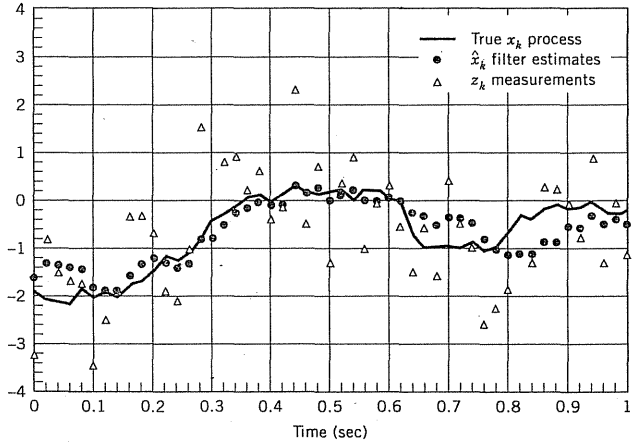


Figure 5.11 Simulation for Gauss–Markov example.

This is consistent, qualitatively at least, with what we see in the results shown in Fig. 5.11. It is comforting to know that our single sample realization of the process is typical rather than atypical. (See Problem 5.19 for a continuation of this example.) ■

5.7 AUGMENTING THE STATE VECTOR AND MULTIPLE-INPUT/MULTIPLE-OUTPUT EXAMPLE

It was mentioned earlier that one of the advantages of the Kalman filter over Wiener methods lies in the convenience of handling multiple-input/multiple-output applications. We will now look at an example of this; in the process, it will be seen that it is sometimes necessary to expand the original state model in order to achieve a credible model that will fit the format required by Eqs. (5.5.1) through (5.5.5). The example that we will consider is based on a paper by B. E. Bona and R. J. Smay that was published in 1966 (14). This paper is of some historical importance, because it was one of the very early applications of real-time Kalman filtering in a terrestial navigation setting. For tutorial purposes, we will consider a simplified version of the state model used in the Bona–Smay paper, and we will then continue the discussion with Problem 5.20.

In marine applications, and especially in the case of submarines, the mission time is usually long, and the ship's inertial navigation system (INS) must operate for long periods without the benefit of position fixes. The major source of position error during such periods is gyro drift. This, in turn, is due to unwanted biases on the axes that control the orientation of the platform (i.e., the inertial instrument cluster). These "biases" may change slowly over long time periods, so they need to be recalibrated occasionally. This is difficult to do at sea, because the biases are hidden from direct one-to-one measurement. One must be content to observe them indirectly through their effect on the inertial system's outputs. Thus, the main function of the Kalman filter in this application is to estimate the three gyro biases and platform azimuth error, so they can be reset to zero. (In this application, the platform tilts are kept close to zero by the gravity vector and by damping the Schuler oscillation with external velocity information from the ship's log.)

In our simplified model, the measurements will be the inertial system's two horizontal position errors, that is, latitude error (N-S direction) and longitude error (E-W direction). These are to be obtained by comparing the INS output with position as determined independently from other sources such as a satellite navigation system, Loran, or perhaps from a known (approximately) position at dockside. The mean-square errors associated with external reference are assumed to be known, and they determine the numerical values assigned to the $\mathbf{R}_k$ matrix of the Kalman filter.

The applicable error propagation equations for a damped inertial navigation system in a slow-moving vehicle are*

$$\dot{\psi}_x - \Omega_z \psi_y = \varepsilon_x \tag{5.7.1}$$

$$\dot{\psi}_y + \Omega_z \psi_x - \Omega_x \psi_z = \varepsilon_y \tag{5.7.2}$$

$$\dot{\psi}_z + \Omega_x \psi_y = \varepsilon_z \tag{5.7.3}$$

where x, y, and z denote the platform coordinate axes in the north, west, and up directions, and

ψ_x = inertial system's west position error (in terms of great circle arc distance in radians)
ψ_y = inertial system's south position error (in terms of great circle arc distance in radians)
ψ_z = [platform azimuth error] − [west position error] · [tan (latitude)]

Also,

Ω_x = x component of earth rate Ω [i.e., $\Omega_x = \Omega \cos(\text{lat.})$]

Ω_z = z component of earth rate Ω [i.e., $\Omega_z = \Omega \sin(\text{lat.})$]

and

ε_x, ε_y, ε_z = gyro drift rates for the x, y, and z axis gyros

We assume that the ship's latitude is known approximately; therefore, Ω_x and Ω_z are known and may be assumed to be constant over the observation interval.

Nonwhite Forcing Functions

The three differential equations, Eqs. (5.7.1) through (5.7.3), represent a third-order linear system with the gyro drift rates, ε_x, ε_y, ε_z, as the forcing functions. These will be assumed to be random processes. However, they certainly are not white noises in this application. Quite to the contrary, they are processes that vary very slowly with time—just the opposite of white noise. Thus, if we were

* Equations (5.7.1) to (5.7.3) are certainly not obvious, and a considerable amount of background in inertial navigation theory is needed to understand the assumptions and approximations leading to this simple set of equations (15, 16). We do not attempt to derive the equations here. For purposes of understanding the Kalman filter, simply assume that these equations do, in fact, accurately describe the error propagation in this application and proceed on to the details of the Kalman filter.

to discretize this third-order system of equations using relatively small sampling intervals, we would find that the resulting sequence of $\mathbf{w}_k$'s would be highly correlated. This would violate the white-sequence assumption that was used in deriving the filter recursive equations (see Eq. 5.5.3). The solution for this is to expand the size of the model and include the forcing functions as part of the state vector. They can be thought of as the result of passing fictitious white noises through a system with linear dynamics. This yields an additional system of equations that can be appended to the original set; in the expanded or augmented set of equations, the new forcing functions will be white noises. We are assured then that when we discretize the augmented system of equations, the resulting $\mathbf{w}_k$ sequence will be white.

In the interest of simplicity, we will model ε_x, ε_y, and ε_z as Gaussian random-walk processes. This allows the "biases" to change slowly with time. Each of the gyro biases can then be thought of as the output of an integrator as shown in Fig. 5.12. The three differential equations to be added to the original set are then

$$\dot{\varepsilon}_x = f_x \tag{5.7.4}$$

$$\dot{\varepsilon}_y = f_y \tag{5.7.5}$$

$$\dot{\varepsilon}_z = f_z \tag{5.7.6}$$

where f_x, f_y, and f_z are independent white-noise processes with power spectral densities equal to W.

We now have a six-state system of linear equations that can be put into the usual state-space form.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \hline \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = \left[\begin{array}{ccc|c} 0 & \Omega_z & 0 & \\ -\Omega_z & 0 & \Omega_x & \mathbf{I} \\ 0 & -\Omega_x & 0 & \\ \hline & \mathbf{0} & & \mathbf{0} \end{array}\right] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \hline x_4 \\ x_5 \\ x_6 \end{bmatrix} + \left[\begin{array}{c|c} \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{I} \end{array}\right] \begin{bmatrix} 0 \\ 0 \\ 0 \\ \hline f_x \\ f_y \\ f_z \end{bmatrix} \tag{5.7.7}$$

The process dynamics model is now in the proper form for a Kalman filter. It is routine to convert the continuous model to discrete form for a given Δt step size. The key parameters in the discrete model are $\boldsymbol{\Phi}_k$ and $\mathbf{Q}_k$, and methods for calculating these are given in Section 5.3.

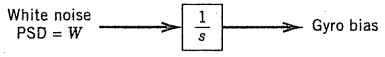


Figure 5.12 Random walk model for gyro bias.

As mentioned previously, we will assume that there are only two measurements available to the Kalman filter at time t_k. They are west position error ψ_x and south position error ψ_y. The matrix measurement equation is then

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix}_k = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{H}_k} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}_k + \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}_k \tag{5.7.8}$$

The measurement model is now complete except for specifying $\mathbf{R}_k$ that describes the mean-square errors associated with the external position fixes. The numerical values will, of course, depend on the particular reference system being used for the fixes.

Nonwhite Measurement Noise

We have just seen an example where it was necessary to expand the state model because the random forcing functions were not white. A similar situation can also occur when the measurement noise is not white. This would also violate one of the assumptions used in the derivation of the Kalman filter equations (see Eq. 5.5.4). The correlated measurement-error problem can also be remedied by augmenting the state vector, just as was done in the preceding gyro-calibration example. The correlated part of the measurement noise is simply moved from $\mathbf{v}_k$ into the state vector, and $\mathbf{H}_k$ is changed accordingly. It should be noted, though, that if the white-noise part of $\mathbf{v}_k$ was zero in the original model, then in the new model, after augmentation, the measurement noise will be zero. In effect, the model is saying that there exists a perfect measurement of certain linear combinations of state variables. The $\mathbf{R}_k$ matrix will then be singular. Technically, this is permissible in the discrete Kalman filter, provided that the $\mathbf{P}_k^-$ matrix that has been projected ahead from the previous step is positive definite, and the measurement situation is not trivial. (See Problem 5.8 for an example where $\mathbf{R}_k = 0$.) The key requirement for permitting a singular $\mathbf{R}_k$ is that $(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)$ be invertible in the gain-computation step. Even so, there is some risk of numerical problems when working with "perfect" measurements. More will be said of this in the section on divergence in Chapter 6.

5.8 THE CONDITIONAL DENSITY VIEWPOINT

In our discussion thus far, we have used minimum mean-square error as the performance criterion, and we have assumed a linear form for the filter. This was partly a matter of convenience, but not entirely so, because we will see

presently that the resulting linear filter has more far-reaching consequences than are apparent at first glance. This is especially so in the Gaussian case. We now elaborate on this.

We first show that if we choose as our estimate the mean of $\mathbf{x}_k$ conditioned on the available measurement stream, then this estimate will minimize the mean-square error. This is a somewhat restrictive form of what is sometimes called the fundamental theorem of estimation theory (10, 17). The same notation and model assumptions that were used in Section 5.5 will be used here, and our derivation follows closely that given by Mendel (10). Also, to save writing, we will temporarily drop the k subscripts, and we will denote the complete measurement stream $\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_k$ simply as $\mathbf{z}^*$. We first write the mean-square estimation error of $\mathbf{x}$, conditioned on $\mathbf{z}^*$, as

$$\begin{aligned} E[(\mathbf{x} - \hat{\mathbf{x}})^T(\mathbf{x} - \hat{\mathbf{x}})|\mathbf{z}^*] &= E[(\mathbf{x}^T\mathbf{x} - \mathbf{x}^T\hat{\mathbf{x}} - \hat{\mathbf{x}}^T\mathbf{x} + \hat{\mathbf{x}}^T\hat{\mathbf{x}})|\mathbf{z}^*] \\ &= E(\mathbf{x}^T\mathbf{x}|\mathbf{z}^*) - E(\mathbf{x}^T|\mathbf{z}^*)\hat{\mathbf{x}} - \hat{\mathbf{x}}^TE(\mathbf{x}|\mathbf{z}^*) + \hat{\mathbf{x}}^T\hat{\mathbf{x}} \end{aligned} \tag{5.8.1}$$

Factoring $\hat{\mathbf{x}}$ away from the expectation operator in Eq. (5.8.1) is justified, because $\hat{\mathbf{x}}$ is a function of $\mathbf{z}^*$, which is the conditioning on the random variable $\mathbf{x}$. We now complete the square of the last three terms in Eq. (5.8.1) and obtain

$$\begin{aligned} &E[(\mathbf{x} - \hat{\mathbf{x}})^T(\mathbf{x} - \hat{\mathbf{x}})|\mathbf{z}^*] \\ &\quad = E(\mathbf{x}^T\mathbf{x}|\mathbf{z}^*) + [\hat{\mathbf{x}} - E(\mathbf{x}|\mathbf{z}^*)]^T[\hat{\mathbf{x}} - E(\mathbf{x}|\mathbf{z}^*)] - E(\mathbf{x}^T|\mathbf{z}^*)E(\mathbf{x}|\mathbf{z}^*) \end{aligned} \tag{5.8.2}$$

Clearly, the first and last terms on the right side of Eq. (5.8.2) do not depend on our choice of the estimate $\hat{\mathbf{x}}$. Therefore, in our search among the admissible estimators (both linear and nonlinear), it should be clear that the best we can do is to force the middle term to be zero. We do this by letting

$$\hat{\mathbf{x}}_k = E(\mathbf{x}_k|\mathbf{z}_k^*) \tag{5.8.3}$$

where we have now reinserted the k subscripts. We have tacitly assumed here that we are dealing with the filter problem, but a similar line of reasoning would also apply to the predictive and smoothed estimates of the $\mathbf{x}$ process. (See Section 4.4 for the definition and a brief discussion of prediction and smoothing.)

Equation (5.8.3) now provides us with a general formula for finding the estimator that minimizes the mean-square error, and it is especially useful in the Gaussian case because it enables us to write out an explicit expression for the optimal estimate in recursive form. Toward this end, we will now assume Gaussian statistics throughout. We will further assume that we have, by some means, an optimal prior estimate $\hat{\mathbf{x}}_k^-$ and its associated error covariance $\mathbf{P}_k^-$. Now, at this point we will stretch our notation somewhat and let $\mathbf{x}_k$ denote the $\mathbf{x}$ random variable at t_k *conditioned* on the measurement stream $\mathbf{z}_{k-1}^*$. We know that the form of the probability density of $\mathbf{x}_k$ is then

$$f_{\mathbf{x}_k} \sim N(\hat{\mathbf{x}}_k^-, \mathbf{P}_k^-) \tag{5.8.4}$$

Now, from our measurement model we know that $\mathbf{x}_k$ is related to $\mathbf{z}_k$ by

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \tag{5.8.5}$$

Therefore, we can immediately write the density function for $\mathbf{z}_k$ as

$$f_{\mathbf{z}_k} \sim N(\mathbf{H}_k\hat{\mathbf{x}}_k^-, \mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k) \tag{5.8.6}$$

(Again, remember that conditioning on $\mathbf{z}_{k-1}^*$ is implied.) Also, from Eq. (5.8.5) we can write out the form for the conditional density of $\mathbf{z}_k$, given $\mathbf{x}_k$. It is

$$f_{\mathbf{z}_k|\mathbf{x}_k} \sim N(\mathbf{H}_k\mathbf{x}_k, \mathbf{R}_k) \tag{5.8.7}$$

Finally, we can now use Bayes formula and write

$$f_{\mathbf{x}_k|\mathbf{z}_k} = \frac{f_{\mathbf{z}_k|\mathbf{x}_k} f_{\mathbf{x}_k}}{f_{\mathbf{z}_k}} \tag{5.8.8}$$

where the terms on the right side of the equation are given by Eqs. (5.8.4), (5.8.6), and (5.8.7). But recall that $\mathbf{x}_k$ itself was conditioned on $\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_{k-1}$. Thus, the density function on the left side of Eq. (5.8.8) is actually the density of the usual random variable $\mathbf{x}_k$, conditioned on the whole measurement stream up through $\mathbf{z}_k$. So, we will change the notation slightly and rewrite Eq. (5.8.8) as

$$f_{\mathbf{x}_k|\mathbf{z}_k} = \frac{[N(\mathbf{H}_k\mathbf{x}_k, \mathbf{R}_k)][N(\hat{\mathbf{x}}_k^-, \mathbf{P}_k^-)]}{[N(\mathbf{H}_k\hat{\mathbf{x}}_k^-, \mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)]} \tag{5.8.9}$$

where it is implied that we substitute the indicated normal functional expressions into the right side of the equation (see Section 1.15 for the vector normal form). It is a routine matter now to make the appropriate substitutions in Eq. (5.8.9) and determine the mean and covariance by inspection of the exponential term. The algebra is routine, but a bit laborious, so we will not pursue it further here (see Problem 5.17). The resulting mean and covariance for $\mathbf{x}_k|\mathbf{z}_k^*$ are

$$\text{Mean} = \hat{\mathbf{x}}_k^- + \mathbf{P}_k^-\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)^{-1}(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-) \tag{5.8.10}$$

$$\text{Covariance} = [(\mathbf{P}_k^-)^{-1} + \mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k]^{-1} \tag{5.8.11}$$

Note that the expression for the mean is identical to the optimal estimate previously derived by other methods and given by Eqs. (5.5.8) and (5.5.17). The expression for the covariance given by Eq. (5.8.11) may not look familiar, but in Chapter 6 it will be shown to be identically equal to the usual $\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-$ expression, provided that $\mathbf{K}_k$ is the Kalman gain.

We also note by comparing Eq. (5.8.10) with Eq. (5.5.8), which was used in the minimum-mean-square-error approach, that the form chosen for the update in Eq. (5.5.8) was correct (for the Gaussian case, at least). Note also that Eq. (5.5.8) can be written in the form

$$\hat{\mathbf{x}}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\hat{\mathbf{x}}_k^- + \mathbf{K}_k\mathbf{z}_k \tag{5.8.12}$$

When the equation is written this way, we see that the updated estimate is formed as a weighted linear combination of two independent measures of $\mathbf{x}_k$; the first is the prior estimate that is the cumulative result of all the past measurements and the prior knowledge of the process statistics, and the second is the new information about $\mathbf{x}_k$ as viewed in the measurement space. Thus, the effective weight factor placed on the new information is $\mathbf{K}_k\mathbf{H}_k$. From Eq. (5.8.12) we see that the weight factor placed on the old information about $\mathbf{x}_k$ is $(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)$, and thus the sum of the weight factors is $\mathbf{I}$ (or just unity in the scalar case). This implies that $\hat{\mathbf{x}}_k$ will be an unbiased estimator, provided, of course, that the two estimates being combined are themselves unbiased estimates. [An estimate is said to be unbiased if $E(\hat{\mathbf{x}}) = \mathbf{x}$.] Note, though, in the Gaussian case we did not start out demanding that the estimator be unbiased. This fell out naturally by simply requiring the estimate to be the mean of the probability density function of $\mathbf{x}_k$, given the measurement stream and the statistics of the process. (This idea of blending together two unbiased estimates to obtain the optimal estimate is exploited further in the chapter on smoothing. See Section 8.5.)

In summary, we see that in the Gaussian case the conditional density viewpoint leads to the same identical result that was obtained in Section 5.5, where we assumed a special linear form for our estimator. There are some far-reaching conclusions that can be drawn from the conditional density viewpoint:

1. Note that in the conditional density function approach, we did not need to *assume* a linear relationship between the estimate and the measurements. Instead, this came out naturally as a consequence of the Gaussian assumption and our choice of the conditional mean as our estimate. Thus, in the Gaussian case, we know that we need not search among nonlinear filters for a better one; it cannot exist. Thus, our earlier linear assumption in the derivation of both the Wiener and Kalman filters turns out to be a fortuitous one. That is, in the Gaussian case, the Wiener–Kalman filter is not just best within a class of linear filters; it is best within a class of all filters, linear or nonlinear.
2. For the Gaussian case, the conditional mean is also the "most likely" value in that the maximum of the density function occurs at the mean. Also, it can be shown that the conditional mean minimizes the expectation of almost any reasonable nondecreasing function of the magnitude of the error (as well as the squared error). [See Meditch (17) for a more complete discussion of this.] Thus, *in the Gaussian case, the Kalman filter is best by almost any reasonable criterion.*
3. In physical problems, we often begin with incomplete knowledge of the process under consideration. Perhaps only the covariance structure of

the process is known. In this case, we can always imagine a corresponding Gaussian process with the same covariance structure. This process is then completely defined and conclusions for the equivalent Gaussian process can be drawn. It is, of course, a bit risky to extend these conclusions to the original process, not knowing it to be Gaussian. However, even risky conclusions are better than none if viewed with proper caution.

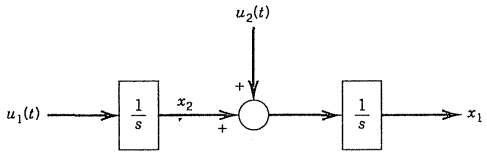
PROBLEMS

5.1 The system shown is driven by two independent Gaussian white sources $u_1(t)$ and $u_2(t)$. Their spectral functions are given by

$$S_{u1} = 4 \text{ (ft/sec}^2\text{)}^2\text{/(rad)/(sec)}$$

$$S_{u2} = 16 \text{ (ft/sec)}^2\text{/(rad)/(sec)}$$

Let state variables be chosen as shown on the diagram, and assume that noisy measurements of x_1 are obtained at unit intervals of time. A discrete Kalman filter model is desired. Find $\mathbf{Q}_k$ for this model.

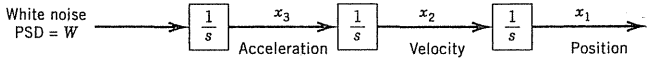


Problem 5.1

5.2 In modern navigation equipment, the Kalman filter is often configured to estimate vehicle acceleration as well as position and velocity. A generic process model for this situation is simply three integrators in cascade as shown in the accompanying figure (for one dimension only). It is usually quite laborious to work out the exact expressions for $\boldsymbol{\phi}_k$ and $\mathbf{Q}_k$, but in this case all the terms in the respective matrices can be found in general form with a modest amount of effort. For a step size of Δt, show that $\boldsymbol{\phi}_k$ and $\mathbf{Q}_k$ are given by

$$\boldsymbol{\phi}_k = \begin{bmatrix} 1 & \Delta t & \Delta t^2/2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix}, \qquad \mathbf{Q}_k = \begin{bmatrix} \frac{W}{20}\Delta t^5 & \frac{W}{8}\Delta t^4 & \frac{W}{6}\Delta t^3 \\ \frac{W}{8}\Delta t^4 & \frac{W}{3}\Delta t^3 & \frac{W}{2}\Delta t^2 \\ \frac{W}{6}\Delta t^3 & \frac{W}{2}\Delta t^2 & W\,\Delta t \end{bmatrix}$$

(*Hint:* Use the same method as in Example 5.3. Also see Problem 5.3 for a variation on this model.)



Problem 5.2

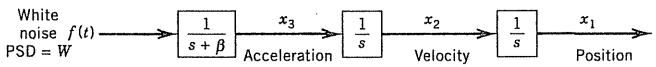
5.3 A variation on the dynamic position-velocity-acceleration (PVA) model given in Problem 5.2 is obtained by modeling acceleration as a Markov process rather than random walk. The model is shown in block-diagram form in the accompanying figure. The linear differential equation for this model is of the form

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{f}$$

(a) Write out the $\mathbf{F}$, $\mathbf{G}$, and $\mathbf{GWG}^T$ matrices for this model, showing each term in the respective matrices explicitly.

(b) Show that the first-order mean-square response (i.e., in Δt) for acceleration is the same here as in the model given in Problem 5.2.

(c) The exact expressions for the terms of $\boldsymbol{\phi}_k$ and $\mathbf{Q}_k$ are considerably more difficult to work out in this model than they were in Problem 5.2. However, their numerical values for any reasonable values of W and β can be found readily using the method referred to as the van Loan method discussed in Section 5.3. Using MATLAB (or other suitable software), find $\boldsymbol{\phi}_k$ and $\mathbf{Q}_k$ for $\beta = 0.2$ sec^{-1}, $W = 10$ (m/sec^2)^2/(rad/sec), and $\Delta t = .1$ sec.
(*Note:* These numerical values yield a relatively high-dynamics model where the sigma of the Markov process is about .5 g with a time constant of 5 sec. Also note that the numerical values obtained in this model correspond to those obtained in Problem 5.2 within a first-order approximation.)



Problem 5.3

5.4 A small intentional random range error is introduced into each of the GPS navigation satellite signals. This is done to degrade the solution accuracy for civil users to 100 m 2 drms (see Chapter 11 for a brief discussion of GPS). One model that has been suggested for this random dithering of the range signal is to model it as a stationary second-order Gauss–Markov process with a power spectral density (PSD) given by (18)

$$S(\mathrm{j}\omega) = \frac{.002585}{\omega^4 + \omega_0^4} \text{ m}^2\text{/(rad/sec)}$$

where

$$\omega_0 = .012 \text{ rad/sec}$$

This PSD corresponds to the autocorrelation function

$$R(\tau) = (23)^2 e^{-\beta|\tau|}(\cos \beta\tau + \sin \beta|\tau|) \text{ m}^2$$

where

$$\beta = \omega_0/\sqrt{2} \text{ rad/sec}$$

(a) First, develop a state model for this process.
(b) Using MATLAB or other suitable software, generate four sample realizations of this process. Let the Δt interval be 10 sec and the total simulated time span be 4000 sec (401 samples for each realization, including the start and end samples.) Give each realization a separate variable name and arrange the sample values as a 1×401 row vector. A rough check on the reasonableness of your simulation results can be obtained by plotting all four sample realizations superimposed on one plot. The results should appear to be stationary with an rms value of about 23 m.
(c) Find the time autocorrelation functions associated with each sample realization of part (b); 40 "lags" will be sufficient. Plot all four sample autocorrelation functions on one plot. Are your results consistent with the theory given in Section 2.15, Chapter 2?
(d) Finally, plot the average of the four sample autocorrelation functions and compare this with the theoretical autocorrelation function given earlier in the problem. Note that one would expect the average experimentally derived autocorrelation function to approximate the theoretical function better than any of the individual functions.

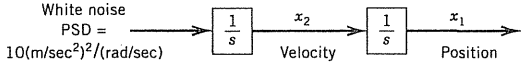
5.5 A stationary Gaussian random process is known to have a spectral density function of the form

$$S_y(j\omega) = \frac{\omega^2 + 1}{\omega^4 + 8\omega^2 + 16}$$

Assume that discrete noisy measurements of y are available at $t = 0, 1, 2, 3, \ldots$ and that the measurement errors are uncorrelated and have a variance of two units. Choose phase variables as state variables and develop a Kalman filter model for this process. That is, find $\boldsymbol{\phi}_k$, $\mathbf{Q}_k$, $\mathbf{H}_k$, $\mathbf{R}_k$ and the initial conditions $\hat{\mathbf{x}}_0^-$ and $\mathbf{P}_0^-$. Note that numerical answers are requested, so MATLAB and the algorithms for determining $\boldsymbol{\phi}_k$ and $\mathbf{Q}_k$ given in Section 5.3 will be helpful. You may assume that in the stationary condition, the x_1 and x_2 state variables are uncorrelated. Thus, $\mathbf{P}_0^-$ will be diagonal. (See Problem 5.9 for more on this.)

5.6 Using the same $\boldsymbol{\phi}_k$, $\mathbf{Q}_k$, $\mathbf{R}_k$, $\mathbf{H}_k$ and $\mathbf{P}_0$ parameters as in Example 5.8, compute the sequence of optimal estimation-error variances for 26 steps using MATLAB. Do this interactively line-by-line following the steps given in Fig. 5.8.
(*Note:* The $\hat{\mathbf{x}}$ update and $\hat{\mathbf{x}}$ projection operations are omitted when doing covariance analysis. Also, a pause statement in the "for" loop will enable you to view the error variance with each recursive step.) The end result is to be a 1×26 row vector containing the a posteriori variances for the 26 steps. Plot the result using the MATLAB plot statement. Note that this is a discrete sequence, so it is best to plot it as a sequence of discrete symbols rather than a continuous curve. Also save the result for future reference. In this example, the estimation error variance reaches a steady-state value of .1653 in about 20 steps. (This problem is continued as Problem 5.7.)

5.7 Having completed Problem 5.6, now write a more general MATLAB covariance analysis program that will accommodate an $n \times 1$ vector process and an $m \times 1$ vector measurement sequence. Make an M-file for your program and give it an appropriate name (e.g., filcovar.m). The number of steps in the Kalman filter is to be s, with the first measurement occurring at $t = 0$ (i.e., step 1 is at $t = 0$). Assume that the step size Δt is fixed and that the system parameters are constant. The desired result is the sequence of optimal a posteriori error covariance matrices for s steps. Arrange your program such that the error covariances are stacked side-by-side into a single large $n \times ns$ matrix that contains all of the error-covariance information for the entire run.
(a) First, test your generalized program on the scalar situation described in Problem 5.6.
(b) Consider next the position-velocity model shown in the accompanying figure. The step size is 1 sec, and position is the measurement. The measurement error variance is 225 m^2, and the initial uncertainties, the x_1 and x_2 estimates, are zero (i.e., $\mathbf{P}_0 = 0$). Do a covariance analysis run for this model for 51 steps. (Last measurement is at $t = 50$ sec.) On the basis of your results, would you say that this system is observable? Give a qualitative justification for your answer.



Problem 5.7

5.8 A classical problem in Wiener-filter theory is one of separating signal from noise when both the signal and noise have exponential autocorrelation functions. Let the noisy measurement be

$$z(t) = s(t) + n(t)$$

and let signal and noise be stationary independent processes with autocorrelation functions

$$R_s(\tau) = \sigma_s^2 e^{-\beta_s|\tau|}$$

$$R_n(\tau) = \sigma_n^2 e^{-\beta_n|\tau|}$$

(a) Assume that we have discrete samples of $z(t)$ spaced Δt apart and wish to form the optimal estimate of $s(t)$ at the sample points. Let $s(t)$ and $n(t)$ be the first and second elements of the process state vector, and then find the parameters of the Kalman filter for this situation. That is, find $\boldsymbol{\phi}_k$, $\mathbf{H}_k$, $\mathbf{Q}_k$, $\mathbf{R}_k$ and the initial conditions $\hat{\mathbf{x}}_0^-$ and $\mathbf{P}_0^-$. Assume that the measurement sequence begins at $t = 0$, and write your results in general terms of σ_s, σ_n, β_s, β_n, and Δt.
(b) To demonstrate that the discrete Kalman filter can be run with $\mathbf{R}_k = 0$ (for a limited number of steps, at least), use the following numerical values in the model developed in part (a):

$$\sigma_s^2 = 9, \qquad \beta_s = .1 \text{ sec}^{-1}$$

$$\sigma_n^2 = 1, \qquad \beta_n = 1 \text{ sec}^{-1}$$

$$\Delta t = 1 \text{ sec}$$

Then run the error covariance part of the Kalman filter for 51 steps beginning at $t = 0$. (You do not need to simulate the $\mathbf{z}_k$ sequence for this problem. Simple covariance analysis will do.)

5.9 Show that the state variables as defined in the model development of Fig. 5.2 are uncorrelated in the steady-state condition. It may be assumed that the poles of the transfer function relating $u(t)$ and $r(t)$ are all in the left-half plane. (*Hint:* Use the methods given in Section 3.8 to evaluate the crosscorrelations, and note that the impulse response begins at zero and approaches zero as $t \to \infty$ when the order of the denominator polynomial is 2 greater than that of the numerator of the corresponding transfer function.)

5.10 In Example 5.7, Section 5.7, the random walk process was assumed to have an initial value of zero, that is, it was a Wiener process. Consider a similar process except that the initial value is a random variable described as $N(0, 1)$. How will this change in initial condition affect the discrete Kalman filter model?

5.11 Suppose we have a scalar process $y(t)$ that is an additive combination of a random bias and a Wiener process (see Section 5.2). The $y(t)$ process can be modeled either as the sum of two separate state variables or as a single variable with a random initial condition as was done in Problem 5.10. Assume that the measurements of $y(t)$ occur at unit intervals beginning at $t = 0$ and that the measurement error has a variance R. Also assume that the random bias component is described as $N(0, \sigma^2)$, and that the Wiener process has a variance that increases in accordance with At. Both σ^2 and A are known parameters.

(a) Demonstrate that both the single-state and two-state models yield the same optimal estimate of y by cycling through two recursive steps for each model.

(b) Assume that only the sum of the random constant and Wiener process is of interest. Which of the two models would you prefer? Why?

5.12 Consider the same Gauss–Markov process and measurement situation discussed in Example 5.8.

(a) Carry out two recursive steps of the Kalman filter and write the estimate at $t = .02$ sec explicitly in terms of z_0 and z_1.

(b) Using the weight factor approach of Section 4.7, write the optimal estimate after two measurements (i.e., at $t = .02$ sec) in terms of z_0 and z_1. [Note that the answer should be the same as that obtained in part (a).]

5.13 Two different state models were given in Section 5.2 for a pure harmonic process. It was also mentioned that a linear transformation will exist that will take one model into the other. Begin with the phase-variable model (Eqs. 5.2.20 and 5.2.21) and

(a) First, write out the explicit matrix transformation $\mathbf{T}$ that takes $\mathbf{x}'$ into $\mathbf{x}$, that is, write $\mathbf{x} = \mathbf{T}\mathbf{x}'$.

(b) Replace $\mathbf{x}$ in the general state equations

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{u}$$

$$\mathbf{y} = \mathbf{B}\mathbf{x}$$

with $\mathbf{T}\mathbf{x}'$ and rearrange the $\mathbf{x}'$ equations into the standard state-space form. The final result of the algebraic manipulations of part (b) should be the same as Eqs. (5.2.23) and (5.2.24).

5.14 Suppose that we make the following linear transformation on the process state vector of Problem 5.8:

$$\begin{bmatrix} x_1' \\ x_2' \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

This transformation is nonsingular, and hence we should be able to consider $\mathbf{x}'$ as the state vector to be estimated and write out the Kalman filter equations accordingly. Specify the Kalman filter parameters for the transformed problem. (*Note:* The specified transformation yields a simplification of the measurement matrix, but this is at the expense of complicating the model elsewhere.)

5.15 It is almost self-evident that if the estimation errors are minimized in one set of state variables, this also will minimize the error in any linear combination of those state variables. This can be shown formally by considering a new state vector $\mathbf{x}'$ to be related to the original state vector $\mathbf{x}$ via a general nonsingular transformation $\mathbf{x}' = \mathbf{A}\mathbf{x}$. Proceeding in this manner, show that the Kalman estimate obtained in the transformed domain is the same as would be obtained by performing the update (i.e., Eq. 5.5.8) in the original $\mathbf{x}$ domain and then transforming this estimate via the $\mathbf{A}$ matrix.

5.16 The matrix differentiation formulas used in Section 5.5 are repeated here for convenience.

$$\frac{d[\text{trace}(\mathbf{AB})]}{d\mathbf{A}} = \mathbf{B}^T \quad (\mathbf{AB} \text{ must be square})$$

$$\frac{d[\text{trace}(\mathbf{ACA}^T)]}{d\mathbf{A}} = 2\mathbf{AC} \quad (\mathbf{C} \text{ must be symmetric})$$

Verify that these formulas are correct. [A straightforward way of doing this is to write out a few terms of the indicated matrices, perform the products, and then differentiate term-by-term as indicated in Eq. (5.5.14).]

5.17 Show that the mean and covariance associated with conditional density function $f_{\mathbf{x}_k|\mathbf{z}_k}$ are as given by Eqs. (5.8.10) and (5.8.11).
(*Hint:* When the expression for $f_{\mathbf{x}_k|\mathbf{z}_k}$ is written out explicitly in terms of $\mathbf{x}_k$, it will be found that the quantity in the exponent is quadratic in $\mathbf{x}_k$. Next, you will need to do the matrix equivalent of completing-the-square in order to recognize the mean and covariance of the density function. Also, you will find the alternative gain and error-covariance expressions derived in Chapter 6 to be useful here. They are, with subscripts omitted,

$$\mathbf{K} = \mathbf{P}\mathbf{H}^T\mathbf{R}^{-1}$$

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

Assume these to be valid and use them wherever they may be helpful in this problem.)

5.18 Occasionally, we need to transform the process state model back to a corresponding scalar differential equation (continuous case) or an ARMA model (discrete case). In either case, we are looking for the direct relationship between the scalar input and the scalar output. In the continuous case the state equation has the following form:

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}u \quad (u \text{ is unit white noise})$$

$$y = \mathbf{B}\mathbf{x} \quad (y \text{ is scalar})$$

If we transform to the s-domain and solve for $Y(s)$, we get

$$Y(s) = [\mathbf{B}(s\mathbf{I} - \mathbf{F})^{-1}\mathbf{G}]U(s)$$

Clearly, the denominator of $[\mathbf{B}(s\mathbf{I} - \mathbf{F})^{-1}\mathbf{G}]$ provides the characteristic polynomial of the desired differential equation, and the numerator gives the coefficients of the forcing function part. Derive the corresponding input–output transfer function relationship in the z-domain for the discrete model and relate the denominator and the numerator to the respective AR and MA parts of the ARMA model. In other words, develop an ARMA model from the discrete state model. Demonstrate that this procedure does yield the correct ARMA equation by applying it to Eqs. (5.3.40) and (5.3.41) of Example 5.6.

5.19 In Example 5.8, the Monte Carlo realizations of the true process and Kalman filter estimate were plotted together for purposes of comparison. It is also informative to plot the estimation *error* for such simulations and plot it with the square root of the error variance for comparison. One simulation run may not be very conclusive, but when the results of a number of Monte Carlo runs are superimposed, one can get a good indication as to whether or not the filter (i.e., estimator) is performing as expected.

Use the same parameters as given in Example 5.8, except here let the correlation time of the process be 4 sec, rather than 1 sec (i.e., let $\beta = .25$ sec^{-1}). Then compute four Kalman filter simulations and save the estimation *error* sequence for each run as a 1×51 row vector. Then plot, superimposed on one plot, (a) the four simulated error trajectories, and (b) both the positive and negative values of $\sqrt{P}$. (It will help here to use continuous plots for the four estimation error curves and discrete symbols for $\sqrt{P}$.) You will find that the two-valued $\sqrt{P}$ plot forms sort of an envelope for the estimation error plots. With Gaussian statistics, one would expect the estimation errors to stay within the envelope about 68 percent of the time. If your results do not agree with this, check your program carefully for possible errors. (Experience has shown that the MATLAB random number generator does a very good job of generating Gaussian statistics when the "normal" option is used.)

5.20 The following numerical exercise is a continuation of the gyro-bias calibration example that was discussed in Section 5.7. For our analysis here, we will approximate the earth as being spherical, and the ship will be moving slowly at a latitude of 45 deg. We will use the following numerical values for the earth's rotation rate and its radius at 45 deg latitude:

$$\Omega = .2625161 \text{ rad/hr} \quad \text{(rotation rate)}$$

$$r_e = 6367253 \text{ m} \quad \text{(radius)}$$

(a) First, look at the natural modes of oscillation of the original set of three differential equations, that is, Eqs. (5.7.1 through 5.7.3). This is easily done by setting the forcing function equal to zero and finding the eigenvalues of $\mathbf{F}$ [roots of $(s\mathbf{I}\text{-}\mathbf{F})$ for the third-order system]. Note that the natural period of oscillation works out to be very long—about 1 solar day! Thus, it is convenient numerically to express time in units of hours, rather than the usual seconds, in this example.

(b) Next, consider the augmented six-state model as discussed in Section 5.7. The gyro biases are to be modeled as random-walk processes, and let the white-noise forcing functions for these processes have power spectral densities that are such as to produce an incremental random-walk variance of $(.001 \text{ deg/hr})^2$ in 1 hr. Discrete measurements will be assumed to be available at intervals of .5 hr. Compute $\boldsymbol{\Phi}_k$ and $\mathbf{Q}_k$ for this model, and check for reasonableness in view of the very long natural period of oscillation for the system.

(c) In the simplified model discussed in Section 5.7, the two horizontal position errors of the inertial system were the measurements. There are problems with system observability for this measurement situation. For this reason, Bona and Smay also considered the possibility of a third external measurement, namely, the platform azimuth error (14). Call it Φ_z and note that it is a linear combination of the state variables ψ_x and ψ_z. At 45 deg latitude, we have

$$\Phi_z = \psi_x + \psi_z \tag{P5.20.1}$$

After we add the azimuth measurement to the two position errors, the $\mathbf{H}_k$ matrix becomes

$$\mathbf{H}_k = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{P5.20.2}$$

We now wish to see how well the inertial system can be calibrated on the basis of a 3-tuple measurement as described by Eq. (P5.21.2). To do this, run a Kalman filter error-covariance analysis for 11 steps with

the measurement interval set at .5 hr. Use the following variances for the initial $\mathbf{P}_0$ matrix:

Position uncertainty (ψ_x and ψ_y): $\left(\frac{1000 \text{ m}}{r_e \text{ (in m)}}\right)^2$

Azimuth and west position combined (ψ_z): (.1 deg)^2

Gyro drift rates (all the same and independent): (.02 deg/hr)^2

The $\mathbf{P}_0$ matrix may be assumed to be diagonal. For the measurement errors, assume independence and let the variance be

Position measurements: $\left(\frac{100 \text{ m}}{r_e \text{ (in m)}}\right)^2$

Azimuth: (1 arc min)^2

The $\mathbf{R}_k$ matrix will be a diagonal 3×3 matrix.

Now run the covariance analysis for 11 steps (beginning at $t = 0$ and ending at $t = 5.0$ hr) and plot the rms estimation errors for ψ_x, ψ_y, ψ_z, ε_x, ,ε_y, and ε_z. On the basis of these plots, would you think that the system is completely observable?

5.21 Consider two different measurement situations for the same random-walk dynamical process:

Process model:

$$x_{k+1} = x_k + w_k$$

Measurement model 1:

$$z_k = .5x_k + v_k$$

Measurement model 2:

$$z_k = (\cos \theta_k)x_k + v_k, \qquad \theta_k = 1 + \frac{k}{120} \text{ rad}$$

Using $Q = 4$, $R = 1$, and $P_0^- = 100$, run error covariance analyses for each measurement model for $k = 0, 1, 2, \ldots, 200$. Plot the estimation error variance for the scalar state x against the time index k for each case. Explain the difference seen between the two plots, particularly as the recursive process approaches and passes $k \approx 70$.

REFERENCES CITED IN CHAPTER 5

1. R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Trans. ASME—J. Basic Engr.,* 35–45 (March 1960).

2. J. J. D'Azzo and C. H. Houpis, *Linear Control System Analysis and Design,* 4th ed., New York: McGraw-Hill, 1995.
3. R. C. Dorf and R. H. Bishop, *Modern Control Systems,* 7th ed., Reading, MA: Addison-Wesley, 1995.
4. T. A. Stansell, Jr., "The Many Faces of Transit," *Navigation,* Journal of the Institute of Navigation, *25:*1, 55–70 (Spring 1978).
5. P. Henrici, *Discrete Variable Methods in Ordinary Differential Equations,* New York: Wiley, 1962.
6. C. R. Wylie, Jr., *Advanced Engineering Mathematics,* 3rd ed., New York: McGraw-Hill, 1966, pp. 108–117.
7. R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering,* 2nd ed., New York: Wiley, 1992.
8. C. F. van Loan, "Computing Integrals Involving the Matrix Exponential," *IEEE Trans. Automatic Control, AC-23:* 3, 395–404 (June 1978).
9. K. S. Shanmugan and A. M. Breipohl, *Random Signals: Detection, Estimation, and Data Analysis,* New York: Wiley, 1988.
10. J. M. Mendel, *Optimal Seismic Deconvolution,* New York: Academic Press, 1983.
11. M. B. Priestly,, *Spectral Analysis and Time Series,* New York: Academic Press, 1981.
12. G. H. Golub and C. F. van Loan, *Matrix Computations,* 2nd ed., Baltimore, MD: The Johns Hopkins University Press, 1989, pp. 141–142.
13. A. H. Jazwinski, *Stochastic Processes and Filtering Theory,* New York: Academic Press, 1970.
14. B. E. Bona and R. J. Smay, "Optimum Reset of Ship's Inertial Navigation System," *IEEE Trans. Aerospace Electr. Syst., AES-2:* 4, 409–414 (July 1966).
15. G. R. Pitman (ed.), *Inertial Guidance,* New York: Wiley, 1962.
16. J. C. Pinson, "Inertial Guidance for Cruise Vehicles," in C. T. Leondes (ed.), *Guidance and Control of Aerospace Vehicles,* New York: McGraw-Hill, 1963.
17. J. S. Meditch, *Stochastic Optimal Linear Estimation and Control,* New York: McGraw-Hill, 1969.
18. "Change No. 1 to RTCA/DO-208," RTCA paper no. 479-93/TMC-106, RTCA, Inc., Washington, DC, Sept. 21, 1993.

Additional References

19. A. Gelb (ed.), *Applied Optimal Estimation,* Cambridge, MA: MIT Press, 1974.
20. P. S. Maybeck, *Stochastic Models, Estimation and Control* (Vol. 1), New York: Academic Press, 1979.
21. A. P. Sage and J. L. Melsa, *Estimation Theory with Applications to Communications and Control,* New York: McGraw-Hill, 1971.
22. S. M. Bozic, *Digital and Kalman Filtering,* London: E. Arnold, Publisher, 1979.
23. B. D. O. Anderson and J. B. Moore, *Optimal Filtering,* Englewood Cliffs, NJ: Prentice-Hall, 1979.
24. C. T. Leondes (ed.), *Theory and Application of Kalman Filtering,* North Atlantic Treaty Organization AGARD rept. no. 139 (Feb. 1970).
25. H. W. Worenson, *Parameter Estimation,* New York: Marcel Dekker, 1980.
26. H. W. Sorenson, "Kalman Filtering Techniques," in *Advances in Control Systems* (Vol. 3), C. T. Leondes (ed.), New York: Academic Press, 1966, pp. 219–289.
27. M. Aoki, *State Space Modeling of Time Series,* Berlin: Springer-Verlag, 1987.
28. R. F. Stengel, *Stochastic Optimal Control–Theory and Application,* New York: Wiley, 1986.
29. M. S. Grewal and A. P. Andrews, *Kalman Filtering Theory and Practice,* Englewood Cliffs, NJ: Prentice-Hall, 1993.
30. G. Minkler and J. Minkler, *Theory and Application of Kalman Filtering,* Palm Bay, FL: Magellan Book Co., 1993.

6

Prediction, Applications, and More Basics on Discrete Kalman Filtering

In the period immediately following Kalman's original work, many extensions and variations were developed that enhanced the usefulness of the technique in applied work. This chapter continues the subject of discrete Kalman filtering with some of the more important extensions and related topics. Also, a limited number of applications are presented to illustrate the versatility of Kalman filtering. The treatment of applications here must be brief. More applications are given in Chapters 9, 10, and 11. There also exist a wealth of application papers in various journals and conference proceedings over the past 35 years. In particular, many papers dealing with navigation and trajectory determination will be found in *Navigation* (*Journal of the Institute of Navigation*), *IEEE Transactions on Aerospace and Electronic Systems,* and *IEEE Transactions on Automatic Control.* Also, Sorenson (1) gives an especially valuable collection of applied papers in his IEEE Press book.

6.1 PREDICTION

Extension of Kalman filtering to prediction is relatively easy. We first note that the projection operation in the filter loop shown in Fig. 5.8 is, in fact, one-step prediction. This was rationalized on the basis of the white-noise assumption for the $\mathbf{w}_k$ sequence in the process model (Eq. 5.5.1). We can use the same identical argument for projecting (i.e., predicting) N steps ahead of the measurement rather than just one step. The obvious equations for N-step prediction are then

$$\hat{\mathbf{x}}(k+N|k) = \boldsymbol{\phi}(k+N, k)\,\hat{\mathbf{x}}(k|k) \tag{6.1.1}$$

$$\mathbf{P}(k+N|k) = \boldsymbol{\phi}(k+N, k)\,\mathbf{P}(k|k)\boldsymbol{\phi}^T(k+N, k) + \mathbf{Q}(k+N, k) \tag{6.1.2}$$

where

$\hat{\mathbf{x}}(k|k)$ = updated filter estimate at time t_k

$\hat{\mathbf{x}}(k+N|k)$ = predictive estimate of $\mathbf{x}$ at time t_{k+N} given all the measurements through t_k

$\mathbf{P}(k|k)$ = error covariance associated with the filter estimate $\hat{\mathbf{x}}(k|k)$

$\mathbf{P}(k+N|k)$ = error covariance associated with the predictive estimate $\hat{\mathbf{x}}(k+N|k)$

$\boldsymbol{\phi}(k+N, k)$ = transition matrix from step k to $k+N$

$\mathbf{Q}(k+N, k)$ = covariance of the cumulative effect of white-noise inputs from step k to step $k+N$

Note that a more explicit notation is required here in order to distinguish between the end of the measurement stream (k) and the point of estimation ($k+N$). (These were the same in the filter problem, and thus a shortened subscript notation could be used without ambiguity.)

There are two types of prediction problems that we will consider:

Case 1: Case 1 is where N is fixed and k evolves in integer steps in time just as in the filter problem. In this case, the predictor is just an appendage that we add to the usual filter loop. This is shown in Fig. 6.1. In off-line analysis work, the $\mathbf{P}(k+N|k)$ matrix is of primary interest. The terms along the major diagonal of $\mathbf{P}(k+N|k)$ give a measure of the quality of the predictive state estimate. On the other hand, in on-line prediction it is $\hat{\mathbf{x}}(k+N|k)$ that is of primary interest. Note that it is not necessary to compute $\mathbf{P}(k+N|k)$ to get

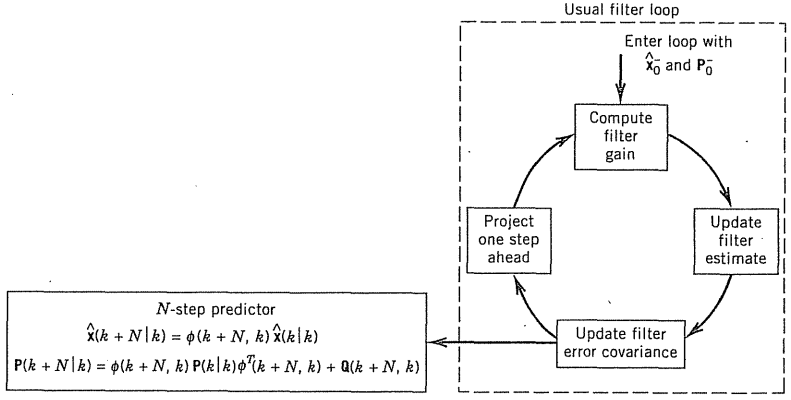


Figure 6.1 *N*-step prediction.

$\hat{\mathbf{x}}(k + N|k)$. Also, Case 1 will be recognized as the discrete recursive version of the Wiener prediction problem that was discussed in Chapter 4.

Case 2: Case 2 is where we fix k and then compute $\hat{\mathbf{x}}(k + N|k)$ and its error covariance for ever-increasing prediction times, that is, $N = 1, 2, 3, \ldots$, etc. The error covariance is of special interest here, because it tells us how the predictive estimate degrades as we reach out further and further into the future. We will now consider an example that illustrates this kind of prediction problem.

EXAMPLE 6.1

It was mentioned previously in Problem 5.4 that the timing on the GPS ranging signals from each of the satellites is dithered randomly in order to limit the civil user's horizontal accuracy to 100 m 2drms. This intentional degradation of the GPS signal is known as selective availability, or just SA for short. In a special form of GPS called differential GPS (15), the user is supplied with range corrections for all satellites as observed by a nearby monitoring station. These corrections include the effects of SA that usually predominate over all other sources of error. The range (and range rate) corrections cannot be transmitted continuously, though, because the amount of data to be transmitted is fairly large relative to the bit rate of the message. Therefore, the corrections are only updated in coarse steps, and the user must extrapolate this information in the interim between updates. It is, of course, of much interest in this application to see how the accuracy of the extrapolation degrades in between updates.

We will now consider an idealized situation where only the SA-induced range errors will be included in the analysis. Furthermore, we will say that at the update time (call it $t = 0$), the user receives perfect range and range-rate corrections for a particular satellite. The user must then extrapolate (predict) the range for a few tens of seconds until a fresh update is received. A precise statistical description of the random process for the SA range dithering has not been made public, but there is some empirical evidence that it can be approximated as a second-order Gauss-Markov process with a power spectral density (PSD) of the form (16):

$$\text{PSD} = S(\omega) = \frac{c}{\omega^4 + \omega_0^4} \text{ m}^2/(\text{rad/sec}) \tag{6.1.3}$$

where

c = constant determined by the level of dithering
$\omega_0 = .012$ rad/sec

As a matter of convenience, the amplitude of the SA dithering is usually expressed in terms of range rather than time (i.e., range = velocity of light × time). Suppose we choose the constant c to correspond to an rms amplitude of 30 m (17). Then, using the methods given in Chapter 3 (Table 3.1), we find that

$$\frac{1}{2\pi j}\int_{-j\infty}^{j\infty} \frac{c}{s^4 + \omega_0^4}\, ds = \frac{c}{2\sqrt{2}\,\omega_0^3} = (30 \text{ m})^2$$

or

$$c = .0043987 \text{ m}^2 \text{ (rad/sec)}^3 \tag{6.1.4}$$

Next, we can use the modeling methods discussed in Section 5.3 to develop a continuous state model. The result is (using range and range rate as the state variables)

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & -\sqrt{2}\omega_0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \sqrt{c} \end{bmatrix} u(t) \tag{6.1.5}$$

where

$u(t)$ = white noise with unity PSD

We will consider the observable to be range, so the discrete measurement equation is

$$z_k = [1 \quad 0] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_k + v_k \tag{6.1.6}$$

where the variance of v_k is R_k. The model is now complete with the specification of the step size Δt, R_k, and the initial conditions.

To find the mean-square error in prediction, one can initialize $\mathbf{P}(k|k)$ at zero and then repeatedly calculate $\mathbf{P}(k + N|k)$ as indicated in Fig. 6.1 for $N = 1, 2, 3$, etc. However, there is an easier way if filter error-covariance software is available. This being the case, we simply initialize $\mathbf{P}(0|0)$ at zero and then run the covariance program an appropriate number of steps with R_k set at an abnormally large value (e.g., 1.0e20). This is the equivalent of telling the Kalman filter that the measurement information is worthless. This, in turn, makes the $\mathbf{P}$ update trivial at each step, and $\mathbf{P}_k$ just propagates ahead in accordance with the $\mathbf{P}_{k+1} = \boldsymbol{\Phi}_k \mathbf{P}_k \boldsymbol{\Phi}_k^T + \mathbf{Q}_k$ equation. The result for a step size of 10 sec and 30 steps is shown in Fig. 6.2. Note that the rms error approaches 30 m as the predictive time becomes large. This is as expected. As we reach out further into the future, the correlation between the present and the future becomes smaller and smaller, and the optimal estimate approaches the mean of the process (zero in this case). The rms estimation error then is just the rms value of the process itself. Also note that if the differential corrections are refreshed every 50 sec, the prediction error can be held to about 10 m, at worst, at the end of the prediction interval. ■

There are a number of interesting variations that can be made on the "type 22" prediction problem illustrated in Example 6.1. One such variation is to carry the problem a bit further and compare the optimal predictor with a particular

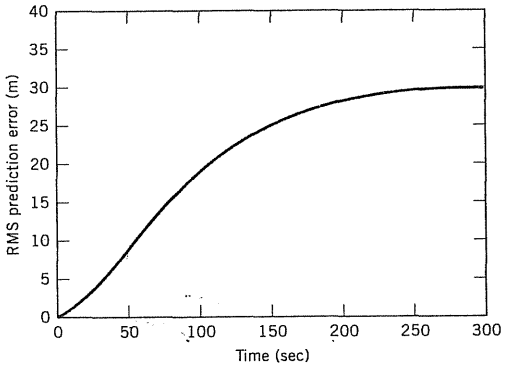


Figure 6.2 Prediction error—SA example.

suboptimal predictor where the prediction is accomplished by simply projecting ahead with a constant rate. This comparison is made in Section 6.7 as an example of suboptimal filter analysis. Another variation is to relax the assumption of beginning the prediction with perfect estimates of range and range rate. An example of this is given in Problem 6.1.

6.2 ALTERNATIVE FORM OF THE DISCRETE KALMAN FILTER

The Kalman filter equations given in Chapter 5 can be algebraically manipulated into a variety of forms (2, 3, 4). An alternative form that is especially useful will now be presented. We begin with the expression for updating the error covariance, Eq. (5.5.22), and we temporarily omit the subscripts to save writing.

$$\mathbf{P} = (\mathbf{I} - \mathbf{KH})\mathbf{P}^- \tag{6.2.1}$$

Recall that the Kalman gain is given by Eq. (5.5.17).

$$\mathbf{K} = \mathbf{P}^-\mathbf{H}^T(\mathbf{H}\mathbf{P}^-\mathbf{H}^T + \mathbf{R})^{-1} \tag{6.2.2}$$

Substituting Eq. (6.2.2) into (6.2.1) yields

$$\mathbf{P} = \mathbf{P}^- - \mathbf{P}^-\mathbf{H}^T(\mathbf{H}\mathbf{P}^-\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H}\mathbf{P}^- \tag{6.2.3}$$

We now wish to show that if the inverses of $\mathbf{P}$, $\mathbf{P}^-$, and $\mathbf{R}$ exist, $\mathbf{P}^{-1}$ can be written as

$$\mathbf{P}^{-1} = (\mathbf{P}^-)^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} \tag{6.2.4}$$

Justification of Eq. (6.2.4) is straightforward. We simply form the product of the right sides of Eqs. (6.2.3) and (6.2.4) and show that this reduces to the identity matrix. Proceeding as indicated, we obtain

$$\begin{aligned}
&[\mathbf{P}^- - \mathbf{P}^-\mathbf{H}^T(\mathbf{H}\mathbf{P}^-\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H}\mathbf{P}^-][(\mathbf{P}^-)^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}] \\
&= \mathbf{I} - \mathbf{P}^-\mathbf{H}^T[(\mathbf{H}\mathbf{P}^-\mathbf{H}^T + \mathbf{R})^{-1} - \mathbf{R}^{-1} + (\mathbf{H}\mathbf{P}^-\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H}\mathbf{P}^-\mathbf{H}^T\mathbf{R}^{-1}]\mathbf{H} \\
&= \mathbf{I} - \mathbf{P}^-\mathbf{H}^T[(\mathbf{H}\mathbf{P}^-\mathbf{H}^T + \mathbf{R})^{-1}(\mathbf{I} + \mathbf{H}\mathbf{P}^-\mathbf{H}^T\mathbf{R}^{-1}) - \mathbf{R}^{-1}]\mathbf{H} \\
&= \mathbf{I} - \mathbf{P}^-\mathbf{H}^T[\mathbf{R}^{-1} - \mathbf{R}^{-1}]\mathbf{H} \\
&= \mathbf{I}
\end{aligned}$$

An alternative expression for the Kalman gain may also be derived. Beginning with Eq. (6.2.2), we have

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

Insertion of $\mathbf{P}\mathbf{P}^{-1}$ and $\mathbf{R}^{-1}\mathbf{R}$ will not alter the gain. Thus, $\mathbf{K}$ can be written as

$$\begin{aligned}
\mathbf{K} &= \mathbf{P}\mathbf{P}^{-1}\mathbf{P}^-\mathbf{H}^T\mathbf{R}^{-1}\mathbf{R}(\mathbf{H}\mathbf{P}^-\mathbf{H}^T + \mathbf{R})^{-1} \\
&= \mathbf{P}\mathbf{P}^{-1}\mathbf{P}^-\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{H}\mathbf{P}^-\mathbf{H}^T\mathbf{R}^{-1} + \mathbf{I})^{-1}
\end{aligned}$$

We now use Eq. (6.2.4) for $\mathbf{P}^{-1}$ and obtain

$$\begin{aligned}
\mathbf{K} &= \mathbf{P}[(\mathbf{P}^-)^{-1} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}]\mathbf{P}^-\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{H}\mathbf{P}^-\mathbf{H}^T\mathbf{R}^{-1} + \mathbf{I})^{-1} \\
&= \mathbf{P}(\mathbf{I} + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{P}^-)\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{H}\mathbf{P}^-\mathbf{H}^T\mathbf{R}^{-1} + \mathbf{I})^{-1} \\
&= \mathbf{P}\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{I} + \mathbf{H}\mathbf{P}^-\mathbf{H}^T\mathbf{R}^{-1})(\mathbf{H}\mathbf{P}^-\mathbf{H}^T\mathbf{R}^{-1} + \mathbf{I})^{-1} \\
&= \mathbf{P}\mathbf{H}^T\mathbf{R}^{-1}
\end{aligned} \tag{6.2.5}$$

The main results have now been derived, and Eqs. (6.2.4) and (6.2.5) may be rewritten with the subscripts reinserted:

$$\mathbf{P}_k^{-1} = (\mathbf{P}_k^-)^{-1} + \mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k \tag{6.2.6}$$

$$\mathbf{K}_k = \mathbf{P}_k\mathbf{H}_k^T\mathbf{R}_k^{-1} \tag{6.2.7}$$

Note that the updated error covariance can be computed without first finding the gain. Also, the expression for gain now involves $\mathbf{P}_k$; therefore, if Eq. (6.2.7) is to be used, $\mathbf{K}_k$ must be computed *after* the $\mathbf{P}_k$ computation. Thus, the order in which the $\mathbf{P}_k$ and $\mathbf{K}_k$ computations appear in the recursive algorithm is reversed from that presented in Chapter 5. The alternative Kalman filter algorithm just derived is summarized in Fig. 6.3.

Note from Fig. 6.3 that two ($n \times n$) matrix inversions are required for each recursive loop. If the order of the state vector is large, this leads to obvious computational problems. Nevertheless, the alternative algorithm has some useful applications. One of these will now be presented as an example.

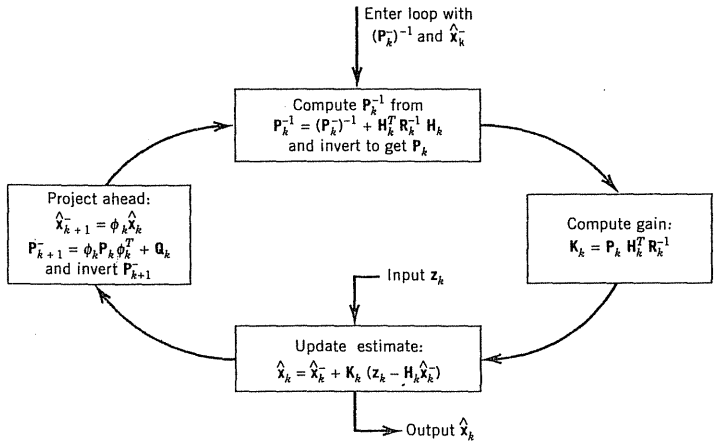


Figure 6.3 Alternative Kalman filter recursive loop.

EXAMPLE 6.2

Suppose we wish to estimate a random constant based on a sequence of independent noisy measurements of the constant. We can think of the constant as being a deterministic random process that satisfies the differential equation

$$\dot{x} = 0 \tag{6.2.8}$$

Thus, ϕ_k and Q_k are 1 and 0, respectively. Let us also speculate that very little is known about the process initially. The constant is equally likely to be positive or negative, and its magnitude could be quite large. It might be thought of as a random variable with a flat probability density function extending from $-\infty$ to $+\infty$, or more realistically, a normal zero-mean random variable with a very large variance. This being the case, the a priori estimate and associated error covariance should be

$$\hat{x}_0^- = 0 \tag{6.2.9}$$

$$P_0^- = \infty \tag{6.2.10}$$

However, this is not permitted in the usual Kalman filter algorithm (Fig. 5.8), because it leads to the indeterminant form ∞/∞ in the gain expression. The alternative algorithm will accommodate this situation, though, because $(P_0^-)^{-1}$ rather than P_0^- appears in the first step.

Time is of no consequence in this example because we are estimating a constant. So let us assume we have N independent noisy measurements of x, each made at $t = 0$ and having an error variance of σ^2. The measurement model is then

$$\begin{bmatrix} z_1 \\ z_2 \\ \vdots \\ z_N \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} [x] + \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_N \end{bmatrix} \tag{6.2.11}$$

and $\mathbf{R}_k$ is the $(N \times N)$ diagonal matrix

$$\mathbf{R}_k = \begin{bmatrix} \sigma^2 & 0 & \cdots \\ 0 & \sigma & \\ \vdots & & \ddots \\ & & \sigma^2 \end{bmatrix} = \sigma^2 \mathbf{I} \tag{6.2.12}$$

Proceeding with the first step of the alternative algorithm yields

$$\begin{aligned} P_0^{-1} &= (P_0^-)^{-1} + \mathbf{H}^T \mathbf{R}_0^{-1} \mathbf{H}_0 \\ &= (\infty)^{-1} + [1 \quad 1 \quad \cdots \quad 1] \frac{1}{\sigma^2} \mathbf{I} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \end{aligned} \tag{6.2.13}$$

or

$$P_0 = \frac{\sigma^2}{N} \tag{6.2.14}$$

Next, the gain is computed as

$$\begin{aligned} \mathbf{K}_0 &= P_0 \mathbf{H}_0^T \mathbf{R}_0^{-1} \\ &= \frac{\sigma^2}{N} [1 \quad 1 \quad \cdots] \begin{bmatrix} \frac{1}{\sigma^2} & 0 & \cdots \\ 0 & \frac{1}{\sigma^2} & \\ \vdots & & \ddots \end{bmatrix} \\ &= \begin{bmatrix} \frac{1}{N} & \frac{1}{N} & \cdots & \frac{1}{N} \end{bmatrix} \end{aligned} \tag{6.2.15}$$

Finally, the estimate is given by

$$\begin{aligned} \hat{x}_0 &= \hat{x}_0^- + \mathbf{K}_0 (\mathbf{z} - \mathbf{H}_0 \hat{x}_0^-) \\ &= \mathbf{K}_0 \mathbf{z} = \frac{z_1}{N} + \frac{z_2}{N} + \cdots + \frac{z_N}{N} \end{aligned} \tag{6.2.16}$$

The final result is no surprise; it is exactly the result one would expect from elementary statistics. The main point of this example is this: The alternative algorithm provides a means of starting the Kalman filter with "infinite uncertainty" if the physical situation under consideration so dictates. ■

6.3 PROCESSING THE MEASUREMENT VECTOR ONE COMPONENT AT A TIME

We now have two different Kalman filter algorithms as summarized in Figs. 5.8 and 6.3. They are, of course, algebraically equivalent and produce identical estimates (with perfect arithmetic). The choice as to which should be used in a particular application is a matter of computational convenience. Both algorithms involve matrix inverse operations, and these may lead to difficulties. When using the alternative algorithm of Fig. 6.3, there is no reasonable way to avoid two $(n \times n)$ matrix inversions with each recursive cycle. If the dimension of the state vector n is large, this is, at best, awkward computationally. On the other hand, the matrix inverse that appears in the regular algorithm given in Fig. 5.8 is the same order as the measurement vector. Since this is often less than the order of the state vector, it is usually the preferred algorithm. Furthermore, if the measurement errors at time t_k are uncorrelated, the inverse operation can be eliminated entirely by processing the scalar measurements one at a time.* This will now be shown.

We begin with the expression for the updated error covariance, Eq. (6.2.6):

$$\mathbf{P}_k^{-1} = (\mathbf{P}_k^-)^{-1} + \left[\mathbf{H}_k^{aT} \,\middle|\, \mathbf{H}_k^{bT} \,\middle|\, \cdots\right] \left[\begin{array}{c|c|c} (\mathbf{R}_k^a)^{-1} & \mathbf{0} & \mathbf{0} \\ \hline \mathbf{0} & (\mathbf{R}_k^b)^{-1} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{0} & \ddots \end{array}\right] \left[\begin{array}{c} \mathbf{H}_k^a \\ \hline \mathbf{H}_k^b \\ \hline \vdots \end{array}\right] \tag{6.3.1}$$

The second term in Eq. (6.3.1) is intentionally written in partitioned form and $\mathbf{R}_k$ is assumed to be at least block diagonal. Physically, this means that the measurements available at t_k can be grouped together such that the measurement errors among the a, b, . . . blocks are uncorrelated. This is often the case when redundant measurements come from different instruments. We next expand the partitions of Eq. (6.3.1) to get

* Processing the vector measurements one component at a time is sometimes referred to as sequential processing. The authors, however, have found that there is some confusion in the meaning of the words sequential and recursive. For this reason, processing the measurements sequentially one at a time *at a given point in time* will be referred to here simply as one-at-a-time processing. The term recursive will be reserved to mean the step-by-step evolution of the measurement processing with time.

$$\mathbf{P}_k^{-1} = \underbrace{\underbrace{\underbrace{(\mathbf{P}_k^-)^{-1} + \mathbf{H}_k^{aT}(\mathbf{R}_k^a)^{-1}\mathbf{H}_k^a}_{\mathbf{P}_k^{-1}\text{ after assimilating block } a \text{ measurements}} + \mathbf{H}_k^{bT}(\mathbf{R}_k^b)^{-1}\mathbf{H}_k^b}_{\mathbf{P}_k^{-1}\text{ after assimilating both block } a \text{ and } b \text{ measurements}} + \cdots}_{\text{and so forth}} \tag{6.3.2}$$

Note that the sum of the first two terms is just the $\mathbf{P}_k^{-1}$ one would obtain after assimilating the "block a" measurement just as if no further measurements were available. The Kalman gain associated with this block of measurements may now be used to update the state estimate accordingly. Now think of making a trivial projection ahead through zero time. The a posteriori $\mathbf{P}$ then becomes the a priori $\mathbf{P}$ for the next step. When this is added to the b term of Eq. (6.3.2), we have the updated $\mathbf{P}_k^{-1}$ after assimilating the second block of data. This can now be repeated until all blocks are processed. The final estimate and associated error is then the same as would be obtained if all the measurements at t_k had been processed simultaneously. Thus, the designer has some flexibility in the design of the system software. The available measurements at any particular time may be processed either in blocks, one block at a time, or all at once, as best suits the situation at hand. One-at-a-time measurement processing is illustrated in the timing diagram of Fig. 6.4. Note that once we have established the validity of one-at-a-time processing, it makes no difference whether we use the "usual" update formula given in Chapter 5 (see Fig. 5.8) or the alternative formula given in Section 6.2. The end results are the same (within the limits of computational arithmetic).

The concept of processing the measurements one block at a time leads to an interesting physical interpretation of $\mathbf{P}$ inverse. With reference to Eq. (6.3.2), think of $(\mathbf{P}_k^-)^{-1}$ as a measure of the information content of the a priori estimate, that is, before the new measurement information is assimilated into the estimate. For simplicity, begin with $(\mathbf{P}_k^-)^{-1} = 0$. This corresponds to infinite uncertainty, or zero information. Then, as each measurement block is processed, we add an amount $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$ to the previous information, until finally the total information is the sum indicated by Eq. (6.3.2). The term "add" is appropriate here because $\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}$ is always positive definite. For the heuristic reasons just noted, $\mathbf{P}$ inverse is often referred to as the *information matrix*. This concept is developed further in Chapter 9 in the discussion of decentralized filters (Section 9.6).

One-at-a-time processing is also useful from a system organization viewpoint. Often the system must have the flexibility to accommodate a variety of measurement combinations at each update point. By block processing, the system may be programmed to cycle through all possible measurement blocks one at a time, processing those that are available and skipping those that are not. Simultaneous processing requires a somewhat more complicated system organization whereby the system must be able to form appropriate $\mathbf{H}_k$ and $\mathbf{R}_k$ matrices for all possible combinations of measurements, and it must be prepared to do the corresponding matrix operations with various dimensionality.

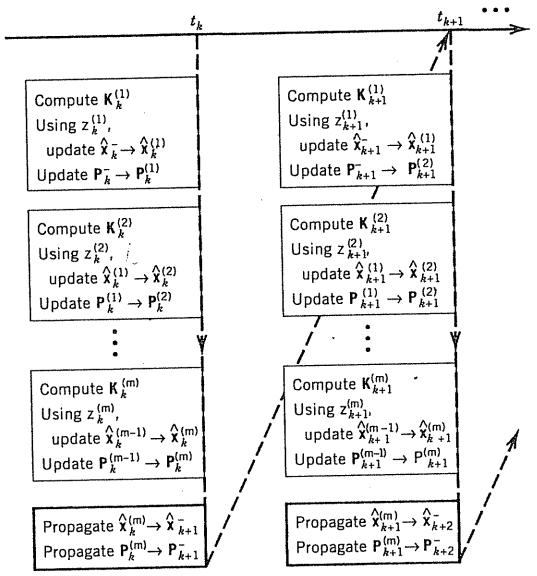


Figure 6.4 Timing diagram for one-at-a-time measurement processing (m = number of elements in the measurement vector $\mathbf{z}_k$).

There are bound to be some applications where the measurement errors are all mutually correlated. The $\mathbf{R}_k$ matrix is then "full." If this is the case, and one-at-a-time processing is desirable, linear combinations of the measurements may be formed in such a way as to form a new set of measurements whose errors are uncorrelated. One technique for accomplishing this is known as the Gram–Schmidt orthogonalization procedure (4). This procedure is straightforward and an exercise is included to demonstrate its application to the problem of decoupling the measurement errors (see Problem 6.2). Note that the intuitive procedure demonstrated in Problem 6.2 is closely related to Cholesky factorization. This is discussed in detail in Section 5.4.

6.4 POWER SYSTEM RELAYING APPLICATION

New applications of Kalman filtering keep appearing regularly, and many of these are now outside the original application area of navigation. One such application has to do with power system relaying (9, 10). When a fault (short) occurs on a three-phase transmission line, it is desirable to sense the problem promptly and take appropriate relaying action to protect the remainder of the system. The hierarchy of decisions that must be made as to which relays should trip (and where) is relatively complicated. It suffices to say here that it is desirable to determine the distance to the fault as soon as possible; and, in order to do this, the steady-state postfault currents and voltages must be estimated. Transient components are superimposed on the steady-state signals immediately after the fault, so that the transients become the corrupting noise in the problem. Normally, these transients are not considered as random noise. However, to model them otherwise complicates the problem immensely because of the many variables involved. So the basic problem is to estimate the steady-state components of the sending-end voltages and current in the presence of the transient (noise) components. It is assumed that digital samples of the various phase voltages and currents are available for processing at a reasonably fast rate, say, 64 samples per cycle of the 60-Hz signal.

Girgis and Brown (10) made an extensive simulation study of the transients accompanying various types of faults for various lengths of line, and so forth. They concluded that the transients could be approximated as nonstationary random processes, and they developed models accordingly that would fit the required format of a Kalman filter. The simulation study indicated that the voltage transients consisted primarily of high-frequency components that decayed exponentially with time. Thus, it seemed reasonable to model the time samples of the process as a white sequence with an exponentially decaying variance. The current transients, however, showed (on the average) a sizable long-time-constant exponential component in addition to the high-frequency components. (Power engineers sometimes refer to this as the "dc offset.") Thus, the model chosen for the current noise was an exponential process with random initial amplitude plus a white sequence with exponentially decaying variance. The signal process to be estimated for both current and voltage was a sine wave with random amplitude and phase. This is readily modeled as a two-element vector, where the state variables are the coefficients of the sine and cosine components of the wave (see Section 5.2). The final models for the currents and voltages may be summarized as follows:

Voltage Model (same for each phase)

1. State equations:

$$\begin{bmatrix} x1_{k+1} \\ x2_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x1_k \\ x2_k \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{6.4.1}$$

2. Measurement equation:

$$z_k = [\cos \omega_0 k\,\Delta t \quad -\sin \omega_0 k\,\Delta t] \begin{bmatrix} x1_k \\ x2_k \end{bmatrix} + v_k \tag{6.4.2}$$

3. Initial conditions:

$$\mathbf{P}_0^- = \begin{bmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_v^2 \end{bmatrix} \tag{6.4.3}$$

$$\hat{\mathbf{x}}_0^- = \text{measured value of } \mathbf{x} \text{ just prior to the fault} \tag{6.4.4}$$

The σ_v^2 parameter was determined by simulation. It is fixed and not determined on-line. On the other hand, the initial estimates of $\mathbf{x}$ are assumed to be determined on-line. There is no reason to waste the available measurement information just prior to the fault. It should be obvious from Eq. (6.4.1) that $\mathbf{Q}_k = 0$ for the voltage model. Also, as mentioned previously, $\mathbf{R}_k$ is assumed to decay exponentially with k, and the exponential parameters are predetermined by simulation or experimental data.

Current Model (same for all phases)

1. State equations:

$$\begin{bmatrix} x1_{k+1} \\ x2_{k+1} \\ x3_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & e^{-\beta \Delta t} \end{bmatrix} \begin{bmatrix} x1_k \\ x2_k \\ x3_k \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ w_k \end{bmatrix} \tag{6.4.5}$$

2. Measurement equation:

$$z_k = [\cos \omega_0 k \, \Delta t \quad -\sin \omega_0 k \, \Delta t \quad 1] \begin{bmatrix} x1_k \\ x2_k \\ x3_k \end{bmatrix} + v_k \tag{6.4.6}$$

3. Initial conditions:

$$\mathbf{P}_0^- = \begin{bmatrix} \sigma_i^2 & 0 & 0 \\ 0 & \sigma_i^2 & 0 \\ 0 & 0 & \sigma_i^2 \end{bmatrix} \tag{6.4.7}$$

$$\hat{\mathbf{x}}_0^- = \begin{bmatrix} x1^- \text{ (meas.)} \\ x2^0 \text{ (meas.)} \\ 0 \end{bmatrix} \tag{6.4.8}$$

Just as in the voltage model, σ_i^2 is determined off-line and the first two elements of $\hat{\mathbf{x}}_0^-$ are obtained from measurements just prior to the fault. The third element of $\hat{\mathbf{x}}_0^-$ (i.e., the exponential component) is assumed to be zero. The measurement error variance R_k is assumed to decay exponentially just as in the voltage model. The $\mathbf{Q}_k$ parameter is not zero, though, because the exponential component was observed in the simulation studies to have a small residual noise associated with it. This is accounted for in the model with $\mathbf{w}_k$, and thus the "33" element of $\mathbf{Q}_k$ is nonzero. In effect, $x3$ is modeled as a nonstationary process with a large random initial value, and then it relaxes to a Markov process with a relatively small rms value in the steady-state condition. This is an unusual model, but perfectly legitimate. Again, this illustrates the versatility of the Kalman filter to adapt to a wide variety of situations.

Figures 6.5 and 6.6 show the voltage and current estimates for a particular simulation of a line-to-ground fault located 90 miles from the sending end. The details of the simulation are not important here, because the results are all relative. Recall that $x1$ and $x2$ are the coefficients of the sine and cosine components

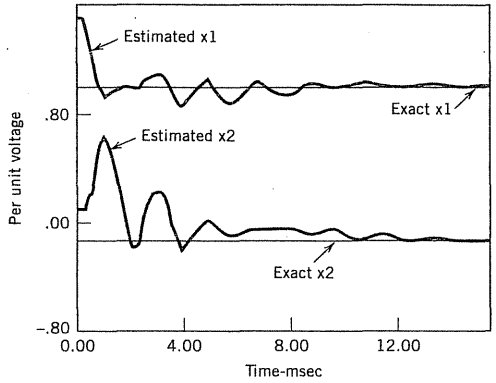


Figure 6.5 Kalman filter estimation of the postfault voltage states.

of the steady-state values, so they are constants. Note that the Kalman filter estimates converge reasonably well to the correct values after about 8 ms (half cycle at 60 Hz). Figure 6.7 shows the result of using the voltage and current estimates of this simulation to compute distances to the fault. A similar distance calculation was also made using current and voltages as determined by a discrete Fourier transform algorithm. Both the Fourier transform and Kalman filter results are shown in Fig. 6.7, and it is clear that the Kalman filter converges on the correct result faster than the other algorithm. This is as expected. The discrete Fourier transform approach does not account for the time-varying nature of the noise, nor does it allow for any a priori knowledge of the parameters being estimated. Of course, both algorithms converge to the correct result eventually. However, time is of the essence! Why accept inferior performance when optimal performance is readily available?

Improvements have been made on this basic scheme, since it was first presented in 1981 by Girgis and Brown (10). The refinements involve adaptive

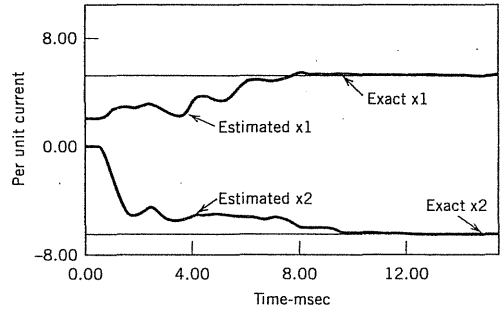


Figure 6.6 Kalman filter estimation of the postfault current states.

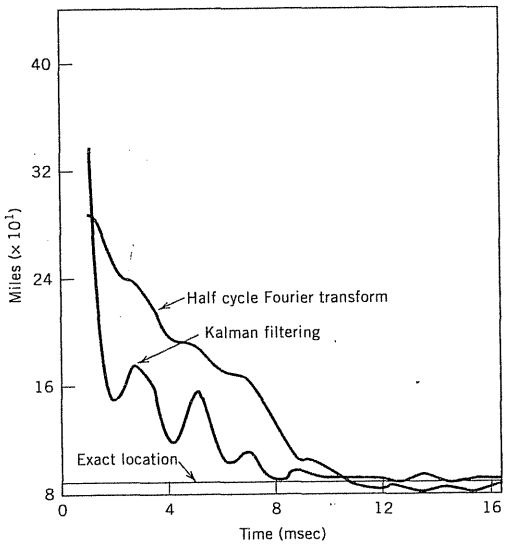


Figure 6.7 The computed distance to the fault using the Kalman filter algorithm and the discrete Fourier transform.

Kalman filtering, though, so further discussion of this application will be deferred until Chapter 9.

6.5 POWER SYSTEM HARMONICS DETERMINATION

A recent application of Kalman filtering in power systems has to do with the determination of the harmonic content of the 60-Hz voltage and current waveforms (18). Ideally, these waveforms should be pure sinusoids. However, transients induced by heavy loads being switched on and off and electronically controlled loads cause distortion of the waveforms. The amount of distortion is best described in terms of the harmonic content of the waveform (i.e., the Fourier series components). The harmonics can change with time, so they need to be monitored continuously. Of course, if the harmonic content changes with time, say, in a random way, the waveform will not be truly periodic in a strict sense. We will assume here that the waveforms that we are dealing with are at least quasiperiodic, and we will be interested in tracking a sort of "average level" of the various harmonics present, averaged over a few cycles of the 60-Hz fundamental.

The harmonic-process modeling for this application will follow closely the in-phase and quadrature state model that was discussed in Section 5.2. There is

one essential difference, though. Here we are looking for a filter that will operate in a steady-state condition. The random process model in this case must not be deterministic. That is, each of the in-phase and quadrature components must be allowed to random walk, and this is accomplished in the model by including a white-noise forcing function for each state variable. The effect of this is to deweight older measurement data, and this allows the filter gains to approach a nontrivial steady-state condition.

We will now look at a specific example. In the interest of simplicity, let us say that we are primarily interested in estimating just the fundamental, 3rd, and 5th harmonics. These are assumed to be the dominant components. For each frequency we will have in-phase and quadrature components as state variables as discussed in Section 5.2. Our 6-state continuous model is then

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \end{bmatrix} = [\mathbf{0}] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \end{bmatrix} \tag{6.5.1}$$

where

x_1, x_2 = in-phase and quadrature components of the fundamental
x_3, x_4 = in-phase and quadrature components of the 3rd harmonic
x_5, x_6 = in-phase and quadrature components of the 5th harmonic
$u_1, u_2, \ldots, u_6$ = independent Gaussian white-noise forcing functions

The corresponding discrete process model is then

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}_{k+1} = [\mathbf{I}] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix}_k + \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \end{bmatrix}_k \tag{6.5.2}$$

The $\boldsymbol{\phi}_k$ and $\mathbf{Q}_k$ parameters of the filter are then

$$\boldsymbol{\phi}_k = \mathbf{I} \quad (6 \times 6 \text{ identity matrix}) \tag{6.5.3}$$

$$\mathbf{Q}_k = \text{cov}[w_1 \quad w_2 \quad w_3 \quad w_4 \quad w_5 \quad w_6]_k^T \quad \text{(diagonal } 6 \times 6 \text{ matrix)} \tag{6.5.4}$$

The process model is now complete except for specifying numerical values for the elements along the diagonal of $\mathbf{Q}_k$ and the step size Δt.

The sampling rate in the Kalman filter should be at least as high as twice the highest frequency component (i.e., the Nyquist rate) in order for the filter to be competitive with FFT methods. For this example we will let the sampling

rate be 32 samples per cycle of the 60-Hz fundamental. Therefore, the update interval will be

$$\Delta t = (1/32)(1/60) = 1/1920 \text{ sec} \tag{6.5.5}$$

The measurement is scalar in this example, and it is given by the equation

$$z_k = \underbrace{x_1 \cos k\omega\Delta t - x_2 \sin k\omega\Delta t}_{\text{Fundamental}} + \underbrace{x_3 \cos 3k\omega\Delta t - x_4 \sin 3k\omega\Delta t}_{\text{3rd harmonic}} + \underbrace{x_5 \cos 5k\omega\Delta t - x_6 \sin 5k\omega\Delta t}_{\text{5th harmonic}} + v_k \tag{6.5.6}$$

where

$\omega = 2\pi \cdot 60$ rad/sec

v_k = measurement noise (white sequence)

It can be seen from Eq. (6.5.6) that the noise term v_k must include everything that has not been accounted for in the three harmonic components. Thus, if we think that there may be higher harmonics present, their effect must be lumped into v_k. This, in turn, will reflect into the numerical value assigned to R_k. This leads to some degree of suboptimality, but it makes the model manageable.* The $\mathbf{H}_k$ matrix is now obvious from Eq. (6.5.6), and it is

$$\mathbf{H}_k = [\cos k\omega\Delta t \quad -\sin k\omega\Delta t \quad \cos 3k\omega\Delta t \quad -\sin 3k\omega\Delta t \quad \cos 5k\omega\Delta t \quad -\sin 5k\omega\Delta t] \tag{6.5.7}$$

Note that the sampling rate was chosen to be an integer multiple of the fundamental frequency. Therefore, the elements of $\mathbf{H}_k$ will repeat themselves every 32 steps. This means that they can be precomputed and stored; they do not have to be computed on-line.

To demonstrate that the Kalman filter will track changes in the harmonic content of a waveform, a Monte Carlo measurement sequence was created with a jump change in the 3rd harmonic in the middle of the run. Specifically, z_k was generated with MATLAB in accordance with the following equation:

* Omission of higher-order harmonics in the state model would be a modeling error (if they were actually present), and this would degrade the filter estimates relative to what would be obtained if they had been correctly included in the model. The effect of mismodeling in this application is somewhat similar to aliasing, which can occur in FFT spectral analysis, but it is not exactly the same. For example, in the problem at hand, if there were a 32nd harmonic present, it would not be aliased down to a zero-frequency component, because we have not included a bias component in our state model. The unmodeled 32nd harmonic would, however, degrade the Kalman filter estimates of the fundamental, 3rd, and 5th in some complex sort of way. We will not elaborate further on the relative merits of Kalman-filter vs. FFT methods here. This is discussed in some detail in the Girgis reference (18).

For $k = 0, 1, 2, \ldots, 239$:

$$z_k = -1.0 \sin k\omega\Delta t - .1 \sin 3k\omega\Delta t - .05 \sin 5k\omega\Delta t + v_k$$

For $k = 240, 241, 242, \ldots, 479$:

$$z_k = -1.0 \sin k\omega\Delta t - .05 \sin 3k\omega\Delta t - .05 \sin 5k\omega\Delta t + v_k \tag{6.5.8}$$

where the coefficients are per unit (p.u.) values. The noise term v_k was a random Gaussian sequence with $\sigma = .03$ p.u.

A 6-state Kalman filter was designed for this example using the following constant parameters:

$$q_{11} = q_{22} = 1/3200 \text{ (p.u.)}^2$$
$$q_{33} = q_{44} = 1/320{,}000 \text{ (p.u.)}^2$$
$$q_{55} = q_{66} = 1/1{,}280{,}000 \text{ (p.u.)}^2$$
$$R_k = .0009 \text{ (p.u.)}^2 \tag{6.5.9}$$

In particular, the q_{44} term (which pertains to the quadrature 3rd harmonic) was set to correspond to an rms random walk of .01 p.u. in one 60-Hz cycle.

Figure 6.8 shows a plot of the filter's estimate of x_4 for a typical run spanning 15 cycles of the fundamental (i.e., .25 sec). The initial transient in this realization can be ignored, because we are mainly interested in the steady-state performance and the filter's ability to track changes in the harmonic content. It can be seen that the filter does a reasonably good job of tracking the change in

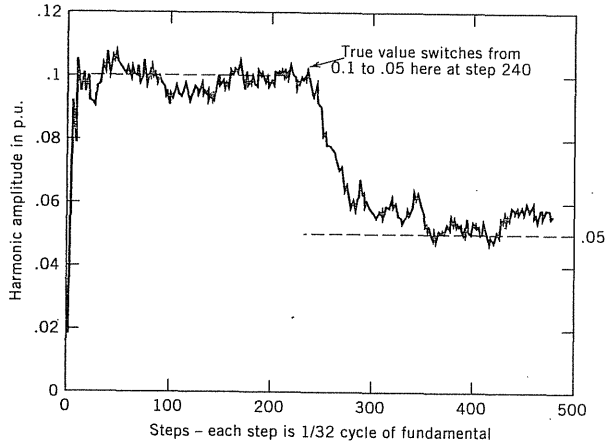


Figure 6.8 Filter's response to step change in 3rd harmonic in the middle of the run.

x_4 in the middle of the run. It required about 40 steps for the filter to readjust to the new value of .05 p.u. This corresponds to slightly more than one cycle of the 60-Hz fundamental. The response time is controlled, to a large extent, by the $\mathbf{Q}_k$ parameters. Making them larger speeds up the response, but at the expense of less smoothing of the measurement noise. The stochastic model was not designed to accommodate step changes in the state variables, but even so, the filter seems to be fairly forgiving of this. There was no attempt to optimize the model parameters in this example.

There are many interesting facets to the harmonic analysis problem that we will not be able to consider here. Many of these are discussed in the Girgis paper (18). Also, the periodic nature of the steady-state gains is the subject of Problem 6.5 at the end of this chapter. One final comment is in order, though, before leaving this application. At first glance, it might appear that a Kalman filter would require much more on-line computational effort than would be the case using FFT methods. This is not so, though. Remember, the Kalman filter will be operating in a steady-state condition. This means that the filter gains can be precomputed (as well as the $\mathbf{H}_k$ terms) and stored. Furthermore, the projection step is quite simple because the transition matrix is just $\mathbf{I}$. Also note that the $\mathbf{P}$ matrix is not needed in the on-line version, because the gains are precomputed and stored. When all of these things are considered, the on-line Kalman filter implementation is quite simple and only requires a modest computer effort.

6.6 DIVERGENCE PROBLEMS

Since the discrete Kalman filter is recursive, the looping can be continued indefinitely, in principle, at least. There are practical limits, though, and under certain conditions divergence problems can arise. We elaborate briefly on three common sources of difficulty.

Roundoff Errors

As with any numerical procedure, roundoff error can lead to problems as the number of steps becomes large. There is no one simple solution to this, and each case has to be examined on its own merits. Fortunately, if the system is observable and process noise drives each of the state variables, the Kalman filter has a degree of natural stability. In this case a stable, steady-state solution for the $\mathbf{P}$ matrix will normally exist, even if the process is nonstationary. If the $\mathbf{P}$ matrix is perturbed from its steady-state solution in such a way as not to lose positive definiteness, then it tends to return to the same steady-state solution. This is obviously helpful, provided $\mathbf{P}$ does not lose its positive definiteness. (See Section 6.9 for more on filter stability.)

Some techniques that have been found useful in preventing, or at least forestalling, roundoff error problems are:

1. Use high-precision arithmetic, especially in off-line analysis work.

2. If measurement data are sparse, beware of propagating the $\mathbf{P}$ matrix in many tiny steps between measurements. (This simplifies the transition matrix calculation but opens "Pandora's box" with regard to roundoff error.)
3. If possible, avoid deterministic (undriven) processes in the filter modeling. (*Example:* a random constant.) These usually lead to a situation where the $\mathbf{P}$ matrix approaches a semidefinite condition as the number of steps becomes large. A small error may then trip the $\mathbf{P}$ matrix into a non-positive-definite condition, and this can then lead to divergence. A good solution is to add (if necessary) small positive quantities to the major-diagonal terms of the $\mathbf{Q}$ matrix. This amounts to inserting a small amount of process noise to each of the states. This leads to a degree of suboptimality, but that is better than having the filter diverge!
4. Symmetrize the $\mathbf{P}$ and $\mathbf{P}^-$ matrices with each recursive step. This is very important. We know that a covariance matrix must be symmetric, so any asymmetry must be due to roundoff error. The asymmetry can grow if left unchecked. [The symmetry problem is automatically solved if, in programming the recursive equations, one assumes symmetry and uses only the upper (or lower) triangular part of the covariance matrix in all required matrix-multiply operations.]
5. Large uncertainty in the initial estimate can sometimes lead to numerical problems. For example, in a navigation situation, if we start the filter's $\mathbf{P}_0^-$ matrix with very large values along the major diagonal, and if we then follow this with a very precise measurement at $t = 0$, the $\mathbf{P}$ matrix must transition from a very large value to a value close to zero in one step. It can be seen from the $\mathbf{P}$-update equation

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^- \tag{6.6.1}$$

that this situation approximates the indeterminate form $0 \times \infty$. One should always be cautious in this kind of numerical situation. One possible solution is to make the elements of $\mathbf{P}_0^-$ artificially smaller and simply recognize that the filter will be suboptimal for the first few steps. Another possibility is to use one of the other $\mathbf{P}$-update formulas that has natural symmetry. One of these, which is sometimes called the *Joseph form* (19), is

$$\mathbf{P} = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T \tag{6.6.2}$$

This form has somewhat better numerical behavior than the simpler form given by Eq. (6.6.1) in unusual numerical situations.
6. In some on-line applications where the filter is expected to operate for long periods, perhaps with poor observability, it may be desirable to implement an alternative algorithm known as *U-D* factorization. This algorithm is mathematically equivalent to the regular and alternative Kalman filter algorithms given in Sections 5.5 and 6.2, but factors of $\mathbf{P}$ are propagated with each step rather than $\mathbf{P}$ itself. The *U-D* form of the

Kalman filter is more difficult to program than the usual form, but it has better numerical behavior (see Chapter 9).

Modeling Errors

Another type of divergence may arise because of inaccurate modeling of the process being estimated. This has nothing to do with numerical roundoff; it occurs simply because the designer (engineer) "told" the Kalman filter that the process behaved one way, whereas, in fact, it behaves another way. As a simple example, if you tell the filter that the process is a random constant (i.e., zero slope), and the actual process is a random ramp (nonzero slope), the filter will be continually trying to fit the wrong curve to the measurement data! This can also occur with nondeterministic as well as deterministic processes, as will now be demonstrated.

EXAMPLE 6.3

Consider a process that is actually random walk but is incorrectly modeled as a random constant. We have then (with numerical values inserted to correspond to a subsequent simulation):

(a) The "truth model":

$$\dot{x} = u(t), \qquad u(t) = \text{unity Gaussian white noise, and } \mathrm{Var}[x(0)] = 1$$
$$z_k = x_k + v_k, \qquad \text{measurement samples at } t = 0, 1, 2, \ldots$$
$$\text{and } \mathrm{Var}(v_k) = .1$$

(b) Incorrect Kalman filter model:

$$x = \text{constant}, \qquad \text{where } x \sim N(0, 1)$$
$$z_k = x_k + v_k \qquad \text{(same as for truth model)}$$

The Kalman filter parameters for the incorrect model (b) are: $\phi_k = 1$, $Q_k = 0$, $H_k = 1$, $R_k = .1$, $\hat{x}_0 = 0$, and $P_0^- = 1$. For the truth model the parameters are the same except that $Q_k = 1$, rather than zero.

The random walk process (a) was simulated using Gaussian random numbers with zero mean and unity variance. The resulting sample process for 35 sec is shown in Fig. 6.9. A measurements sequence z_k of this sample process was also generated using another set of $N(0, 1)$ random numbers for v_k. This measurement sequence was first processed using the incorrect model (i.e., $Q_k = 0$), and again with the correct model (i.e., $Q_k = 1$). The results are shown along with the sample process in Fig. 6.9. In this case, the measurement noise is relatively small ($\sigma \approx .3$), and we note that the estimates of the correctly modeled filter follow the random walk quite well. On the other hand, the incorrectly modeled filter does very poorly after the first few steps. This is due to the filter's gain becoming less and less with each succeeding step. At the 35th step the gain

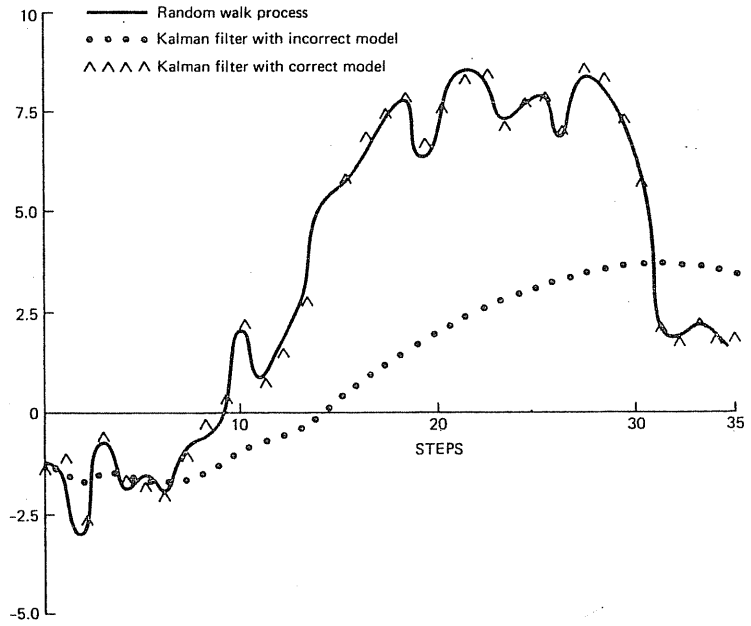


Figure 6.9 Simulation results for random walk example.

is almost two orders of magnitude less than at the beginning. Thus, the filter becomes very sluggish and will not follow the random walk. Had the simulation been allowed to go on further, it would have become even more sluggish. ▩

The moral to Example 6.3 is simply this. Any model that assumes the process, or any facet of the process, to be absolutely constant forever and ever is a risky model. In the physical world, very few things remain absolutely constant. Instrument biases, even though called "biases," have a way of slowly changing with time. Thus, most instruments need occasional recalibration. The obvious remedy for this type of divergence problem is always to insert some process noise into each of the state variables. Do this even at the risk of some degree of suboptimality; it makes for a much safer filter than otherwise. It also helps with potential roundoff problems. (*Note:* Don't "blame" the filter for this kind of divergence problem. It is the fault of the designer/analyst, not the filter!)

Observability Problem

There is a third kind of divergence problem that may occur when the system is not observable. Physically, this means that there are one or more state variables (or linear combinations thereof) that are hidden from the view of the observer

(i.e., the measurements). As a result, if the unobserved processes are unstable, the corresponding estimation errors will be similarly unstable. This problem has nothing to do with roundoff error or inaccurate system modeling. It is just a simple fact of life that sometimes the measurement situation does not provide enough information to estimate all the state variables of the system. In a sense, this type of problem should not even be referred to as divergence, because the filter is still doing the best estimation possible under adverse circumstances.

There are formal tests of observability that may be applied to systems of low dimensionality. These tests are not always practical to apply, though, in higher-order systems. Sometimes one is not even aware that a problem exists until after extensive error analysis of the system (see Section 6.7). If unstable estimation errors exist, this will be evidenced by one or more terms along the major diagonal of $\mathbf{P}$ tending to increase without bound. If this is observed, and proper precautions against roundoff error have been taken, the analyst knows an observability problem exists. The only really good solution to this kind of divergence is to improve the observability situation by adding appropriate measurements to make the system completely observable.

We shall leave the subject of divergence with the simple precautions listed in the preceding paragraphs. Further comments relating to real-time implementation are continued in Section 6.11. Also, a recent book by Grewal and Andrews (19) gives a good discussion of Kalman filter implementation problems, and this reference is recommended for more on the subject.

6.7 OFF-LINE SYSTEM ERROR ANALYSIS

Optimal Error Covariance Analysis

In addition to its value as an estimator, the Kalman filter also provides a convenient means of system error analysis. In preliminary analysis and design, the analyst often needs to assess various competing designs relative to their effect on system accuracy. Recall that in the filter recursive equations, the error covariance $\mathbf{P}$ is propagated along with the estimate $\hat{\mathbf{x}}$. Usually, this cannot be avoided because $\mathbf{P}$ is needed for the gain computation and subsequent updating of the estimate.* The reverse is not true, though. The $\mathbf{P}$ matrix can be propagated without forming the estimate. With a modest amount of algebra, one can write $\mathbf{P}_{k+1}$ explicitly as a function of $\mathbf{P}_k$, and thus have a single recursive equation for the error covariance matrix (see Problem 6.4). This leads to no significant saving in the number of arithmetic steps needed in the loop, though, so the analyst usually programs the filter loop shown in Fig. 5.8 with the estimate computations omitted. The abbreviated loop usually used for analysis of the optimal estimator

* The exception is the steady-state filter where the optimal gain (or gain sequence in the periodic case) can be precomputed and stored. It is not necessary to compute $\mathbf{P}$ on-line in this case.

is summarized in Fig. 6.10. It is important to remember that in optimal error analysis it is assumed that the filter model and the true process are identical and that the optimal gain is used in the $\mathbf{P}$-update operation. It should also be mentioned that the $\mathbf{P}$ matrix does not depend on the actual measurement data received on any particular sample run. Thus, the Kalman filter in its pure form is not adaptive.

Suboptimal Error Covariance Analysis

When the filter model that is actually implemented is not the same as the true process model, a suboptimal gain sequence is generated by the real-life filter. This, of course, results in suboptimal estimates. In off-line analysis we often want to assess the degree of suboptimality caused by the mismatch between the truth model and the implemented one. This problem, in its general form, is relatively complicated, and a good deal has been written about it. The Gelb book (3) is still a good reference on the subject in spite of its age (1974). Here we will confine our attention to just the special cases where the mismatch involves the $\mathbf{R}_k$ or $\mathbf{Q}_k$ parameters (or both). These cases are amenable to relatively simple analysis.

We begin our comparative suboptimal analysis by considering the three filters shown in Fig. 6.11. The top two filters are conceptual, so they are shown with dashed lines. The lower filter is the real-life filter that we assume is actually implemented in hardware/software.

The optimal filter in Fig. 6.11 has already been discussed at some length, so no further elaboration on it is necessary. Its error covariance can be used as a baseline measure of the best possible filter performance, given the true stochastic process model and measurement situation.

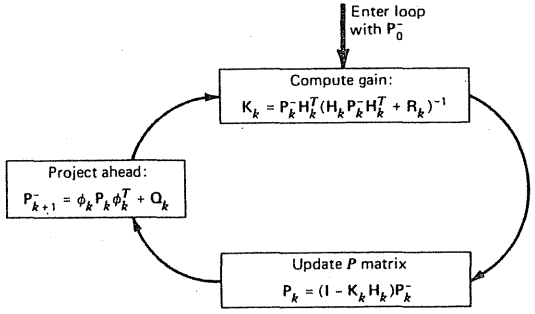


Figure 6.10 Recursive loop for propagating optimal error covariance matrix.

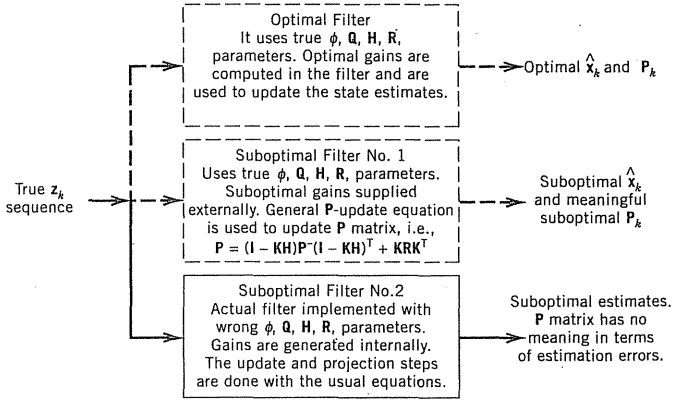


Figure 6.11 Comparison of three filters used in suboptimal filter analysis.

The significance of the conceptual suboptimal filter no. 1 in Fig. 6.11 is considerably more subtle than that for the optimal filter. We can imagine a hypothetical filter that uses the true model parameters, just as in the optimal filter, but instead of generating the gain sequence within the filter, suppose we use a gain sequence that comes from some source external to the filter. The gains may be suboptimal. Now, referring back to Chapter 5 and considering the derivation leading to Eq. (5.5.11), we find that if (1) $\mathbf{P}_k^-$ is truly representative of the error in $\hat{\mathbf{x}}_k^-$, and (2) the model parameters are correct, then $\mathbf{P}_k$ as given by the general P-update equation,

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T \qquad (6.7.1)$$

is representative of the error associated with the updated estimate, irrespective of the gain used in the update equation. This means that if we use suboptimal gains from any source in suboptimal filter no. 1, and if we are careful to use the general P-update equation (i.e., Eq. 6.7.1), then the calculated $\mathbf{P}_k$ sequence has its usual meaning in terms of the errors associated with the suboptimal estimates so generated. Note especially that suboptimal filter no. 1 is assumed to operate with the true $\boldsymbol{\phi}_k$, $\mathbf{Q}_k$, $\mathbf{H}_k$, $\mathbf{R}_k$ parameters.

Finally, we come to suboptimal filter no. 2, the one that is actually implemented in real life. How good are its estimates? This is really the central question of suboptimal analysis. We will not try to answer this question in general terms. Rather, we will look in detail at just the cases where we have implemented the wrong $\mathbf{R}_k$ or $\mathbf{Q}_k$, or both.

Special Case (a): Incorrect $\mathbf{R}_k$ In this case, all the model parameters in the real-life filter are assumed to be correct except for $\mathbf{R}_k$. We allow it to differ from the true value in most any way, except that the implemented $\mathbf{R}_k$ must be symmetric and positive definite. Having an incorrect $\mathbf{R}_k$ in the implemented filter will, of course, result in suboptimal gains and estimates. Now assume that the

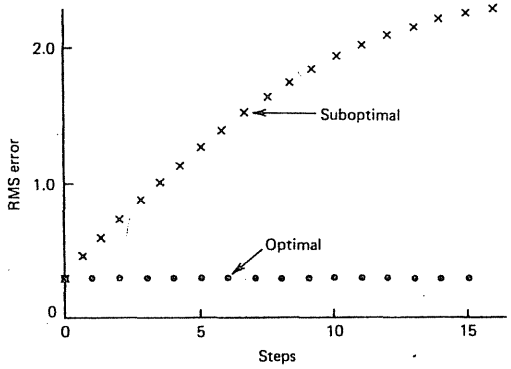
suboptimal gain sequence that is generated in the implemented filter is cycled through suboptimal filter no. 1 (the one with the correct model parameters). This being the case, the two filters will generate identical estimate sequences. This must be so, because the estimates depend only on the estimate-update and estimate-projection steps; these, in turn, depend only on the H, K, and $\boldsymbol{\phi}$ parameters that are the same in both the no. 1 and 2 suboptimal filters. The P sequence coming out of suboptimal filter no. 1 will then give a meaningful measure of the estimation errors in the implemented filter (as well as for the no. 1 filter). This is the desired result in analyzing the suboptimality effect of implementing the wrong $\mathbf{R}_k$. (Note that the P matrix generated by the implemented filter is meaningless in terms of mean-square error.)

Special Case (b): Incorrect $\mathbf{Q}_k$ All of the arguments used for special case (a) are also applicable to the case where $\mathbf{Q}_k$ in the implemented filter is not correct. If we use the suboptimal gains from the implemented filter in suboptimal filter no. 1, then the gains, $\boldsymbol{\phi}_k$ and $\mathbf{H}_k$ will be the same in both filters, and the two filters will generate identical estimate sequences. Thus, the P matrix coming out of the no. 1 suboptimal filter will be meaningful of the estimation errors in the implemented filter, just as in the wrong $\mathbf{R}_k$ case.

In summary, if either $\mathbf{R}_k$ or $\mathbf{Q}_k$ (or both) is incorrect in the real-life implemented filter, the suboptimal gains so generated may be cycled through the truth model to obtain a P matrix that accurately describes the estimation errors in the implemented filter. In doing this kind of suboptimal analysis, though, we must be careful to always use the general P-update formula [i.e., $\mathbf{P} = (\mathbf{I} - \mathbf{KH})\mathbf{P}^-(\mathbf{I} - \mathbf{KH})^T + \mathbf{KRK}^T$] in the truth-model filter. Also, there is a firm requirement that the $\boldsymbol{\phi}_k$ and $\mathbf{H}_k$ matrices in the implemented filter be the same as in the truth model. We will now look at two examples, one where this condition is satisfied and one where it is not.

EXAMPLE 6.4

We return to Example 6.3 in which a random walk process was incorrectly modeled as a random constant. One sample run of a random walk process was used to demonstrate the divergence phenomenon. However, this can hardly be called proof of divergence. We could, of course, make many more runs using new sets of random numbers, and then average the results to find the rms error. This would be doing the analysis the hard way, though. In this example it is only the $\mathbf{Q}_k$ parameter that is incorrect in the implemented filter. Therefore, all we need to do is to consider the random walk model as the truth model (suboptimal filter no. 1 in Fig. 6.11), and then feed the suboptimal gains generated by the implemented filter into the truth model. This was done for the first 15 steps of this example, and the resulting rms error along with suboptimal gains is shown in Fig. 6.12. The optimal rms error is also shown for comparison. It can be seen that divergence does occur in this situation. In this example, it can be easily verified that the suboptimal filter approaches a steady-state condition where the error variance increases by a fixed amount with each step. Thus, the rms error increases as the square root of the number of steps as indicated in Fig. 6.12. ▩



Suboptimal Gains

k	Gain
0	.9091
1	.4762
2	.3226
3	.2439
4	.1960
5	.1639
6	.1408
7	.1234
8	.1099
9	.0990
10	.0901
11	.0826
12	.0763
13	.0709
14	.0662
15	.0621

Figure 6.12 Suboptimal analysis for random walk example.

EXAMPLE 6.5

An optimal prediction application to GPS was presented in Example 6.1. The end result of the example was a plot of the rms prediction error for the range correction for the optimal predictor. It is also of interest to compare the optimal results with corresponding results for a suboptimal predictor that is being considered for this application (17). The suboptimal predictor simply takes the range and range-rate corrections, as provided at the start time, and projects these ahead with a constant rate (much as is done in dead reckoning). This, of course, does not take advantage of any prior knowledge of the spectral characteristics of the SA process.

The continuous dynamic model for the suboptimal predictor is

$$\ddot{x} = w(t) \tag{6.7.2}$$

where x is range and $w(t)$ is white noise. [The PSD of $w(t)$ does not affect the projection of x.] If we choose range and range rate as our state variables, the continuous state model becomes

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}}_{\mathbf{F}_{\text{sub}}} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} w(t) \tag{6.7.3}$$

The $\mathbf{F}$ matrix for the suboptimal model can now be compared with $\mathbf{F}$ for the optimal model from Example 6.1. It is

$$\mathbf{F}_{\text{SA}} = \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & -\sqrt{2}\,\omega_0 \end{bmatrix} \quad \text{(optimal model)} \tag{6.7.4}$$

The optimal system is the truth model in this example, and clearly, $\mathbf{F}_{\text{SA}}$ and $\mathbf{F}_{\text{sub}}$

are quite different. This means then that the $\boldsymbol{\phi}_k$ matrices for the two models will be different, and this precludes the use of the "recycling suboptimal gains" method of analyzing the suboptimal system performance. All is not lost, though. In this simple situation we can return to basics and write out an explicit expression for the suboptimal prediction error. An explicit equation for the error covariance as a function of prediction time can then be obtained.

The optimal model is the SA model given in Example 6.1. Therefore, the true $\mathbf{x}$ at step $k + N$ is

$$\mathbf{x}_{k+N} = \boldsymbol{\phi}_{\text{SA}}(k + N, k)\,\mathbf{x}_k + \mathbf{w}_N \tag{6.7.5}$$

where N denotes the steps ahead, beginning at step k, and $\mathbf{w}_N$ is the white-noise contribution to the state vector that accumulates for N steps. The covariance associated with $\mathbf{w}_N$ is

$$\mathbf{Q}_{\text{SA}}(k + N, k) = E[\mathbf{w}_N \mathbf{w}_N^T] \tag{6.7.6}$$

The parameters of the continuous SA model are known, so $\boldsymbol{\phi}_{\text{SA}}$ and $\mathbf{Q}_{\text{SA}}$ can be computed as a function of N.

The state transition matrix for the suboptimal dynamic model of Eq. (6.7.3) is

$$\boldsymbol{\phi}_{\text{sub}}(N) = \begin{bmatrix} 1 & N\,\Delta t \\ 0 & 1 \end{bmatrix} \tag{6.7.7}$$

where

N = number of prediction steps
Δt = step size for each step (sec)

The predictive estimate produced by the suboptimal predictor is then

$$\hat{\mathbf{x}}(k + N|k) = \boldsymbol{\phi}_{\text{sub}}(N)\,\hat{\mathbf{x}}(k|k) \tag{6.7.8}$$

But, we assume that we begin the prediction with a perfect estimate of $\mathbf{x}$ at $t = t_k$. Therefore,

$$\hat{\mathbf{x}}(k|k) = \mathbf{x}_k \tag{6.7.9}$$

We can now form the difference between the true $\mathbf{x}$ and its suboptimal estimate. Using Eqs. (6.7.5) to (6.7.9) leads to

$$\begin{aligned} \mathbf{e}(N) &= \mathbf{x}_{k+N} - \hat{\mathbf{x}}(k + N|k) \\ &= [\boldsymbol{\phi}_{\text{SA}}(N) - \boldsymbol{\phi}_{\text{sub}}(N)]\mathbf{x}_k + \mathbf{w}_N \end{aligned} \tag{6.7.10}$$

and

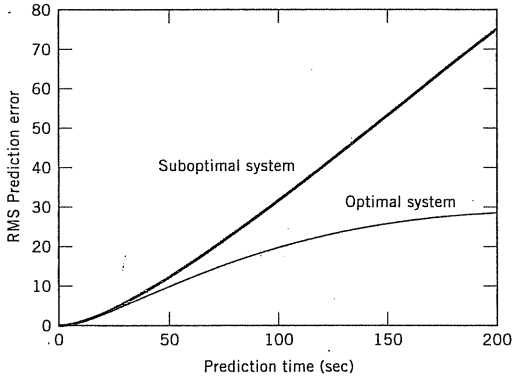


Figure 6.13 Comparison of rms prediction errors for optimal and suboptimal predictors.

$$\begin{aligned}\mathbf{P}_{\text{sub}} &= E[\mathbf{e}(N)\mathbf{e}^T(N)] \\ &= [\boldsymbol{\phi}_{\text{SA}}(N) - \boldsymbol{\phi}_{\text{sub}}(N)]\, E(\mathbf{x}_k\mathbf{x}_k^T) \\ &\quad \times [\boldsymbol{\phi}_{\text{SA}}(N) - \boldsymbol{\phi}_{\text{sub}}(N)]^T + \mathbf{Q}_{\text{SA}}(k+N, k)\end{aligned} \tag{6.7.11}$$

The true process $\mathbf{x}_k$ is stationary, so $E(\mathbf{x}_k\mathbf{x}_k^T)$ is easily evaluated using the methods of Chapter 3. For the model parameters given in Example 6.1, we have

$$E(\mathbf{x}_k\mathbf{x}_k^T) = \begin{bmatrix} (30\text{ m})^2 & 0 \\ 0 & (.36\text{ m/sec})^2 \end{bmatrix} \tag{6.7.12}$$

Now, all the remains to be done is to evaluate $\boldsymbol{\phi}_{\text{SA}}$, $\boldsymbol{\phi}_{\text{sub}}$, and $\mathbf{Q}_{\text{SA}}$ for $N = 0, 1, 2, \ldots$, and then use Eq. (6.7.11) to get the estimation error covariances for the desired number of steps. This is easily done with MATLAB. The desired comparative results are shown in Fig. 6.13. For the first few seconds of prediction, there is very little difference between the optimal and suboptimal predictors. However, as the prediction time increases, the difference becomes more pronounced. For a prediction time of 50 sec (which was used as a reference point in Example 6.1), the error comparison is about 10 m for the optimal predictor vs. 12.5 m for the suboptimal one. The difference is significant, and the improvement in going from suboptimal to optimal is achieved simply with software. ▩

6.8 RELATIONSHIP TO DETERMINISTIC LEAST SQUARES AND NOTE ON ESTIMATING A CONSTANT

Both Kalman and Wiener filtering are sometimes referred to simply as least-squares filtering (11, 12, 13). It was mentioned in Chapter 4 that this is somewhat

an oversimplification, because the criterion for optimization is minimum *mean*-square error and not the squared error in a deterministic sense. There is, however, a coincidental connection between Kalman/Wiener filtering and deterministic least squares, and this will now be demonstrated. The presentation here follows closely that of Sorenson (4).

Consider a set of m linear equations in $\mathbf{x}$ specified in matrix form by

$$\mathbf{M}\mathbf{x} = \mathbf{b} \tag{6.8.1}$$

In Eq. (6.8.1) we think of $\mathbf{M}$ and $\mathbf{b}$ as being given, and $\mathbf{x}$ is $(n \times 1)$, $\mathbf{b}$ is $(m \times 1)$, and thus $\mathbf{M}$ is $(m \times n)$. Let us assume that $m > n$, and that $\mathbf{x}$ is overdetermined by the system of equations represented by Eq. (6.8.1). Thus, no solution for $\mathbf{x}$ will satisfy all equations. This situation arises frequently in physical experiments where redundant noisy measurements are made of linear combinations of fixed parameters. In such cases it is logical to ask, "What solution will best fit all the equations?" The term *best* must, of course, be defined and it is frequently defined to be the particular $\mathbf{x}$, say $\mathbf{x}_{\text{opt}}$, that minimizes the sum of the squared residuals. That is, move $\mathbf{b}$ to the left side of Eq. (6.8.1) and substitute $\mathbf{x}_{\text{opt}}$ for $\mathbf{x}$. This yields a residual vector $\boldsymbol{\varepsilon}$ given by

$$\mathbf{M}\mathbf{x}_{\text{opt}} - \mathbf{b} = \boldsymbol{\varepsilon} \tag{6.8.2}$$

and $\mathbf{x}_{\text{opt}}$ is chosen such that $\boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon}$ is minimized. A perfect fit, of course, would make $\boldsymbol{\varepsilon}^T\boldsymbol{\varepsilon} = 0$.

We can generalize at this point and consider a weighted sum of squared residuals specified by

$$\begin{bmatrix}\text{Weighted sum of} \\ \text{squared residuals}\end{bmatrix} = (\mathbf{M}\mathbf{x}_{\text{opt}} - \mathbf{b})^T\,\mathbf{W}(\mathbf{M}\mathbf{x}_{\text{opt}} - \mathbf{b}) \tag{6.8.3}$$

We assume that the weighting matrix $\mathbf{W}$ is symmetric and positive definite and, hence, so is its inverse. If we wish equal weighting of the residuals, we simply let $\mathbf{W}$ be the identity matrix. The problem now is to find the particular $\mathbf{x}$ (i.e., $\mathbf{x}_{\text{opt}}$) that minimizes the weighted sum of the residuals. Toward this end, the expression given by Eq. (6.8.3) may be expanded and differentiated term by term and then set equal to zero.* This leads to

* The derivative of a scalar s with respect to a vector $\mathbf{x}$ is defined to be

$$\frac{ds}{d\mathbf{x}} = \begin{bmatrix} \frac{ds}{dx_1} \\ \frac{ds}{dx_2} \\ \vdots \\ \frac{ds}{dx_n} \end{bmatrix}$$

(continued on next page)

$$\frac{d}{d\mathbf{x}_{\text{opt}}}\left[\mathbf{x}_{\text{opt}}^T(\mathbf{M}^T\mathbf{W}\mathbf{M})\mathbf{x}_{\text{opt}} - \mathbf{b}^T\mathbf{W}\mathbf{M}\mathbf{x}_{\text{opt}} - \mathbf{x}_{\text{opt}}^T\mathbf{M}^T\mathbf{W}\mathbf{b} + \mathbf{b}^T\mathbf{b}\right]$$

$$= 2(\mathbf{M}^T\mathbf{W}\mathbf{M})\mathbf{x}_{\text{opt}} - (\mathbf{b}^T\mathbf{W}\mathbf{M})^T - \mathbf{M}^T\mathbf{W}\mathbf{b} = 0 \tag{6.8.4}$$

Equation (6.8.4) may now be solved for $\mathbf{x}_{\text{opt}}$. The result is

$$\mathbf{x}_{\text{opt}} = [(\mathbf{M}^T\mathbf{W}\mathbf{M})^{-1}\mathbf{M}^T\mathbf{W}]\mathbf{b} \tag{6.8.5}$$

and this is the solution of the deterministic least-squares problem.

Next consider the Kalman filter solution for the same measurement situation. The vector $\mathbf{x}$ is assumed to be a random constant, so the differential equation for $\mathbf{x}$ is

$$\dot{\mathbf{x}} = 0 \tag{6.8.6}$$

The corresponding discrete model is then

$$\mathbf{x}_{k+1} = \mathbf{I} \cdot \mathbf{x}_k + \mathbf{0} \tag{6.8.7}$$

The measurement equation is

$$\mathbf{z}_k = \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k \tag{6.8.8}$$

where $\mathbf{z}_k$ and $\mathbf{H}_k$ play the same roles as $\mathbf{b}$ and $\mathbf{M}$ in the deterministic problem. Since time is of no consequence, we assume that all measurements occur simultaneously. Furthermore, we assume that we have no a priori knowledge of $\mathbf{x}$, so the initial $\hat{\mathbf{x}}_0^-$ will be zero and its associated error covariance will be ∞. Therefore, using the alternative form of the Kalman filter (Section 6.2), we have

$$\mathbf{P}_0^{-1} = (\infty)^{-1} + \mathbf{H}_0^T\mathbf{R}_0^{-1}\mathbf{H}_0$$

$$= \mathbf{H}_0^T\mathbf{R}_0^{-1}\mathbf{H}_0 \tag{6.8.9}$$

The Kalman gain is then

The two matrix differentiation formulas used to arrive at Eq. (6.8.4) are

$$\frac{d(\mathbf{x}^T\mathbf{A}\mathbf{x})}{d\mathbf{x}} = 2\mathbf{A}\mathbf{x} \quad \text{(for symmetric A)}$$

and

$$\frac{d(\mathbf{a}^T\mathbf{x})}{d\mathbf{x}} = \frac{d(\mathbf{x}^T\mathbf{a})}{d\mathbf{x}} = \mathbf{a}$$

Both of these formulas can be verified by writing out a few scalar terms of the matrix expressions and using ordinary differentiation methods.

$$\mathbf{K}_0 = (\mathbf{H}_0^T\mathbf{R}_0^{-1}\mathbf{H}_0)^{-1}\mathbf{H}_0^T\mathbf{R}_0^{-1}$$

and the Kalman filter estimate of $\mathbf{x}$ at $t = 0$ is

$$\hat{\mathbf{x}}_0 = [(\mathbf{H}_0^T\mathbf{R}_0^{-1}\mathbf{H}_0)^{-1}\mathbf{H}_0^T\mathbf{R}_0^{-1}]\mathbf{z}_0 \tag{6.8.10}$$

This is the same identical expression obtained for $\mathbf{x}_{\text{opt}}$ in the deterministic least-squares problem with $\mathbf{R}_0^{-1}$ playing the role of the weighting matrix $\mathbf{W}$.

Let us now recapitulate the conditions under which the Kalman filter estimate coincides with the deterministic least-squares estimate. First, the system state vector was assumed to be a random constant (the dynamics are thus trivial). Second, we assumed the measurement sequence was such as to yield an overdetermined system of linear equations [otherwise $(\mathbf{H}_0^T\mathbf{R}_0^{-1}\mathbf{H}_0)^{-1}$ will not exist]. And, finally, we assumed that we had no prior knowledge about the constant vector being estimated. This latter assumption is unusual because in many situations we have at least some a priori knowledge of the process being estimated. One of the things that distinguishes the Kalman filter from other estimators is the convenient way in which it accounts for this prior knowledge via the initial conditions of the recursive process. (This was used to good advantage in the power system relaying application of Section 6.4.). Of course, if there is truly no prior knowledge to use, the Kalman filter advantage is lost (in this respect), and it degenerates to a least-squares fit under the conditions just stated.

The coincidence in the deterministic least-squares and Kalman filter estimates is really rather remarkable. Remember, one solution was obtained by posing a *deterministic* optimization problem, the other by posing a similar *stochastic* problem. There is no reason *offhand* to think these two approaches would lead to identical solutions. Yet they do under certain circumstances. The circumstances may be generalized somewhat from those of this example, but not to the complete extent of the general process model used in the Kalman filter. [See Sorenson (12) for more on this point.] Thus, this happy coincidence in the two solutions will not always exist.

There are subtleties that need to be recognized when using Kalman filtering to estimate a constant. These are related to deterministic least squares. A simple example will illustrate the significance of the Kalman filter estimate in the unknown-constant estimation problem.

EXAMPLE 6.6

Consider an elementary physics experiment that is intended to measure the gravity constant g. A mass is released at $t = 0$ in a vertical, evacuated column, and multiple-exposure photographs of the falling mass are taken at .05-sec intervals beginning at $t = .05$ sec. A sequence of N such exposures is taken, and then the position of the mass at each time is read from a scale in the photograph. There will be experimental errors for a number of reasons; let us assume that they are random (i.e., not systematic) and are such that the statistical uncertainties in all position readings are the same and that the standard deviation of these is σ.

Consider g to be an unknown constant, and suppose we say that we have no prior knowledge about its value. (We will elaborate on this assumption later

in the example.) We wish to develop a Kalman filter for processing the noisy position measurements. We begin our model by letting g be the single state variable x. The discrete state equation is then

$$x_{k+1} = 1 \cdot x_k + 0 \tag{6.8.11}$$

The measurement sequence z_k is related to x_k via the equations

$$\begin{aligned} z_1 &= (\tfrac{1}{2}t_1^2)x_1 + v_1 \quad \text{(the first measurement is at } t_1\text{, not } t_0\text{)} \\ z_2 &= (\tfrac{1}{2}t_2^2)x_2 + v_2 \\ &\vdots \\ z_N &= (\tfrac{1}{2}t_N^2)x_N + v_N \end{aligned} \tag{6.8.12}$$

The filter parameters are then

$$\begin{aligned} \phi_k &= 1, \qquad Q_k = 0 \\ H_k &= \tfrac{1}{2}t_k^2, \qquad R_k = \sigma^2 \end{aligned} \tag{6.8.13}$$

The measurement sequence begins at $t = t_1 = .05$ sec, so we enter the recursive loop at $k = 1$ with the initial conditions

$$\begin{aligned} \hat{x}_1^- &= 0 \\ P_1^{-1} &= \infty \quad \text{(no prior knowledge of } g\text{)} \end{aligned} \tag{6.8.14}$$

The alternative algorithm is useful here to get the recursive process started. Therefore, we update P_1^{-1} first using Eq. (6.2.6).

$$P_1^{-1} = (P_1^-)^{-1} + (\tfrac{1}{2}t_1^2)\left(\frac{1}{\sigma^2}\right)(\tfrac{1}{2}t_1^2)$$

Or

$$P_1 = \frac{4}{t_1^4}\sigma^2 \tag{6.8.15}$$

The Kalman gain is computed next using Eq. (6.2.7).

$$K_1 = P_1 H_1^T R_1^{-1} = \frac{2}{t_1^2} \tag{6.8.16}$$

Finally, the estimate is updated using the usual update equation:

$$\hat{x}_1 = 0 + \left(\frac{2}{t_1^2}\right)(z_1 - 0) = \left(\frac{2}{t_1^2}\right)z_1 \tag{6.8.17}$$

The a posteriori P is finite, so we can now project ahead to the next measurement and proceed with the recursive processing using either the regular algorithm (Chapter 5) or the alternative algorithm (Chapter 6). This is routine, so we will not pursue the algebra further here. It is easily demonstrated that in these circumstances the Kalman filter estimate obtained at each step is identical to that obtained from deterministic least squares and batch processing the data (see Problem 6.7).

This example is similar to Example 6.2 in many respects, but there are subtle differences that warrant further elaboration. In Example 6.2 the process envisioned was an ensemble of random constants (see also Example 2.5). Thus, when we think of this in terms of a statistical experiment, each time function in the ensemble is constant with time, but their amplitudes are different in that they are sample realizations of a zero-mean random variable with a large variance. Thus, the setting for Example 6.2 is truly a stochastic setting, and all the conclusions about minimizing the mean-square error, and so forth, are applicable. Now contrast this with the setting for the gravity-determination experiment of the present example. Not only is it unrealistic to say that we have no prior knowledge of g, but the unknown g is not really properly modeled as a random process. Presumably, if we were to repeat our elementary physics experiment over and over again at the same location, we would have the same g each time the experiment was performed. Thus, the stochastic setting here is different from the one for Example 6.2. The gravity-experiment measurement noise may be random, but the quantity being estimated (even if unknown) is not. Now, there is nothing wrong with applying the Kalman filter algorithm in this setting. After all, it can be viewed as just one of many arithmetic rules for processing numerical data. The fault, if there is fault, comes with the interpretation of the results. In Example 6.2 we have a right to expect the Kalman filter to minimize the mean-square error in a truly stochastic (ensemble averaging) sense. In the present example we cannot draw the same inference. We do know in the present example that the resulting estimate is identical with the ordinary deterministic least-squares estimate (see Problem 6.8), but beyond this we should not draw hasty conclusions about optimality. ▩

6.9 DISCRETE KALMAN FILTER STABILITY

A Kalman filter is sometimes referred to as a time-domain filter, because the design is done in the time domain rather than the frequency domain; of course, one of the beauties of the Kalman filter is its ability to accommodate time-variable parameters. However, there are some applications where the filter, after many recursive steps, approaches a steady-state condition. When this happens and the sampling rate is fixed, the Kalman filter behaves much the same as any other digital filter (20), the main difference being the vector input/output prop-

erty of the Kalman filter. The stability of conventional digital filters is easily analyzed with z-transform methods. We shall proceed to do the same for the Kalman filter.

We begin by assuming that the Kalman filter under consideration has reached a constant-gain condition. The basic estimate update equation is repeated here for convenience:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-) \tag{6.9.1}$$

We first need to rewrite Eq. (6.9.1) as a first-order vector difference equation. Toward this end, we replace $\hat{\mathbf{x}}_k^-$ with $\boldsymbol{\phi}_{k-1}\hat{\mathbf{x}}_{k-1}$ in Eq. (6.9.1). The result is

$$\hat{\mathbf{x}}_k = (\boldsymbol{\phi}_{k-1} - \mathbf{K}_k\mathbf{H}_k\boldsymbol{\phi}_{k-1})\hat{\mathbf{x}}_{k-1} + \mathbf{K}_k\mathbf{z}_k \tag{6.9.2}$$

We now take the z-transform of both sides of Eq. (6.9.2) and note that retarding $\hat{\mathbf{x}}_k$ by one step in the time domain is the equivalent of multiplying by z^{-1} in the z-domain. This yields (in the z-domain)

$$\hat{\mathbf{X}}_k(z) = (\boldsymbol{\phi}_{k-1} - \mathbf{K}_k\mathbf{H}_k\boldsymbol{\phi}_{k-1})z^{-1}\hat{\mathbf{X}}_k(z) + \mathbf{K}_k\mathbf{Z}_k(z) \tag{6.9.3}$$

Or, after rearranging terms, we have

$$[z\mathbf{I} - (\boldsymbol{\phi}_{k-1} - \mathbf{K}_k\mathbf{H}_k\boldsymbol{\phi}_{k-1})]\hat{\mathbf{X}}(z) = z\mathbf{K}_k\mathbf{Z}_k(z) \tag{6.9.4}$$

[We note that in Eqs. (6.9.3) and (6.9.4), italic z denotes the usual z-transform variable, whereas boldface $\mathbf{Z}_k$ refers to the z-transformed measurement vector.]

We know from linear system theory that the bracketed quantity on the left side of Eq. (6.9.4) describes the natural modes of the system. The determinant of the bracketed $n \times n$ matrix gives us the characteristic polynomial for the system, that is,

$$\text{Characteristic polynomial} = |z\mathbf{I} - (\boldsymbol{\phi}_{k-1} - \mathbf{K}_k\mathbf{H}_k\boldsymbol{\phi}_{k-1})| \tag{6.9.5}$$

and the roots of this polynomial provide information about the filter stability. If all the roots lie inside the unit circle in the z-plane, the filter is stable; conversely, if any root lies on or outside the unit circle, the filter is unstable. [As a matter of terminology, the roots of the characteristic polynomial are the same as the eigenvalues of $(\boldsymbol{\phi}_{k-1} - \mathbf{K}_k\mathbf{H}_k\boldsymbol{\phi}_{k-1})$]. A simple example will illustrate the usefulness of the stability concept.

EXAMPLE 6.7

Let us return to the random walk problem of Example 6.3 and investigate the stability of the filter in the steady-state condition. The discrete model in this example is

$$x_{k+1} = x_k + w_k \tag{6.9.6}$$

$$z_k = x_k + v_k \tag{6.9.7}$$

and the discrete filter parameters are

$$\phi_k = 1, \quad H_k = 1, \quad Q_k = 1, \quad R_k = .1, \quad P_0^- = 1, \quad \hat{x}_0^- = 0$$

In this example the gain reaches steady state in just a few steps, and it is easily verified that its steady-state value is

$$K_k \approx .916$$

We can now form the characteristic polynomial from Eq. (6.9.5):

$$\begin{aligned}\text{Characteristic polynomial} &= z - [1 - (.916)(1)(1)] \\ &= z - .084\end{aligned} \tag{6.9.8}$$

The characteristic root is at .084, which is well within the unit circle in the z-plane. Thus we see that the filter is highly stable in this case.

Note that even though the input in this case is nonstationary, the filter itself is intrinsically stable. Furthermore, the filter pole location tells us that any small perturbation from the steady-state condition (e.g., due to roundoff error) will damp out quickly. Any such perturbations will be attenuated by a factor of .084 with each step in this case, so their effect "vaporizes" rapidly. This same kind of reasoning can be extended to the vector case, provided that the $\mathbf{P}$ matrix is kept symmetric in the recursive process and that it is never allowed to lose its positive definiteness. Thus, we see that we can gain considerable insight into the filter operation just by looking at its characteristic poles in the steady-state condition, provided, of course, that a steady-state condition exists. ■

6.10 DETERMINISTIC INPUTS

In many situations the random processes under consideration are driven by deterministic as well as random inputs. That is, the process equation may be of the form

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{u} + \mathbf{B}\mathbf{u}_d \tag{6.10.1}$$

where $\mathbf{B}\mathbf{u}_d$ is the additional deterministic input. Since the system is linear, we can use superposition and consider the random and deterministic responses separately. Thus, the discrete Kalman filter equations are modified only slightly. The only change required is in the estimate projection equation. In this equation

the contribution due to $\mathbf{B}\mathbf{u}_d$ must be properly accounted for. Using the same zero-mean argument as before, relative to the random response, we then have

$$\hat{\mathbf{x}}_{k+1}^- = \boldsymbol{\phi}_k \hat{\mathbf{x}}_k + 0 + \int_{t_k}^{t_{k+1}} \boldsymbol{\phi}(t_{k+1}, \tau)\mathbf{B}(\tau)\mathbf{U}_d(\tau)\, d\tau \qquad (6.10.2)$$

where the integral term is the contribution due to $\mathbf{B}\mathbf{u}_d$ in the interval (t_k, t_{k+1}). The associated equation for $\mathbf{P}_{k+1}^-$ is $(\boldsymbol{\phi}_k \mathbf{P}_k \boldsymbol{\phi}_k^T + \mathbf{Q}_k)$, as before, because the uncertainty in the deterministic term is zero. Also, the estimate update and associated covariance expressions (see Fig. 5.8) are unchanged, provided the deterministic contribution has been properly accounted for in computing the a priori estimate $\hat{\mathbf{x}}_k^-$.

Another way of accounting for the deterministic input is to treat the problem as a superposition of two entirely separate estimation problems, one deterministic and the other random. The deterministic one is trivial, of course, and the random one is not trivial. This complete separation approach is not necessary, though, provided one properly accounts for the deterministic contribution in the projection step.

6.11 REAL-TIME IMPLEMENTATION ISSUES

One of the more attractive features of the Kalman filter is its recursive nature and its modest use of memory storage. This aspect of the Kalman filter makes it a very useful data processing tool in real-time applications. In such applications, the computations of the Kalman filter must take less time to execute than the time interval containing the total number of measurements processed by that Kalman filter. If this were not true, one of two things could happen: The Kalman filter processing will only process a reduced number of measurements in order to keep up with the progression of time and ignore the remaining measurements, or the Kalman filter will insist on processing everything presented to it and gradually fall behind in the timeliness of computing its solution. The former is still a real-time filter but may be a suboptimal one. The latter is certainly no longer considered a real-time filter. In this section, we shall focus on Kalman filter implementation issues in the context of being both real-time and optimal.

Data Latency

In the real-time world, it is impossible to expect the processing of a solution to be completed instantaneously at where the measurements are made. There is always a finite time delay that must be accommodated. This *latency* may be associated with the delivery of the measurements, the Kalman filter computation time, and also the conveyance of the solution to where it is needed (see Figure 6.14). In general, with transmission and processing delays, the solution latency may constitute a significant fraction of the time interval between measurements.

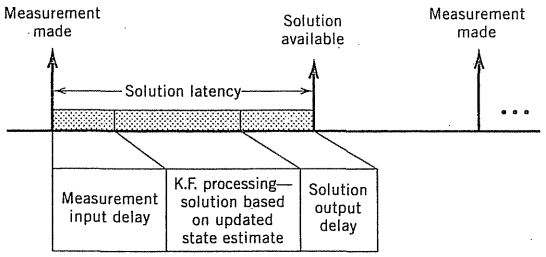


Figure 6.14 Timeline for a typical real-time processing with solution latency.

To present a timely solution at the point when the solution is ready, one alternative is to project the Kalman filter solution during the processing to a time in the future; the solution presented at that point in time would be *timely*. The penalty one pays for this timeliness is that the error covariance associated with the solution may not be at the lowest possible value because the solution propagation constitutes an additional prediction error. The choice of where to project this solution is up to the designer's judgment. However, one convenient choice is to project the solution to the time point where the next measurement is expected to be made (see Figure 6.15) because this computation already exists as part of the normal Kalman filter cycle. The solution can simply be derived from the projection of the state vector without the need to project to an intermediate time point.

Processor Loading

The determination of how much the processor is being exercised when running a Kalman filter in real time is commonly known as loading or throughput analysis. Such studies are ad hoc in nature because of dependencies on the type and speed of the processor used. It is not within our scope here to pursue such an analysis. In general, if the entire processor is dedicated solely to Kalman filter

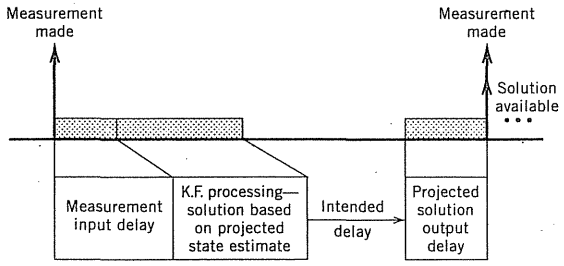


Figure 6.15 Timeline for "timely" projected solution with no solution latency.

computations, the throughput analysis is quite straightforward. The total amount of time needed to execute the entire algorithm for a single cycle of the Kalman filter can be derived from the number of additions and multiplications needed to implement the matrix equations associated with the Kalman filter. How this length of time fits into the interval between the arrival of measurements will determine the viability of the real-time processing. In most real-life situations, however, the processor that runs the Kalman filter is also tasked with running other functions that are sometimes more critical and also more time-consuming than the Kalman filter itself. In such multitasking environments, the answer to the feasibility of running a Kalman filter in real time is much less obvious.

To lower processor loading, computational efficiency may be improved by taking advantage of matrix sparseness or matrix symmetry where such characteristics appear. Particularly in systems where the dimensionality is high, the Kalman parameters $\boldsymbol{\phi}$, $\mathbf{Q}$, $\mathbf{H}$, and $\mathbf{R}$ are generally quite sparse, containing in them many zero- and singular-valued elements. To exploit the sparseness, certain matrix equations of the standard Kalman filter will have to be written out explicitly rather than as generalized matrix algorithms. The following example illustrates the comparison between the two approaches for the state projection of a 3-tuple state vector:

$$\hat{\mathbf{x}}_{k+1}^- = \boldsymbol{\phi}\hat{\mathbf{x}}_k$$

The generalized computation would be

$$\left.\begin{aligned} \hat{x}_{1\,k+1}^- &= \phi_{11}\cdot\hat{x}_{1k} + \phi_{12}\cdot\hat{x}_{2k} + \phi_{13}\cdot\hat{x}_{3k} \\ \hat{x}_{2\,k+1}^- &= \phi_{21}\cdot\hat{x}_{1k} + \phi_{22}\cdot\hat{x}_{2k} + \phi_{23}\cdot\hat{x}_{3k} \\ \hat{x}_{3\,k+1}^- &= \phi_{31}\cdot\hat{x}_{1k} + \phi_{32}\cdot\hat{x}_{2k} + \phi_{33}\cdot\hat{x}_{3k} \end{aligned}\right\} \text{ 9 multiplies and 6 additions}$$

Suppose that

$$\boldsymbol{\phi} = \begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The specialized computation would then be

$$\left.\begin{aligned} \hat{x}_{1\,k+1}^- &= \hat{x}_{1k} + \Delta t\cdot\hat{x}_{2l} \\ \hat{x}_{2\,k+1}^- &= \hat{x}_{2k} \\ \hat{x}_{3\,k+1}^- &= \hat{x}_{3k} \end{aligned}\right\} \text{ 1 multiply and 1 addition}$$

The computational savings in this simple example are significant. It is generally the case that additional software code must be generated to implement a specialized matrix computation. In the example above, however, it turns out that the software code needed to implement the specialized computation is actually simpler than the generalized one.

Another simplification that may potentially be exploited involves systems whose measurement availability and rate are high. In such systems, the error covariance and the corresponding Kalman gain do not change much from one cycle to the next. If this is the case, then instead of updating these quantities every cycle, doing so at a lower rate will result in a substantial amount of computational savings. This approximation should not cause any instability as long the system remains observable from a steady rate of new measurement information.

State Estimates and Error Covariance Propagation

In systems where the time interval between measurements can vary substantially, there are at least two ways of mechanizing the propagation of the state estimate vector $\hat{\mathbf{x}}_k$ and the error covariance matrix $\mathbf{P}_k$. One way is to define the fundamental (shortest) time interval between measurements and propagate consistently using values of the state transition matrix $\boldsymbol{\phi}$ and the process noise covariance $\mathbf{Q}$ computed for a fixed Δt. This approach is well suited to a real-time environment where it is usually important to maintain regularity in the computational cycles for the sake of consistency. In other words, even if there are no measurements available for a particular Kalman filter cycle, the state estimate and error covariance propagation will be carried out after a trivial update of those very quantities (without need to compute the gain, the update equations are simply $\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^-$ and $\mathbf{P}_k = \mathbf{P}_k^-$). In the absence of measurements, the state estimate and error covariance are simply propagated repeatedly. However, this continuous propagation, if carried out over a large number of cycles, may result in problems of numerical stability. In other words, the drawback of this approach becomes evident when the time interval between measurements is large compared to the fundamental computational time interval.

A second way to handle variable propagation time intervals is to specially compute the appropriate state transition and process noise covariance matrices for the entire time interval needed for the propagation. The difficulty of accommodating the "on demand" computation of $\boldsymbol{\phi}_k$ and $\mathbf{Q}_k$ may vary depending on how each parameter is derived. In any case, for a real-time implementation of this approach, a slight adjustment to the sequence of computation is in order. We have, up to this point, been considering the propagation equations to be made after the state estimate and error covariance updates in the processing cycle (see Fig. 5.8). At that point of the processing cycle when the propagation equations are to be computed, unless the availability of the next measurement is predetermined, the stretch of time over which the state estimates and error covariance must be propagated is indeterminable. Instead, it is usually best to defer the propagation computations to the start of the following computational cycle at the time when the measurements next become available.

6.12 PERSPECTIVE

We have now presented the basics of Kalman filtering and looked at a few examples of how the technique can be applied in physical situations. More is yet to come in the remaining chapters. However, this is a good place to pause

for a moment and reflect on just what we have (and do not have) with this thing we call a Kalman filter.

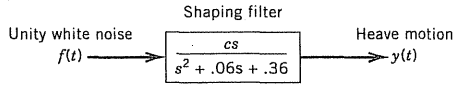
1. The Kalman filter is intended to be used for estimating *random* processes. Any application in a nonrandom setting must be viewed with caution (see Example 6.6).
2. The Kalman filter is model-dependent. This is to say that we assume that we know a priori the model parameters. These, in turn, come from the second-order "statistics" of the various processes involved in the application at hand. Therefore, in its most primitive form, the Kalman filter is not adaptive or self-learning.
3. The Kalman filter is a linear estimator. When all the processes involved are Gaussian, the filter is optimal in the minimum-mean-square-error sense within a class of *all* estimators, linear and nonlinear. [See Meditch (2) for a good discussion of optimiality.]
4. Various Kalman filter recursive algorithms exist. The "usual" algorithm was given in Chapter 5, an alternative one was presented in Chapter 6, and a third one (*U-D* factorization) will be presented in Chapter 9. All of these yield identical results (assuming perfect arithmetic).
5. Under certain special circumstances, the Kalman filter yields the same result obtained from deterministic least squares (see Section 6.8).
6. Kalman filtering is especially useful as an analysis tool in off-line error analysis studies. The optimal filter error covariance equation can be propagated recursively without actual (or simulated) measurement data. This is also true for the suboptimal filter with some restrictions (see Section 6.7).

With these brief comments in mind, we are now ready to proceed to variations on the original discrete Kalman filter. It is worth mentioning that the discrete filter came first historically (1960). The continuous version and other variations followed the discrete filter.

PROBLEMS

6.1 The process of landing on an aircraft carrier is a highly complex operation primarily because the carrier deck is constantly in motion with a certain degree of randomness that is attributable to wind and sea conditions. In particular, one motion called *heaving* changes the vertical displacement of the carrier deck. Accurate prediction of the heave motion even 10 to 15 sec into the future will significantly enhance the success of the landing operation.

In a paper published in 1983, Sidar and Doolin (21) suggested using Kalman filter methods to predict the motion of the carrier deck. On the basis of empirical data, they developed a power spectral density (PSD) for the heave motion, and then they worked out an optimal predictor based on this spectral model. The functional form for the PSD to be used here comes from the Sidar–Doolin paper, but the amplitude factor has been changed for convenience. Also, the measurement noise variance $\mathbf{R}_k$ and the sampling interval used here are hypothetical. Thus, there is no claim that the results of this problem represent an exact real-life situation.



Problem 6.1

The random process model for the heave motion is described in the accompanying figure. First note that the undamped natural frequency for the transfer function in the figure is $\sqrt{.36} = .6$ rad/sec. The damping ratio is then $.06/(2)(.6) = .05$. Thus, the heave motion is a relatively narrowband noise process with most of its spectral content concentrated around .6 rad/sec (or about .1 Hz).

(a) First, develop a suitable continuous state model for the heave motion. Choose the scale factor c such that the steady-state rms heave motion is 2 m. Note that for this process there is some risk in choosing velocity as one of the state variables. Theoretically, it has infinite variance. (See Section 5.2 on modeling processes with rational PSD functions.)

(b) Now design a Kalman filter/predictor for the model developed in part (a). The sampling interval is to be 1 sec, and $\mathbf{R}_k = 1$ m^2. The measurement sequence can be thought of as coming from a source completely independent of the landing operation (e.g., GPS). For this situation, find the rms prediction error for N steps ahead for $N = 0, 1, 2, \ldots, 20$. Plot the result. The plot should show a significant reduction of estimation error for prediction times around 10 sec, as compared with the 2-m rms error that would exist without prediction.

6.2 Consider the measurement to be a 2-tuple $[z_1, z_2]^T$, and assume that the measurement errors are correlated such that the $\mathbf{R}$ matrix is of the form

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} \\ r_{12} & r_{22} \end{bmatrix}$$

(a) Form a new measurement pair, z_1' and z_2', as a linear combination of the original pair such that the errors in the new pair are uncorrelated. (*Hint:* First, let $z_1' = z_1$ and then assume $z_2' = c_1 z_1 + c_2 z_2$ and choose the constants c_1 and c_2 such that the new measurement errors are uncorrelated.)

(b) Find the $\mathbf{H}$ and $\mathbf{R}$ matrices associated with the new z' measurement vector.

6.3 The current model of the relaying application presented in Section 6.4 contained a nonstationary state variable with exponential properties. Assume that the initial value of this state variable is a random variable and has a variance that is 100 times larger than the steady-state variance of the process. Sketch three typical sample realizations of the process.

6.4 (a) Derive an explicit expression for $\mathbf{P}_{k+1}$ in terms of $\mathbf{P}_k$ and the $\boldsymbol{\phi}$, $\mathbf{H}$, $\mathbf{Q}$, $\mathbf{R}$ parameters. (The a priori error covariance matrix $\mathbf{P}^-$ and gain $\mathbf{K}$ should *not* appear in your final expression.)

(b) Find a similar difference equation for $\mathbf{P}_{k+1}^-$ in terms of $\mathbf{P}_k^-$ and the system parameters.

6.5 It was mentioned in the power system harmonics example (Section 6.5) that the gain sequences become periodic in the steady-state condition. Verify this by programming the 6-state Kalman filter model using the same parameter values given in Section 6.5. Let the Kalman filter cycle through enough steps to reach a steady-state condition. Plot the gain sequences for the fundamental, 3rd, and 5th harmonics on three separate graphs with the in-phase and quadrature gains superimposed on each graph. Note the quadrature relationship between the two gains for each harmonic.

6.6 Suboptimal filter analysis is often used in sensitivity analysis. This is where the analyst wishes to assess the sensitivity of the system error to changes in certain parameters. In the random walk example of Section 6.7 (Example 6.4), the true value of Q was said to be 1.0, the true R was .1, and the rms estimation error worked out to be about .302668.

(a) Now suppose the designer incorrectly models the filter to be used in this application using $Q = 1.1$ rather than the true value. The filter will then be suboptimal to some extent. Compute the filter rms error for this suboptimal situation.

(b) Suppose the designer errs in the other direction and chooses the filter Q to be 0.9 rather than the true value of 1.0. Find the filter rms error for this situation.

(c) The percent variation of Q on either side of truth in parts (a) and (b) is about 10 percent. What is the corresponding percent variation in the rms estimation error? Would you call this a high or low sensitivity situation?

(*Note:* Convergence to the steady-state condition is quite rapid in this example. Thus, the analysis can be carried out easily with a hand-held calculator.)

6.7 In the power systems harmonics application discussed in Section 6.5, it is envisioned that the numerical values of $\mathbf{Q}_k$ and $\mathbf{R}_k$ would be fixed, that is, the filter is not self-learning. The value assigned to the $\mathbf{R}_k$ will depend on the accuracy of the measurement equipment used in the field, and a fairly reliable value can be assigned to this parameter. The same is not so for $\mathbf{Q}_k$. This parameter is associated with the random variations of the harmonic content of the signal, and thus the value assigned to $\mathbf{Q}_k$ is bound to be more "fuzzy" than the value assigned to $\mathbf{R}_k$. Therefore, it is of interest to assess the filter's sensitivity to the $\mathbf{Q}_k$ parameter.

Assume that the numerical values used in Section 6.5 are the true parameter values. Now suppose that the q_{33} and q_{44} values (for the 3rd harmonic components) actually implemented in the on-line filter are greater than the true value by a factor of 4. Using the suboptimal methods discussed in Section 6.7, assess the effect of this mismodeling on the rms error in the estimates of the in-phase and quadrature components of the 3rd harmonics. On the basis of your investigation, would you say this is a low- or high-sensitivity situation?

6.8 The recursive process in Example 6.6 was carried through only one step. Continue the process through a second recursive step, and obtain an explicit expression for $\hat{x}$ in terms of z_1 and z_2. Next, compute the ordinary least-squares estimate of g using just two measurements z_1 and z_2. Do this on a batch basis using the equation

$$\hat{g} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{z}$$

where

$$\mathbf{H} = \begin{bmatrix} \frac{1}{2}t_1^2 \\ \frac{1}{2}t_2^2 \end{bmatrix} \quad \text{and} \quad \mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

Compare the least-squares result with that obtained after carrying the Kalman filter recursive process through two steps.

6.9 In Example 6.6 the initial position and velocity for the falling mass were assumed to be known, that is, they were exactly zero; the gravity constant was presumed to be unknown and to be estimated. Bozic (14) presents an interesting variation on this problem where the situation is reversed; g is assumed to be known perfectly and the initial position and velocity are random variables with known Gaussian distribution. Even though the trajectory obeys a known deterministic equation of motion, the random initial conditions add sufficient uncertainty to the motion to make the trajectory a legitimate random process. Assume that the initial position and velocity are normal random variables described by $N(0, \sigma_p^2)$ and $N(0, \sigma_v^2)$. Let state variables x_1 and x_2 be position and velocity measured downward, and let the measurements take place at uniform intervals Δt beginning at $t = 0$. The measurement error variance is R. Work out the key parameters for the Kalman filter model for this situation. That is, find Φ_k, $\mathbf{Q}_k$, $\mathbf{H}_k$, $\mathbf{R}_k$ and the initial $\hat{\mathbf{x}}_0^-$ and $\mathbf{P}_0^-$. (Note that a deterministic forcing function has to be accounted for in this example. The effect of this forcing function, though, will appear in projecting $\hat{\mathbf{x}}_{k+1}^-$, but not in the model parameters.)

6.10 The scalar Gauss–Markov model has been used extensively in both Chapters 5 and 6. A more general second-order Gauss–Markov model that requires two state variables is also used frequently in Kalman filter applications. This process has a damped harmonic autocorrelation function, and one special version of this type of process can be generated with the shaping filter shown in the accompanying figure. Note that three parameters describe this process completely: σ_y^2, the variance; ω_0, the undamped natural frequency; and ζ, the damping ratio.

Consider a second-order Gaussian process y as described in the figure with the following parameters:

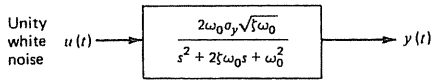
$$\sigma_y = 20 \text{ units}, \qquad \omega_0 = .02 \text{ rad/sec}, \qquad \zeta = .5$$

This will be referred to as the "signal" in our estimator. We wish now to look at a discrete Kalman filter where we have a direct one-to-one measurement of y corrupted by measurement noise. The measurement error in this case consists of an additive combination of a scalar Gauss–Markov process and a Gaussian white sequence. The parameters for the measurement noises are as follows:

Scalar Markov component: $\sigma_m = 10$ units, $\beta_m = .002$ rad/sec

White component: $\sigma_{\text{white}} = 10$ units

The sampling interval is 10 sec.



Problem 6.10

We wish to determine the steady-state rms estimator error for this situation. Using appropriate covariance analysis software, cycle the Kalman filter through the required number of steps to reach a steady-state condition. Do this two ways:

(a) First, initialize the filter $\mathbf{P}$ matrix by assuming that the initial state estimate is set equal to zero and that the system state is in a stationary condition when the filter begins to process measurements. (See Problem 5.9.)

(b) Next, arbitrarily initialize $\mathbf{P}_0^-$ to be zero (more precisely the null matrix). Then run the error covariance program to the steady-state condition and compare the result with that obtained in part (a). This might be thought of as the lazy way to solve the problem (but effective!).

6.11 The third-order Kalman filter of Problem 6.10 works out to be stable in the steady-state condition. (The gain matrix approaches a constant as the number of steps becomes large.) A method for finding the characteristic roots (eigenvalues) for a filter with constant gain was discussed in Section 6.9. Find the characteristic roots for the system of Problem 6.10 and comment on the number of steps required for the filter to reach a steady-state condition (say, within about 99 percent of the final value). Are the results of this problem consistent with the empirical results observed in part (b) of Problem 6.10?
(*Hint:* The MATLAB built-in function eig(A) returns the eigenvalues of the matrix A.)

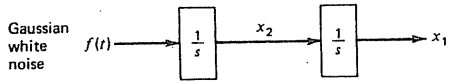
6.12 The accompanying block diagram shows two cascaded integrators driven by white noise. The two state variables x_1 and x_2 can be thought of as position and velocity for convenience, and the forcing function is acceleration. Let us suppose that we have a noisy measurement of velocity, but there is no direct observation of position. From linear control theory, we know that this system is not observable on the basis of velocity measurements alone. (This is also obvious from the ambiguity in initial position, given only the integral of velocity.) Clearly, there will be no divergence of the estimation error in x_2, because we have a direct measurement of it. However, divergence of the estimation error of x_1 is not so obvious.

The question of divergence of the error in estimating x_1 is easily answered empirically by cycling through the Kalman filter error covariance equations until either (a) a stationary condition for p_{11} is reached, or (b) divergence becomes obvious by continued growth of p_{11} with each recursive step. Perform the suggested experiment using appropriate covariance analysis software. You will find the following numerical values suitable for this exercise:

$$\text{Power spectral density of } f(t) = .1 \text{ (m/sec}^2\text{)}^2\text{/(rad/sec)}$$

$$\text{Step size } \Delta t = 1 \text{ sec}$$

$$\text{Measurement error variance} = .01 \text{ (m/sec)}^2$$



Problem 6.12

Note that if divergence is found, it is not the "fault" of the filter. It simply reflects an inadequate measurement situation. This should not be confused with *computational* divergence.

REFERENCES CITED IN CHAPTER 6

1. H. W. Sorenson (ed.), *Kalman Filtering: Theory and Application,* New York: IEEE Press, 1985.
2. J. S. Meditch, *Stochastic Optimal Linear Estimation and Control,* New York: McGraw-Hill, 1969.
3. A. Gelb (ed.), *Applied Optimal Estimation,* Cambridge, MA: MIT Press, 1974.
4. H. W. Sorenson, "Kalman Filtering Techniques," in C. T. Leondes (ed.), *Advances in Control Systems,* Vol. 3, New York: Academic Press, 1966.
5. R. P. Denaro, "Navstar: The All-Purpose Satellite," *IEEE Spectrum, 18*(5):35–40 (May 1981).
6. B. W. Parkinson and S. W. Gilbert, "NAVSTAR: Global Positioning System—Ten Years Later," *Proc. IEEE, 71*(10):1177–1186 (October 1983).
7. R. G. Brown, "Integrated Navigation Systems and Kalman Filtering: A Perspective," *Navigation, J. Inst. Navigation, 19*(4):335–362 (Winter 1972–73).
8. J. D. Salisbury, "Comments on Integrated Navigation Systems and Kalman Filtering: A Perspective," *Navigation, J. Inst. Navigation, 20*(2):190 (Summer 1973).
9. A. A. Girgis, "Application of Kalman Filtering in Computer Relaying of Power Systems," Ph.D. dissertation, Iowa State University, Ames, 1981.
10. A. A. Girgis and R. G. Brown, "Application of Kalman Filtering in Computer Relaying," *IEEE Trans. Power Apparatus Systs., PAS-100*(7):3387–3397 (July 1981).
11. H. W. Bode and C. E. Shannon, "A Simplified Derivation of Linear Least Squares Smoothing and Prediction Theory," *Proc. I.R.E., 38:*417–424 (April 1950).
12. H. W. Sorenson, "Least-Squares Estimation: From Gauss to Kalman," *IEEE Spectrum, 7:*63–68 (July 1970).
13. T. Kailath, "A View of Three Decades of Linear Filtering Theory," *IEEE Trans. Information Theory, IT-20*(2):146–181 (March 1974).
14. S. M. Bozic, *Digital and Kalman Filtering,* London: E. Arnold, Publisher, 1979.
15. *Global Positioning System,* Vol. IV, The Institute of Navigation, Alexandria, VA, 1993.
16. "Change No. 1 to RTCA/DO-208," RTCA paper no. 479-93/TMC-106, RTCA, Inc., Washington, DC, Sept. 21, 1993.
17. P. Y. C. Hwang, "Recommendation for Enhancement of RTCM-104 Differential Standard and Its Derivatives," *Proceedings of ION-GPS-93,* The Institute of Navigation, Sept. 22–24, 1993, pp. 1501–1508.
18. A. A. Girgis, W. B. Chang, and E. B. Makram, "A Digital Recursive Measurement Scheme for On-Line Tracking of Power System Harmonics," *IEEE Trans. Power Delivery, 6:*3, 1153–1160 (July 1991).
19. M. S. Grewal and A. P. Andrews, *Kalman Filtering Theory and Practice,* Englewood Cliffs, NJ: Prentice-Hall, 1993 (see Section 4.2 and Chapter 6).
20. A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing,* Englewood Cliffs, NJ: Prentice-Hall, 1989.

21. M. M. Sidar and B. F. Doolin, "On the Feasibility of Real-Time Prediction of Aircraft Carrier Motion at Sea," *IEEE Trans. Automatic Control, AC-28,* pp. 350–355 (March 1983). (Also reprinted in H. W. Sorenson, ed., *Kalman Filtering: Theory and Application,* New York: IEEE Press, 1985.)

Additional Reference on Applied Kalman Filtering

22. G. Minkler and J. Minkler, *Theory and Application of Kalman Filtering,* Palm Bay, FL: Magellan Book Co., 1993.

7

The Continuous Kalman Filter

About a year after his paper on discrete-data filtering, R. E. Kalman coauthored a second paper with R. S. Bucy on continuous filtering (1). This paper also proved to be a milestone in the area of optimal filtering. Our approach here will be somewhat different from theirs, in that we will derive the continuous filter equations as a limiting case of the discrete equations as the step size becomes small.* Philosophically, it is of interest to note that we begin with the discrete equations and then go to the continuous equations. So often in numerical procedures, we begin with the continuous dynamical equations; these are then discretized and the discrete equations become approximations of the continuous dynamics. Not so with the Kalman filter! The discrete equations are exact and stand in their own right, provided, of course, that the difference equation model of the process is exact and not an approximation.

The continuous Kalman filter is probably not as important in applications as the discrete filter, especially in real-time systems. However, the continuous filter is important for both conceptual and theoretical reasons, so this chapter will be devoted primarily to continuous filtering.

* One has to be careful in applying the methods of ordinary differential calculus to stochastic differential equations. Such methods are legitimate here only because we are dealing exclusively with linear dynamical systems [see, e.g., Jazwinski (2)]. It is worth noting that it is only the estimate equation that is stochastic. The error covariance equation (which is nonlinear) is deterministic. It depends only on the model parameters, which are not random and are assumed to be known.

7.1 TRANSITION FROM THE DISCRETE TO CONTINUOUS FILTER EQUATIONS

First, we assume the process and measurement models to be of the form:

$$\text{Process model:} \quad \dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{u} \tag{7.1.1}$$

$$\text{Measurement model:} \quad \mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{v} \tag{7.1.2}$$

where

$$E[\mathbf{u}(t)\mathbf{u}^T(\tau)] = \mathbf{Q}\delta(t - \tau) \tag{7.1.3}$$

$$E[\mathbf{v}(t)\mathbf{v}^T(\tau)] = \mathbf{R}\delta(t - \tau) \tag{7.1.4}$$

$$E[\mathbf{u}(t)\mathbf{v}^T(\tau)] = 0 \tag{7.1.5}$$

We note that in Eqs. (7.1.1) and (7.1.2), $\mathbf{F}$, $\mathbf{G}$, and $\mathbf{H}$ may be time-varying. Also, by analogy with the discrete model, we assume that $\mathbf{u}(t)$ and $\mathbf{v}(t)$ are vector white-noise processes with zero crosscorrelation. The covariance parameters $\mathbf{Q}$ and $\mathbf{R}$ play roles similar to $\mathbf{Q}_k$ and $\mathbf{R}_k$ in the discrete filter, but they do not have the same numerical values. The relationships between the corresponding discrete and continuous filter parameters will be derived presently.

Recall that for the discrete filter,

$$\mathbf{Q}_k = E[\mathbf{w}_k\mathbf{w}_k^T] \tag{7.1.6}$$

$$\mathbf{R}_k = E[\mathbf{v}_k\mathbf{v}_k^T] \tag{7.1.7}$$

To make the transition from the discrete to continuous case, we need the relations between $\mathbf{Q}_k$ and $\mathbf{R}_k$ and the corresponding $\mathbf{Q}$ and $\mathbf{R}$ for a small step size Δt. Looking at $\mathbf{Q}_k$ first and referring to Eq. (5.3.6), we note that $\boldsymbol{\phi} \approx \mathbf{I}$ for small Δt and thus

$$\mathbf{Q}_k \approx \int_{\text{small } \Delta t}\int \mathbf{G}(\xi)E[\mathbf{u}(\xi)\mathbf{u}^T(\eta)]\mathbf{G}^T(\eta)d\xi\, d\eta \tag{7.1.8}$$

Next, substituting Eq. (7.1.3) into (7.1.8) and integrating over the small Δt interval yield

$$\mathbf{Q}_k = \mathbf{G}\mathbf{Q}\mathbf{G}^T\,\Delta t \tag{7.1.9}$$

The derivation of the equation relating $\mathbf{R}_k$ and $\mathbf{R}$ is more subtle. In the continuous model $\mathbf{v}(t)$ is white, so simple sampling of $\mathbf{z}(t)$ leads to measurement noise with infinite variance. Hence, in the sampling process, we have to imagine averaging the continuous measurement over the Δt interval to get an equivalent

discrete sample. This is justified because $\mathbf{x}$ is not white and may be approximated as a constant over the interval. Thus, we have

$$\mathbf{z}_k = \frac{1}{\Delta t}\int_{t_{k-1}}^{t_k} \mathbf{z}(t)\,dt = \frac{1}{\Delta t}\int_{t_{k-1}}^{t_k} [\mathbf{H}\mathbf{x}(t) + \mathbf{v}(t)]\,dt$$

$$\approx \mathbf{H}\mathbf{x}_k + \frac{1}{\Delta t}\int_{t_{k-1}}^{t_k} \mathbf{v}(t)\,dt \tag{7.1.10}$$

The discrete-to-continuous equivalence is then

$$\mathbf{v}_k = \frac{1}{\Delta t}\int_{\text{small } \Delta t} \mathbf{v}(t)\,dt \tag{7.1.11}$$

From Eq. (7.1.7) we have

$$E[\mathbf{v}_k\mathbf{v}_k^T] = \mathbf{R}_k = \frac{1}{\Delta t^2}\int_{\text{small } \Delta t}\int E[\mathbf{v}(u)\mathbf{v}^T(v)]\,du\,dv \tag{7.1.12}$$

Substituting Eq. (7.1.4) into (7.1.12) and integrating yield the desired relationship

$$\mathbf{R}_k = \frac{\mathbf{R}}{\Delta t} \tag{7.1.13}$$

At first glance, it may seem strange to have the discrete measurement error approach ∞ as $\Delta t \to 0$. However, this is offset by the sampling rate becoming infinite at the same time.

In making the transition from the discrete to continuous case, we first note from the error covariance projection equation (i.e., $\mathbf{P}_{k+1}^- = \boldsymbol{\phi}_k\mathbf{P}_k\boldsymbol{\phi}_k^T + \mathbf{Q}_k$) that $\mathbf{P}_{k+1}^- \to \mathbf{P}_k$ as $\Delta t \to 0$. Thus, we do not need to distinguish between a priori and a posteriori $\mathbf{P}$ matrices in the continuous filter. We proceed with the derivation of the continuous gain expression. Recall that the discrete Kalman gain is given by (see Fig. 5.8)

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k)^{-1} \tag{7.1.14}$$

Using Eq. (7.1.13) and noting that $\mathbf{R}/\Delta t \gg \mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T$ lead to

$$\mathbf{K}_k = \mathbf{P}_k^-\mathbf{H}_k^T(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k + \mathbf{R}/\Delta t)^{-1} \approx \mathbf{P}_k^-\mathbf{H}_k^T\mathbf{R}^{-1}\,\Delta t$$

We can now drop the subscripts and the super minus on the right side and we obtain

$$\mathbf{K}_k = (\mathbf{P}\mathbf{H}^T\mathbf{R}^{-1})\,\Delta t \tag{7.1.15}$$

We *define* the continuous Kalman gain as the coefficient of Δt in Eq. (7.1.15), that is,

$$\mathbf{K} \triangleq \mathbf{P}\mathbf{H}^T\mathbf{R}^{-1} \tag{7.1.16}$$

Next, we look at the error covariance equation. From the projection and update equations (Fig. 5.9), we have

$$\begin{aligned}\mathbf{P}_{k+1}^- &= \boldsymbol{\phi}_k\mathbf{P}_k\boldsymbol{\phi}_k^T + \mathbf{Q}_k \\ &= \boldsymbol{\phi}_k(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-\boldsymbol{\phi}_k^T + \mathbf{Q}_k \\ &= \boldsymbol{\phi}_k\mathbf{P}_k^-\boldsymbol{\phi}_k^T - \boldsymbol{\phi}_k\mathbf{K}_k\mathbf{H}_k\mathbf{P}_k^-\boldsymbol{\phi}_k^T + \mathbf{Q}_k\end{aligned} \tag{7.1.17}$$

We now approximate $\boldsymbol{\phi}_k$ as $\mathbf{I} + \mathbf{F}\Delta t$ and note from Eq. (7.1.15) that $\mathbf{K}_k$ is of the order of Δt. After we neglect higher-order terms in Δt, Eq. (7.1.17) becomes

$$\mathbf{P}_{k+1}^- = \mathbf{P}_k^- + \mathbf{F}\mathbf{P}_k^-\,\Delta t + \mathbf{P}_k^-\mathbf{F}^T\,\Delta t - \mathbf{K}_k\mathbf{H}_k\mathbf{P}_k^- + \mathbf{Q}_k \tag{7.1.18}$$

We next substitute the expressions for $\mathbf{K}_k$ and $\mathbf{Q}_k$, Eqs. (7.1.15) and (7.1.9), and form the finite difference expression

$$\frac{\mathbf{P}_{k+1}^- - \mathbf{P}_k^-}{\Delta t} = \mathbf{F}\mathbf{P}_k^- + \mathbf{P}_k^-\mathbf{F}^T - \mathbf{P}_k^-\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}_k\mathbf{P}_k^- + \mathbf{G}\mathbf{Q}\mathbf{G}^T \tag{7.1.19}$$

Finally, passing to the limit as $\Delta t \to 0$ and dropping the subscripts and super minus lead to the matrix differential equation

$$\begin{aligned}\dot{\mathbf{P}} &= \mathbf{F}\mathbf{P} + \mathbf{P}\mathbf{F}^T - \mathbf{P}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{P} + \mathbf{G}\mathbf{Q}\mathbf{G}^T \\ \mathbf{P}(0) &= \mathbf{P}_0\end{aligned} \tag{7.1.20}$$

Next, consider the state estimation equation. Recall the discrete equation is

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-) \tag{7.1.21}$$

We now note that $\hat{\mathbf{x}}_k^- = \boldsymbol{\phi}_{k-1}\hat{\mathbf{x}}_{k-1}$. Thus, Eq. (7.1.21) can be written as

$$\hat{\mathbf{x}}_k = \boldsymbol{\phi}_{k-1}\hat{\mathbf{x}}_{k-1} + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\boldsymbol{\phi}_{k-1}\hat{\mathbf{x}}_{k-1}) \tag{7.1.22}$$

Again, we approximate $\boldsymbol{\phi}$ as $\mathbf{I} + \mathbf{F}\Delta t$. Then, neglecting higher-order terms in Δt and noting that $\mathbf{K}_k = \mathbf{K}\Delta t$ lead to

$$\hat{\mathbf{x}}_k - \hat{\mathbf{x}}_{k-1} = \mathbf{F}\hat{\mathbf{x}}_{k-1}\,\Delta t + \mathbf{K}\Delta t(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_{k-1}) \tag{7.1.23}$$

Finally, dividing by Δt, passing to the limit, and dropping the subscripts yield the differential equation

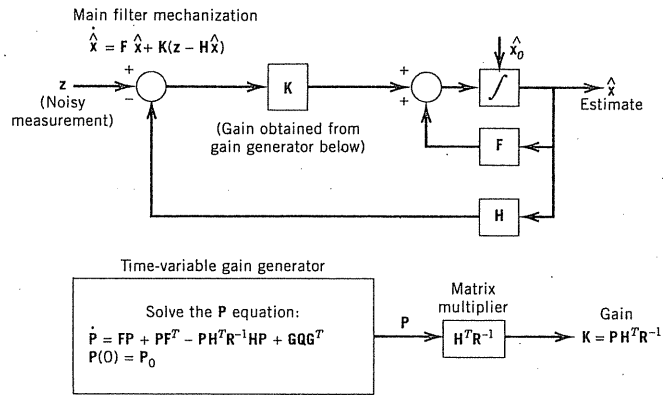


Figure 7.1 On-line block diagram for the continuous Kalman filter.

$$\dot{\hat{\mathbf{x}}} = \mathbf{F}\hat{\mathbf{x}} + \mathbf{K}(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}) \tag{7.1.24}$$

Equations (7.1.16), (7.1.20), and (7.1.24) comprise the continuous Kalman filter equations and these are summarized in Fig. 7.1. If the filter were to be implemented on-line, note that certain equations would have to be solved in real time as indicated in Fig. 7.1. Theoretically, the differential equation for $\mathbf{P}$ could be solved off-line, and the gain profile could be stored for later use on-line. However, the main $\hat{\mathbf{x}}$ equation must be solved on-line, because $\mathbf{z}(t)$, that is, the noisy measurement, is the input to the differential equation.

The continuous filter equations as summarized in Fig. 7.1 are innocent looking because they are written in matrix form. They should be treated with respect, though. It does not take much imagination to see the degree of complexity that results when they are written out in scalar form. If the dimensionality is high, an analog implementation is completely unwieldy.

Note that the error covariance equation must be solved in order to find the gain, just as in the discrete case. In the continuous case, though, a differential rather than difference equation must be solved. Furthermore, the differential equation is nonlinear because of the $\mathbf{P}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{P}$ term, which complicates matters. This will be explored further in the next section.

7.2 SOLUTION OF THE MATRIX RICCATI EQUATION

The error covariance equation

$$\begin{aligned}\dot{\mathbf{P}} &= \mathbf{F}\mathbf{P} + \mathbf{P}\mathbf{F}^T - \mathbf{P}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{P} + \mathbf{G}\mathbf{Q}\mathbf{G}^T \\ \mathbf{P}(0) &= \mathbf{P}_0\end{aligned} \tag{7.2.1}$$

is a special form of nonlinear differential equation known as the matrix Riccati equation. This equation has been studied extensively, and an analytical solution exists for the constant-parameter case. The general procedure is to transform the single nonlinear equation into a system of two simultaneous linear equations; of course, analytical solutions exist for linear differential equations with constant coefficients. Toward this end we assume that $\mathbf{P}$ can be written in product form as

$$\mathbf{P} = \mathbf{X}\mathbf{Z}^{-1}, \qquad \mathbf{Z}(0) = \mathbf{I} \tag{7.2.2}$$

or

$$\mathbf{P}\mathbf{Z} = \mathbf{X} \tag{7.2.3}$$

Differentiating both sides of Eq. (7.2.3) leads to

$$\dot{\mathbf{P}}\mathbf{Z} + \mathbf{P}\dot{\mathbf{Z}} = \dot{\mathbf{X}} \tag{7.2.4}$$

Next, we substitute $\dot{\mathbf{P}}$ from Eq. (7.2.1) into Eq. (7.2.4) and obtain

$$(\mathbf{P}\mathbf{F} + \mathbf{P}\mathbf{F}^T - \mathbf{P}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{P} + \mathbf{G}\mathbf{Q}\mathbf{G}^T)\mathbf{Z} + \mathbf{P}\dot{\mathbf{Z}} = \dot{\mathbf{X}} \tag{7.2.5}$$

Rearranging terms and noting that $\mathbf{P}\mathbf{Z} = \mathbf{X}$ lead to

$$\mathbf{P}(\mathbf{F}^T\mathbf{Z} - \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{X} + \dot{\mathbf{Z}}) + (\mathbf{F}\mathbf{X} + \mathbf{G}\mathbf{Q}\mathbf{G}^T\mathbf{Z} - \dot{\mathbf{X}}) = 0 \tag{7.2.6}$$

Note that if both terms in parentheses in Eq. (7.2.6) are set equal to zero, equality is satisfied. Thus, we have the pair of linear differential equations

$$\dot{\mathbf{X}} = \mathbf{F}\mathbf{X} + \mathbf{G}\mathbf{Q}\mathbf{G}^T\mathbf{Z} \tag{7.2.7}$$

$$\dot{\mathbf{Z}} = \mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{X} - \mathbf{F}^T\mathbf{Z} \tag{7.2.8}$$

with initial conditions

$$\mathbf{X}(0) = \mathbf{P}_0$$

$$\mathbf{Z}(0) = \mathbf{I} \tag{7.2.9}$$

These can now be solved by a variety of methods, including Laplace transforms. Once $\mathbf{P}$ is found, the gain $\mathbf{K}$ is obtained as $\mathbf{P}\mathbf{H}^T\mathbf{R}^{-1}$, and the filter parameters are determined. An example illustrates the procedure.

EXAMPLE 7.1

We now return to the continuous filter problem considered previously in Example 4.5. The problem was solved there using Wiener methods, and we now wish to apply Kalman filtering methods. Let the signal and noise be statistically independent with autocorrelation functions

$$R_s(\tau) = e^{-|\tau|} \qquad \left[\text{or } S_s(s) = \frac{2}{-s^2 + 1}\right] \tag{7.2.10}$$

$$R_n(\tau) = \delta(\tau) \qquad (\text{or } S_n = 1) \tag{7.2.11}$$

Since this is a one-state system, x is a scalar. Let x equal the signal. The additive measurement noise is white and thus no augmentation of the state vector is required. The process and measurement models are then

$$\dot{x} = -x + \sqrt{2}u, \qquad u = \text{unity white noise} \tag{7.2.12}$$

$$z = x + v, \qquad v = \text{unity white noise} \tag{7.2.13}$$

Thus, the system parameters are

$$F = -1, \qquad G = \sqrt{2}, \qquad Q = 1, \qquad R = 1 \qquad H = 1$$

The differential equations for X and Z are then

$$\dot{X} = -X + 2Z, \qquad X(0) = P_0$$

$$\dot{Z} = X + Z, \qquad Z(0) = 1 \tag{7.2.14}$$

Equations (7.2.14) may be solved readily using Laplace-transform techniques. The result is

$$X(t) = P_0 \cosh \sqrt{3}\, t + \frac{(2 - P_0)}{\sqrt{3}} \sinh \sqrt{3}\, t$$

$$Z(t) = \cosh \sqrt{3}\, t + \frac{(P_0 + 1)}{\sqrt{3}} \sinh \sqrt{3}\, t \tag{7.2.15}$$

The solution for P may now be formed as $P = XZ^{-1}$:

$$P = \frac{P_0 \cosh \sqrt{3}\, t + \dfrac{(2 - P_0)}{\sqrt{3}} \sinh \sqrt{3}\, t}{\cosh \sqrt{3}\, t + \dfrac{(P_0 + 1)}{\sqrt{3}} \sinh \sqrt{3}\, t} \tag{7.2.16}$$

Once P is found, the gain K is given by

$$K = PH^TR^{-1}$$

and the filter yielding $\hat{x}$ is determined.

Obviously, there should be a correspondence between this solution and the one obtained by Wiener methods. The general connection between the two methods will be discussed in more detail in Section 7.7; here we simply consider a limiting case check as $t \to \infty$. This should yield the same steady-state (stationary)

solution obtained previously with Wiener methods. Letting $t \to \infty$ in Eq. (7.2.16) and noting that $P_0 = 1$ yield

$$P \to \frac{1 \cdot e^{\sqrt{3}t} + \dfrac{2-1}{\sqrt{3}} e^{\sqrt{3}t}}{e^{\sqrt{3}t} + \dfrac{1+1}{\sqrt{3}} e^{\sqrt{3}t}} = \sqrt{3} - 1 \tag{7.2.17}$$

The Kalman filter block diagram for this example is then as shown in Fig. 7.2. This can be systematically reduced to yield the following overall transfer function relating $\hat{x}$ to z:

$$G(s) = \frac{\text{Laplace transform of } \hat{x}}{\text{Laplace transform of } z} = \frac{\sqrt{3}-1}{s+\sqrt{3}} \tag{7.2.18}$$

This is the same result obtained using Wiener methods. ■

7.3 CORRELATED MEASUREMENT AND PROCESS NOISE

Thus far we have considered the process noise $\mathbf{u}$ and the measurement noise $\mathbf{v}$ to have zero crosscorrelation. This is often a reasonable assumption in physical problems, but not always. We will now see how the filter equations can be modified to account for crosscorrelation between the process and measurement noises. We will consider the continuous filter problem here and defer the corresponding solution for the discrete filter until Chapter 9.

We first define the process and measurement models. They are

$$\left.\begin{aligned} \dot{\mathbf{x}} &= \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{u} \\ \mathbf{z} &= \mathbf{H}\mathbf{x} + \mathbf{v} \end{aligned}\right\} \quad \text{(as before)} \tag{7.3.1}$$

(7.3.2)

where

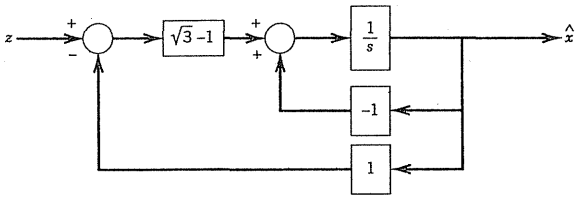


Figure 7.2 Stationary Kalman filter for Example 7.1.

$$\left.\begin{aligned} E[\mathbf{u}(t)\mathbf{u}^T(\tau)] &= \mathbf{Q}\delta(t-\tau) \\ E[\mathbf{v}(t)\mathbf{v}^T(\tau)] &= \mathbf{R}\delta(t-\tau) \end{aligned}\right\} \quad \text{(as before)} \tag{7.3.3}$$

and

$$E[\mathbf{u}(t)\mathbf{v}^T(\tau)] = \mathbf{C}\delta(t-\tau) \quad \text{(rather than zero, as before)} \tag{7.3.4}$$

Our general approach is to form a new process model whose input noise has zero crosscorrelation with $\mathbf{v}$. We first note that $\mathbf{z} - \mathbf{H}\mathbf{x} - \mathbf{v}$ is zero, and thus $\mathbf{D}(\mathbf{z} - \mathbf{H}\mathbf{x} - \mathbf{v})$ may be added to the right side of Eq. (7.3.1). Thus, we have

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{u} + \mathbf{D}(\mathbf{z} - \mathbf{H}\mathbf{x} - \mathbf{v}) \tag{7.3.5}$$

where the constant $\mathbf{D}$ will be chosen presently. However, first we rearrange the terms of Eq. (7.3.5) to obtain a new process model.

$$\text{New process model:} \quad \dot{\mathbf{x}} = (\mathbf{F} - \mathbf{D}\mathbf{H})\mathbf{x} + \mathbf{D}\mathbf{z} + (\mathbf{G}\mathbf{u} - \mathbf{D}\mathbf{v}) \tag{7.3.6}$$

The random process $\mathbf{x}$ is the same as before, but Eq. (7.3.6) shows that we can think of $\mathbf{x}$ as a superposition of two responses. One part is due to $\mathbf{D}\mathbf{z}(t)$, which we will treat as if it were a known explicit input. (It is observed and available directly as a measurement.) The other part of the response is due to $(\mathbf{G}\mathbf{u} - \mathbf{D}\mathbf{v})$, and this is a white-noise input that is not observed (directly, at least). For lack of better names, we will refer to these two component responses as the explicit and random parts, and they will be denoted as $\mathbf{x}_e$ and $\mathbf{x}_r$. The total response is then

$$\mathbf{x} = \mathbf{x}_e + \mathbf{x}_r \tag{7.3.7}$$

Now, in the process model, we wish to choose $\mathbf{D}$ such that $(\mathbf{G}\mathbf{u} - \mathbf{D}\mathbf{v})$ and $\mathbf{v}$ have zero crosscorrelation, that is

$$E\{[\mathbf{G}\mathbf{u}(t) - \mathbf{D}\mathbf{v}(t)]\mathbf{v}(\tau)^T\} = 0 \tag{7.3.8}$$

or

$$E[\mathbf{G}\mathbf{u}(t)\mathbf{v}(\tau)^T] = E[\mathbf{D}\mathbf{v}(t)\mathbf{v}(\tau)^T] \tag{7.3.9}$$

Next, substituting Eqs. (7.3.3) and (7.3.4) into Eq. (7.3.9) leads to

$$\mathbf{G}\mathbf{C}\delta(t-\tau) = \mathbf{D}\mathbf{R}\delta(t-\tau) \tag{7.3.10}$$

We now see that if we choose $\mathbf{D}$ to be

$$\mathbf{D} = \mathbf{G}\mathbf{C}\mathbf{R}^{-1} \tag{7.3.11}$$

the desired decorrelation is effected. The new process model now satisfies all

the necessary conditions imposed in Section 7.1, so the error covariance expression may be written as

$$\dot{\mathbf{P}} = (\mathbf{F} - \mathbf{DH})\mathbf{P} + \mathbf{P}(\mathbf{F} - \mathbf{DH})^T - \mathbf{PH}^T\mathbf{R}^{-1}\mathbf{HP} + \mathbf{Q}' \tag{7.3.12}$$

where $\mathbf{Q}'$ is defined by

$$E\{[\mathbf{Gu}(t) - \mathbf{Dv}(t)][\mathbf{Gu}(\tau) - \mathbf{DV}(\tau)]^T\} = \mathbf{Q}'\delta(t - \tau) \tag{7.3.13}$$

Expanding Eq. (7.3.13) and using Eqs. (7.3.3), (7.3.4), and (7.3.11) lead to

$$\mathbf{Q}' = \mathbf{G}(\mathbf{Q} - \mathbf{CR}^{-1}\mathbf{C}^T)\mathbf{G}^T \tag{7.3.14}$$

The equation for $\dot{\mathbf{P}}$ may now be rewritten as

$$\dot{\mathbf{P}} = (\mathbf{F} - \mathbf{DH})\mathbf{P} + \mathbf{P}(\mathbf{F} - \mathbf{DH})^T - \mathbf{PH}^T\mathbf{R}^{-1}\mathbf{HP} + \mathbf{G}(\mathbf{Q} - \mathbf{CR}^{-1}\mathbf{C}^T)\mathbf{G}^T \tag{7.3.15}$$

The expression for gain in the new model is then

$$\mathbf{K}' = \mathbf{PH}^T\mathbf{R}^{-1} \tag{7.3.16}$$

and the motive for using the prime will be made clear shortly.

We now look at the equation for the estimate. Just as in the case of the process itself, we wish to think of the estimate as a superposition of an explicit part, which is an estimate of $\mathbf{x}_e$, and another part, which is an estimate of the random component $\mathbf{x}_r$. Thus, we have for the two estimates

$$\dot{\hat{\mathbf{x}}}_e = (\mathbf{F} - \mathbf{DH})\hat{\mathbf{x}}_e + \mathbf{Dz}(t) \tag{7.3.17}$$

$$\dot{\hat{\mathbf{x}}}_r = (\mathbf{F} - \mathbf{DH})\hat{\mathbf{x}}_r + \mathbf{K}'(\mathbf{z}_r - \mathbf{H}\hat{\mathbf{x}}_r) \tag{7.3.18}$$

The error associated with $\hat{\mathbf{x}}_e$ is, of course, zero because $\mathbf{z}(t)$ is known, that is, in real time it is the total measurement and it is known exactly. The measurement of $\mathbf{x}_r$ has been denoted as $\mathbf{z}_r$, and this requires some explanation. From the basic measurement relationship, we have

$$\mathbf{z} = \mathbf{Hx} + \mathbf{v} = \mathbf{Hx}_e + \mathbf{Hx}_r + \mathbf{v}$$

or

$$(\mathbf{z} - \mathbf{Hx}_e) = \mathbf{Hx}_r + \mathbf{v} \tag{7.3.19}$$

Thus, $(\mathbf{z} - \mathbf{Hx}_e)$ must be considered as the noisy measurement of $\mathbf{x}_r$. This is denoted as $\mathbf{z}_r$ in Eq. (7.3.18). Thus, making the substitution into Eq. (7.3.18) leads to

$$\dot{\hat{\mathbf{x}}}_r = (\mathbf{F} - \mathbf{DH})\hat{\mathbf{x}}_r + \mathbf{K}'(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}_e - \mathbf{H}\hat{\mathbf{x}}_r) \tag{7.3.20}$$

Now, adding Eqs. (7.3.17) and (7.3.20), and noting that $\hat{\mathbf{x}} = \hat{\mathbf{x}}_e + \hat{\mathbf{x}}_r$ yield

$$\dot{\hat{\mathbf{x}}} = (\mathbf{F} - \mathbf{DH})\hat{\mathbf{x}} + (\mathbf{D} + \mathbf{K}')\mathbf{z} - \mathbf{K}'\mathbf{H}\hat{\mathbf{x}} = \mathbf{F}\hat{\mathbf{x}} + (\mathbf{D} + \mathbf{K}')(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}) \tag{7.3.21}$$

It can be seen from the form of Eq. (7.3.21) that $(\mathbf{D} + \mathbf{K}')$ plays the role of "gain" in the estimation equation for the total $\mathbf{x}$ quantity. The error associated with the $\hat{\mathbf{x}}_e$ component is zero, so the $\mathbf{P}$ equation, which was derived for the $\hat{\mathbf{x}}_r$ component, also applies to the total $\hat{\mathbf{x}}$. From Eqs. (7.3.11) and (7.3.16), we see that the gain $\mathbf{D} + \mathbf{K}'$ is just $(\mathbf{PH}^T + \mathbf{GC})\mathbf{R}^{-1}$. Thus, the final estimation equations for $\hat{\mathbf{x}}$ may be summarized as

1. ***Estimation equation:***

$$\dot{\hat{\mathbf{x}}} = \mathbf{F}\hat{\mathbf{x}} + \mathbf{K}(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}), \qquad \hat{\mathbf{x}}(0) = \mathbf{x}_0 \tag{7.3.22}$$

2. ***Gain equation:***

$$\mathbf{K} = (\mathbf{PH}^T + \mathbf{GC})\mathbf{R}^{-1} \tag{7.3.23}$$

3. ***Error covariance equation:***

$$\dot{\mathbf{P}} = (\mathbf{F} - \mathbf{DH})\mathbf{P} + \mathbf{P}(\mathbf{F} - \mathbf{DH})^T - \mathbf{PH}^T\mathbf{R}^{-1}\mathbf{HP} + \mathbf{G}(\mathbf{Q} - \mathbf{CR}^{-1}\mathbf{C}^T)\mathbf{G}^T \tag{7.3.24}$$
$$\mathbf{P}(0) = \mathbf{P}_0$$

where

$$\mathbf{D} = \mathbf{GCR}^{-1} \tag{7.3.25}$$

We note that if $\mathbf{C}$ is zero, the above estimation equations reduce to the previous equations where we had zero crosscorrelation between $\mathbf{u}$ and $\mathbf{v}$. This is as expected. Also, note that in the process of deriving the filter equations for the correlated case, it was necessary to consider a superposition of explicit and random responses. Since this can always be done in a linear system, the addition of an explicit or deterministic driving function to the process equation presents no particular problem. We simply treat it separately and note that it contributes zero error to the estimate (see Problem 7.8). An example illustrating the use of the equations for correlated $\mathbf{u}$ and $\mathbf{v}$ will be presented in Section 7.4.

7.4 COLORED MEASUREMENT NOISE

In the equations for the continuous Kalman filter, the inverse of the $\mathbf{R}$ matrix appears in both the gain and error covariance expressions. If the measurement

noise is colored, rather than white, it must be modeled with additional state variables. This leaves zero for the white component, and thus $\mathbf{R}^{-1}$ will not exist. This leads to obvious difficulty. Of course, one remedy would be to add a small white component artificially, and then proceed on with the filter design using the usual equations. However, this is begging the issue. There are legitimate situations where one simply does not want to model any part of the measurement error as white noise.

A general solution to the colored measurement noise problem was first presented by Bryson and Johansen (3). Our approach here will be slightly different and, we hope, more intuitively satisfying. If $\mathbf{R}^{-1}$ does not exist, we have a situation where certain linear combinations of the state variables are being measured perfectly. Obviously, if a particular state variable is known perfectly, it can be removed from the estimation problem. There is certainly no need for filtering something that is already free of corrupting noise. Therefore, our general approach will be to remove those quantities that are known perfectly from those that are not, and then solve the remaining estimation problem. This usually necessitates a linear transformation of the original state vector in order to decouple the known states from the others. This adds to the algebra, but it certainly presents no theoretical problem because the state estimate transforms with exactly the same transformation as the state itself.

EXAMPLE 7.2

Consider the problem of separating an additive combination of two independent Markov processes with identical spectral functions. We know the solution calls for a trivial filter with a gain function of $\frac{1}{2}$. Let the autocorrelation functions of the signal and noise be

$$R_s(\tau) = e^{-|\tau|}, \qquad R_n(\tau) = e^{-|\tau|} \tag{7.4.1}$$

The measurement z is the additive combination

$$z = s + n \tag{7.4.2}$$

Now the obvious way to model the continuous Kalman filter is to let s be x_1 and n be x_2, and then we have

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} \sqrt{2} & 0 \\ 0 & \sqrt{2} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{7.4.3}$$

$$[z] = [1 \quad 1] \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + [0] \tag{7.4.4}$$

where u_1 and u_2 are independent, unity white-noise driving functions. Notice that v is zero and thus R^{-1} does not exist. Since we have a perfect measurement of the linear combination $x_1 + x_2$, let us transform to new states, y_1 and y_2, according to

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{7.4.5}$$

or

$$\mathbf{y} = \mathbf{\Lambda x}$$

Making this transformation leads to the new state and measurement models

$$\begin{bmatrix} \dot{y}_1 \\ \dot{y}_2 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + \begin{bmatrix} \sqrt{2} & 0 \\ \sqrt{2} & \sqrt{2} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{7.4.6}$$

$$[z] = [0 \quad 1] \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} + [0] \tag{7.4.7}$$

Note that z is now a perfect estimate of y_2 alone, so that we need to worry only about estimating y_1. Thus, the order of the problem is reduced from 2 to 1.

Consider next the y_1 state equation

$$\dot{y}_1 = -y_1 + \sqrt{2}\, u_1 \tag{7.4.8}$$

This is in the correct form as it is. (Usually, y_2 would also appear, and if so, it would be treated as a *known* driving function because it is equal to $z(t)$.)

Next, we need a measurement relationship of the proper form, that is, there must be nontrivial additive white noise and some linear connection to y_1 (and y_1 alone). In this case, since z has zero additive white noise, we can consider its derivative $\dot{z}$ as also being available as a measurement.

$$\dot{z} = \dot{y}_2 = -y_2 + \sqrt{2}\, u_1 + \sqrt{2}\, u_2 \tag{7.4.9}$$

Observe that $\dot{z}$ contains additive white noise as required. However, it also involves y_2. In order to eliminate y_2, we note

$$z = y_2 \tag{7.4.10}$$

Now, add z and $\dot{z}$ to eliminate y_2. This leads to

$$\dot{z} + z = 0 \cdot y_1 + \sqrt{2}\,(u_1 + u_2) \tag{7.4.11}$$

We now have a linear connection between $\dot{z} + z$ and y_1 with additive white noise, which is in the correct form. In the new problem of estimating y_1, we then have

$$F = -1$$
$$G = \sqrt{2}$$
$$u = u_1$$
$$Q = 1$$
$$H = 0$$
$$v = \sqrt{2}\,(u_1 + u_2) \tag{7.4.12}$$

and $\dot{z} + z$ plays the role of the "measurement." (After all, if z is known, so is $\dot{z}$.)

Note that u and v are correlated, so we must use the correlated form of Kalman filter equations given in Section 7.3. Using the notation of Section 7.3, we then have the additional parameters

$$C = \sqrt{2}$$
$$R = 4$$
$$G(Q - CR^{-1}C^T)G^T = 1$$
$$D = GCR^{-1} = \tfrac{1}{2} \tag{7.4.13}$$

Therefore, the optimal filter is given by the differential equation

$$\dot{\hat{y}}_1 = -\hat{y}_1 + \tfrac{1}{2}[(\dot{z} + z) - 0 \cdot \hat{y}_1] \tag{7.4.14}$$

This will be recognized as the equivalent transfer function relationship

$$\frac{\hat{y}_1(s)}{z(s)} = \frac{\frac{1}{2}(s+1)}{(s+1)} = \frac{1}{2} \tag{7.4.15}$$

This checks with Wiener filter theory, that is,

$$\hat{y}_1 = \tfrac{1}{2}z \tag{7.4.16}$$

We now have optimal estimates of y_1 and y_2. All we have to do is transform back to the x domain to get optimal estimates of x_1 and x_2. This yields

$$\begin{bmatrix}\hat{x}_1\\ \hat{x}_2\end{bmatrix} = \begin{bmatrix}1 & 0\\ 1 & 1\end{bmatrix}^{-1}\begin{bmatrix}\hat{y}_1\\ \hat{y}_2\end{bmatrix} = \begin{bmatrix}1 & 0\\ -1 & 1\end{bmatrix}\begin{bmatrix}\frac{1}{2}z\\ z\end{bmatrix} = \begin{bmatrix}\frac{1}{2}z\\ \frac{1}{2}z\end{bmatrix} \tag{7.4.17}$$

Thus, the optimal estimates of both x_1 and x_2 check with the results of the Wiener theory. ■

The procedure for dealing with colored measurement noise may now be summarized as follows:

1. Decouple the quantities that are known perfectly from those that are not. Use a linear transformation and, as a matter of convenience, define the new state variables such that the perfectly known variables are at the "bottom" of the new state vector. The bottom elements should then have a one-to-one correspondence to the perfect measurements.
2. Consider a reduced state estimation problem where the state vector contains only the "top" elements of the new state vector. These are not known perfectly from the measurement information, so a nontrivial estimation problem exists relative to these variables. Note that the perfectly known bottom elements that appear on the right side of the state equation may be replaced with measurements z_1, z_2, etc. They may then be grouped with the driving functions because they are known quantities. The reduced state equations must fit the general form

$$\dot{\mathbf{x}}_T = \mathbf{F}_T\mathbf{x}_T + (\text{known inputs}) + (\text{white-noise inputs}) \tag{7.4.18}$$

   where the subscript T is used to indicate top elements of the transformed state vector.
3. Rearrange the measurement equation to achieve the desired form

$$\mathbf{z}_T = \mathbf{H}_T\mathbf{x}_T + (\text{white noise}) \tag{7.4.19}$$

   Note that a nontrivial white-noise component must be associated with each measurement element. Also, the number of elements in $\mathbf{z}_T$ must be the same as for the original $\mathbf{z}$ vector in order that no measurement information is ignored in estimating $\mathbf{x}_T$. The white-noise components are brought in by repeated differentiation of the perfect z elements until a white-noise term appears. The new $\mathbf{z}_T$ vector is then formed from appropriate linear combinations of the elements of $\mathbf{z}$ and their derivatives. Sometimes, considerable algebraic manipulation is needed to twist the equations into the form demanded by Eq. (7.4.19), but this can always be done.
4. Solve the reduced state estimation problem. Usually, the filter equations for correlated $\mathbf{u}$ and $\mathbf{v}$, as given in Section 7.3, will have to be used. The result will be a differential equation for the reduced state vector. This equation will normally contain derivatives of some of the measurements, as well as the measurements, as driving functions. The derivatives usually present no problems and may be left there as driving functions. If, however, one wishes to remove them, there are standard techniques in linear system theory for doing so [e.g., see Ogata (4) or Chen (5)]. The reduced state equation is usually solved (conceptually, at least) for either the transient or steady-state solution, whichever is desired. The end result is the best estimate of the reduced state vector (i.e., the imperfectly known elements) in the transformed domain.
5. Finally, append the perfectly known elements to the bottom of those estimated in step 4, and transform the total state estimate back to the

original state space. This gives the optimal estimate for the original problem.

7.5 SUBOPTIMAL ERROR ANALYSIS

Just as in the discrete case, an error covariance equation can be derived that is applicable to suboptimal as well as optimal gain. The derivation is similar to that given in Section 7.1, and the relationships between $\mathbf{Q}$ and $\mathbf{Q}_k$ and $\mathbf{R}$ and $\mathbf{R}_k$ are given there. We begin with the projection equation for the discrete filter

$$\mathbf{P}_{k+1}^- = \boldsymbol{\phi}_k \mathbf{P}_k \boldsymbol{\phi}_k^T + \mathbf{Q}_k \tag{7.5.1}$$

We now replace $\mathbf{P}_k$ with the general $\mathbf{P}$-update expression given by Eq. (5.4.11). This yields

$$\mathbf{P}_{k+1}^- = \boldsymbol{\phi}_k[(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T]\boldsymbol{\phi}_k^T + \mathbf{Q}_k \tag{7.5.2}$$

Next, the following substitutions are made in Eq. (7.5.2):

$$\begin{aligned} \boldsymbol{\phi}_k &= \mathbf{I} + \mathbf{F}\,\Delta t \\ \mathbf{K}_k &= \mathbf{K}\,\Delta t \\ \mathbf{Q}_k &= \mathbf{G}\mathbf{Q}\mathbf{G}^T\Delta t \\ \mathbf{R}_k &= \frac{\mathbf{R}}{\Delta t} \end{aligned} \tag{7.5.3}$$

Then, after we neglect higher-order terms in Δt, the expression for $\mathbf{P}_{k+1}^-$ becomes

$$\begin{aligned} \mathbf{P}_{k+1}^- = \mathbf{P}_k^- - \mathbf{K}\mathbf{H}_k\mathbf{P}_k^-\,\Delta t - \mathbf{P}_k^-\mathbf{H}_k^T\mathbf{K}^T\,\Delta t + \mathbf{K}\mathbf{R}\mathbf{K}^T\,\Delta t + \mathbf{F}\mathbf{P}_k^-\,\Delta t \\ + \mathbf{P}_k^-\mathbf{F}^T\,\Delta t + \mathbf{G}\mathbf{Q}\mathbf{G}^T\,\Delta t \end{aligned} \tag{7.5.4}$$

Finally, forming the difference $\mathbf{P}_{k+1}^- - \mathbf{P}_k^-$, dividing by Δt, and passing to the limit lead to

$$\dot{\mathbf{P}} = \mathbf{F}\mathbf{P} + \mathbf{P}\mathbf{F}^T - \mathbf{K}\mathbf{H}\mathbf{P} - \mathbf{P}\mathbf{H}^T\mathbf{K}^T + \mathbf{K}\mathbf{R}\mathbf{K}^T + \mathbf{G}\mathbf{Q}\mathbf{G}^T \tag{7.5.5}$$

This equation may now be used with any gain $\mathbf{K}$ and is useful in suboptimal error analysis. A word of caution is in order, though. Note that "gain" means the $\mathbf{K}$ coefficient in the differential equation

$$\dot{\hat{\mathbf{x}}} = \mathbf{F}\hat{\mathbf{x}} + \mathbf{K}(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}) \tag{7.5.6}$$

and the equations describing the suboptimal filter under consideration must be put into the proper format before Eq. (7.5.5) can be applied. Also note that there must be a correspondence of certain parameters in the suboptimal and truth models in order for $\mathbf{P}$ to be meaningful as the error covariance for the suboptimal filter (see Section 6.7).

7.6 FILTER STABILITY IN STEADY-STATE CONDITION

In many applications the Kalman filter will reach a steady-state condition (or at least quasi-steady-state) within a reasonable time after initialization. When this happens, the gain becomes constant. In the continuous case the filter then looks much the same as any other analog fixed-parameter filter except, perhaps, it may be of the multiple input–output variety rather than just a single input–output filter. Conceptually, the filter is simply a linear operator that processes a set of inputs (the measurements) and transforms them to a corresponding set of outputs (the state estimates). We know from classical filter theory that the characteristic poles of the filter (roots of the transfer function denominator) tell us much about the stability of the filter, so we will now extend this idea to the Kalman filter (in the steady state, of course). The stability analysis here is similar to that presented in Section 6.9 for the discrete filter, except that Laplace transforms are used rather than z-transforms.

We will go directly to the differential equation relating the input $\mathbf{z}(t)$ to the output $\hat{\mathbf{x}}(t)$. Equation (7.1.24) is repeated for convenience:

$$\dot{\hat{\mathbf{x}}} = \mathbf{F}\hat{\mathbf{x}} + \mathbf{K}(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}) \tag{7.6.1}$$

Now assume that $\mathbf{F}$, $\mathbf{K}$, and $\mathbf{H}$ are constant and take the Laplace transform of both sides of Eq. (7.6.1). (We will tacitly assume that the initial conditions are zero.) After we rearrange terms, this leads to

$$[s\mathbf{I} - (\mathbf{F} - \mathbf{K}\mathbf{H})]\hat{\mathbf{X}}(s) = \mathbf{K}\mathbf{Z}(s) \tag{7.6.2}$$

It can be seen that the bracketed term on the left side of Eq. (7.6.2) is the characteristic matrix of the system, and its determinant will be the characteristic polynomial of the system, that is,

$$\text{Characteristic polynomial} = |s\mathbf{I} - (\mathbf{F} - \mathbf{K}\mathbf{H})| \tag{7.6.3}$$

The roots of this polynomial are then the characteristic poles of the system. [Or, if you prefer, the eigenvalues of $(\mathbf{F} - \mathbf{K}\mathbf{H})$ are the characteristic eigenvalues of the system.]

A simple example will illustrate the use of Eq. (7.6.3). Refer to Example 7.1 and note that the steady-state gain is

$$K = PH^TR^{-1} = (\sqrt{3} - 1)(1)(1) = \sqrt{3} - 1$$

Also, in this example F and H are

$$F = -1, \qquad H = 1$$

Therefore, the characteristic polynomial is

$$s - [-1 - (\sqrt{3} - 1)(1)] = s + \sqrt{3}$$

The characteristic pole is seen to be at $-\sqrt{3}$ in the s-plane, indicating that the filter is stable. This is, of course, the same result that was arrived at by block diagram reduction in Example 7.1. However, it is important to note that we did this without any such block diagram manipulation. Remember that state-space block diagrams like the one shown in Fig. 7.1 are vector *time-domain* diagrams, and extracting the transfer function in the s-domain can be very complicated (if not virtually impossible) in higher-order systems. On the other hand, it is routine to find the eigenvalues of a square matrix via computer methods, even in high-order systems. [Note that MATLAB has a built-in function eig(A) that returns the eigenvalues of a square matrix A.]

7.7 RELATIONSHIP BETWEEN WIENER AND KALMAN FILTERS

It is appropriate here to pause and reflect on the connection between Wiener and Kalman filter theories. Both the Wiener and Kalman filters are minimum mean-square-error estimators, both require the same a priori information about the processes being estimated, and both yield identical estimates. We saw in Chapter 4 that the Wiener approach leads to an integral equation with the filter weighting function as the unknown. After solution of the integral equation, the weighting function then describes the relationship between input and output. On the other hand, the end result of continuous Kalman filter theory is a differential equation relating input and output. There must be equivalence, and books on linear system theory [e.g., Chen (5)] may be consulted for the details of how to get back and forth between the two descriptions. There are, however, certain subtleties about this particular problem that warrant some further amplification.

First, in the Wiener theory we related the filter response to the input via the superposition (convolution) integral, which was written in the form

$$\hat{x}(t) = \int_0^t g(\tau, t)z(t - \tau)\, d\tau \tag{7.7.1}$$

This form was convenient because the first argument of g has the significance of the "age" variable, and g tells us how the past values of the input are weighted to yield the present value of the output (6). The second argument of g (i.e., t) simply appears as a parameter that may be considered fixed in the optimization process. Frequently, though, books on linear system theory [e.g., see Chen (5)] write the superposition integral in the form

$$\hat{x}(t) = \int_0^t h(t, \tau)z(\tau)\, d\tau \tag{7.7.2}$$

(This was mentioned before in Chapter 4.) When $\hat{x}$ is written in the form of Eq. (7.7.2), $h(t, \tau)$ has a physical interpretation as the system response at time t to a unit impulse applied at time τ. By making a simple change of dummy variables in Eq. (7.7.1), we can obtain the relationship between g and h:

$$g(t - \tau, t) = h(t, \tau) \tag{7.7.3}$$

This is not a point of confusion in constant-parameter systems. It is, however, in time-variable systems, and we have to be watchful of this little detail in converting from a weighting-function description to a state-realization (differential equation) description.

In addition, the Wiener filter is always a single-output estimator. That is, we may have multiple inputs, but we always choose a single scalar process (usually called "signal") as the quantity being estimated. For example, consider the problem where we have one Markov process that we call signal s, and the additive noise consists of another Markov process n_1 plus a white component n_2. In Wiener theory, n_1 gets lumped with n_2 as the corrupting noise and the estimator yields just an estimate of s. On the other hand, with Kalman filtering, one models both s and n_1 as state variables, and the filter estimates both s and n_1 simultaneously (even though $\hat{n}_1$ may not be of interest). In effect, the Kalman filter does the work of two Wiener filters. Furthermore, the Kalman filter automatically provides information about the quality of the estimates (i.e., their mean-square errors) while doing the estimation. The Wiener filter does not provide this information, and one has to go to considerable extra effort to get mean-square error information. Thus, we have in the Kalman filter a group of estimators, all packaged into a single matrix algorithm. The price, of course, is extra computation effort. We often find in Kalman filtering that we are forced to carry along considerable excess baggage in order to obtain estimates of a few select quantities of interest.

Another subtlety that appears when comparing Wiener and Kalman methods has to do with the initial conditions. It may appear at first glance that we are allowed to choose $\hat{\mathbf{x}}_0$ and $\mathbf{P}_0$ as we wish in a Kalman filter, whereas no such explicit choice exists in the corresponding Wiener formulation. This apparent discrepancy exists because the Wiener filter output was written as a superposition integral, which tacitly implies zero initial conditions for the filter. This is justified, provided the input processes have zero mean and we have no prior information about the processes other than what was already assumed in modeling the various spectral functions. To get correspondence in the Kalman filter, we must choose $\hat{\mathbf{x}}_0$ to be zero, and after we have done this, $\mathbf{P}_0$ must be the covariance of the process itself. Thus, we really do not have as much legitimate choice in

the initial $\hat{\mathbf{x}}_0$ and $\mathbf{P}_0$ as it seems at first glance; at least this is true if we want the result to be optimal. The correspondence between the Kalman and Wiener solutions, properly initialized, is easily verified for simple cases, and Problems 7.3 and 7.4 are intended to demonstrate this.

In summary, it is not proper to say that a Kalman filter is either better or worse than a Wiener filter, because both yield identical results once they are realized. However, in the realization, either for analysis purposes or on-line, the Kalman approach clearly has two distinct advantages:

1. With one common matrix formulation, we can accommodate a large class of estimation problems with relatively complicated process and measurement relationships.
2. The recursive feature of the Kalman filter makes it readily adaptable to computer solutions. This is certainly of considerable practical importance. Solutions of relatively complex estimation problems are often quite feasible using Kalman filtering methods, whereas the same problems may be completely intractable using Wiener methods.

PROBLEMS

7.1 Consider a scalar process $x(t)$ that may be thought of as the result of integrating Gaussian white noise with a spectral amplitude of 16 units. Let us say the integrator is "zeroed" at $t = 0$, and thus we know $x(0) = 0$. (This is a Wiener process.) Let us further say that we have a continuous noisy measurement of x where the measurement noise is white, Gaussian, independent of x, and has a spectral amplitude of 4 units.

(a) Find the optimal Kalman filter for this situation. (Your solution may be left in the form of a differential equation, but all parameters of the differential equation are to be written out explicitly.)

(b) Find the optimal filter transfer function and the corresponding rms error for the steady-state condition.

7.2 The optimal estimator for a Markov signal plus additive white noise was given in Example 7.1. The gain worked out to be time variable, but it did approach a constant value of $(\sqrt{3} - 1)$ as $t \to \infty$. Investigate the effect of using this constant gain for all $t \geq 0$ relative to the filter rms error. Sketch plots of both the optimal and suboptimal rms errors for purposes of comparison.

7.3 Consider a stationary Gaussian Markov signal $x(t)$ whose autocorrelation function is

$$R_x(\tau) = 4e^{-|\tau|}$$

We make two noisy measurements of the signal, one at $t = 0$ and the other at $t = 1$, and we denote these as z_0 and z_1. Each discrete measurement is known to have an error variance of unity, and the errors are statistically independent. We have no prior knowledge of the signal other than its autocorrelation function as stated above.

(a) Using the methods of Section 4.7 (i.e., the weight factor approach), write an explicit expression for the optimal estimate of $x(1)$ in terms of z_0 and z_1.

(b) Repeat part (a) using discrete Kalman filtering and compare the result with that obtained in part (a).

(*Note:* The Wiener and Kalman estimates should be identical.)

7.4 The same estimation problem was used in both Examples 4.5 and 7.1. Show that the two estimators yield identical results for all $t > 0$ as well as for the steady-state condition.

[*Hint:* In the Wiener solution, $\hat{x}(t)$ can be written in terms of a convolution integral where the weighting function is known explicitly. First, convert the weighting function to impulsive-response form and then substitute the integral expression for $\hat{x}(t)$ into the differential equation describing the Kalman estimator and show that it satisfies the differential equation.]

7.5 A certain noisy measurement is known to be of the form

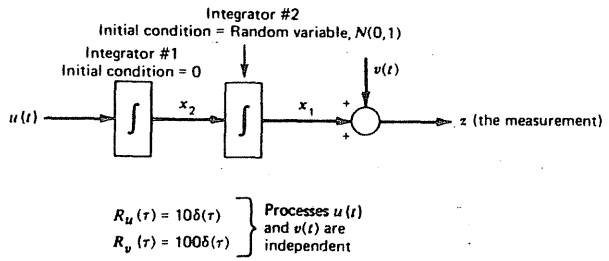
$$z(t) = a_0 + n(t), \qquad t \geq 0$$

where a_0 is an unknown random constant and $n(t)$ is white Gaussian noise with a spectral amplitude of A. Initially, at $t = 0$, all that is known about a_0 is that it has a zero-mean normal distribution with a variance σ^2. We wish to estimate a_0 on a "running time" basis beginning at $t = 0$. Find the appropriate continuous Kalman filter for estimating a_0 and sketch a block diagram for the filter.

(*Note:* This is the same estimation problem as that of Problem 4.11. However, when using Kalman filtering, we do not need to consider a random constant as a limiting case of a Markov process.)

7.6 The resulting optimal filter for Problem 7.5 is described by a differential equation relating the input $z(t)$ to the output $\hat{x}(t)$. When the same estimation problem is solved with Wiener methods, the result is a weighting function relating $z(t)$ and $\hat{x}(t)$. Show that the two results are equivalent.

7.7 Consider the two-state Gaussian random process shown in the block diagram below. Clearly, the measurement is of state variable 1, and it may be assumed that the measurement begins at $t = 0$. The initial condition on the first integrator (state variable 2) is zero, whereas the second integrator has an initial condition that is a zero-mean normal random variable with $\sigma^2 = 1$. Write the differential equation for the $\mathbf{P}$ matrix for the continuous Kalman filter for this



Problem 7.7.

situation. Be sure to specify the appropriate initial conditions for $\mathbf{P}$, and also be sure to specify all elements of the matrix parameters (such as $\mathbf{F}$, $\mathbf{R}$, and $\mathbf{Q}$) of the differential equation. You do not need to solve the differential equation.

7.8 Modification of the discrete Kalman filter equations to include a deterministic input was discussed in Section 6.10. Show that the corresponding modification in the continuous case is accomplished with the addition of $\mathbf{Bu}_d$ as a driving function in the $\hat{\mathbf{x}}$ differential equation. Specifically, assume the process equation is of the form

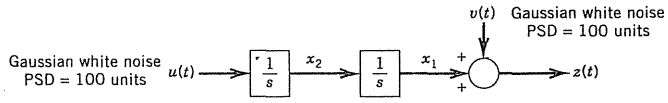
$$\dot{\mathbf{x}} = \mathbf{Fx} + \mathbf{Gu} + \mathbf{Bu}_d \qquad \text{(P7.8.1)}$$

where $\mathbf{Bu}_d$ is the deterministic input. Then show that the corresponding differential equation for the estimate is

$$\dot{\hat{\mathbf{x}}} = \mathbf{F}\hat{\mathbf{x}} + \mathbf{K}(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}) + \mathbf{Bu}_d \qquad \text{(P7.8.2)}$$

7.9 The accompanying figure shows a continuous random process model consisting of two integrators in cascade driven by white noise. The noisy observable is the output of the second integrator corrupted by additive white noise. This system is observable, so one would expect the Kalman filter error covariance equation to reach a steady-state condition after a suitable settling time.

(a) Find the characteristic poles for this Kalman filter in the steady-state condition.



Problem 7.9.

[*Hint:* The dynamics for this process are relatively simple, so it is feasible to obtain an explicit solution for $\mathbf{P}$. All one has to do is let $\dot{\mathbf{P}} = \mathbf{0}$ in the Riccati equation, Eq. (7.2.1), and then solve for the components of $\mathbf{P}$ algebraically. This, in turn, makes it possible to find the steady-state gain from Eq. (7.1.16).]

(b) Consider a similar discrete Kalman filter problem (but not exactly the same) where the two-state process is the same, the sampling interval is .5 sec, and $\mathbf{R}_k = 200$ units (obtained from Eq. 7.1.13). Find the characteristic poles for the discrete Kalman filter in the steady-state condition. (See Section 6.9 for help on this part.) You should find a rough correspondence (but not exact) between the results of this part and those of part (a).

7.10 The matrix Riccati equation, Eq. (7.2.1), describes the estimation error in a continuous Kalman filter. Note that setting $\mathbf{R}^{-1} = 0$ corresponds to no measurement. In this case the best estimate of the state is zero, and $\mathbf{P}(t)$ is just the covariance of the $\mathbf{x}$ process beginning at $t = 0$. If we let the initial value of $\mathbf{P}(t)$ be zero, we then obtain the differential equation for the $\mathbf{Q}_k$ matrix that was discussed in Chapter 5. That is, with the step size Δt as the argument, we have

$$\dot{\mathbf{Q}}_k = \mathbf{FQ}_k + \mathbf{Q}_k\mathbf{F}^T + \mathbf{GQG}^T, \qquad \mathbf{Q}_k(0) = 0 \qquad \text{(P7.10.1)}$$

where $\mathbf{GQG}^T$ is the power spectral density associated with the vector white-noise input $\mathbf{Gu}$. For the scalar Markov process discussed in Example 5.8, $\mathbf{Q}_k$ was found to be $(1 - e^{-2\Delta t})$. Show that the solution of the above differential equation, Eq. (P7.10.1), yields the same result.

REFERENCES CITED IN CHAPTER

1. R. E. Kalman and R. S. Bucy, "New Results in Linear Filtering and Prediction," *Trans. ASME, J. Basic Engr., 83:* 95–108 (1961).
2. A. H. Jazwinski, *Stochastic Processes and Filtering Theory,* New York: Academic Press, 1970.
3. A. E. Bryson, Jr. and D. E. Johansen, "Linear Filtering for Time-Varying Systems Using Measurements Containing Colored Noise," *IEEE Trans. Automatic Control, AC-10* (1): 4–10 (Jan. 1965).
4. K. Ogata, *State Space Analysis of Control Systems,* Englewood Cliffs, NJ: Prentice-Hall, 1967.
5. C. T. Chen, *Linear System Theory and Design,* New York: Holt, Rinehart, and Winston, 1984.
6. R. G. Brown and J. W. Nilsson, *Introduction to Linear Systems Analysis,* New York: Wiley, 1962.

Additional References on Continuous Kalman Filtering

7. A. Gelb (ed.), *Applied Optimal Estimation,* Cambridge, MA: MIT Press, 1974.
8. A. P. Sage and J. L. Melsa, *Estimation Theory with Applications to Communications and Control,* New York: McGraw-Hill, 1971.
9. J. S. Meditch, *Stochastic Optimal Linear Estimation and Control,* New York: McGraw-Hill, 1969.
10. P. S. Maybeck, *Stochastic Models, Estimation and Control* (Vol. 1), New York: Academic Press, 1979.
11. S. M. Bozic, *Digital and Kalman Filtering,* London: E. Arnold, Publisher, 1979.
12. M. S. Grewal and A. P. Andrews, *Kalman Filtering Theory and Practice,* Englewood Cliffs, NJ: Prentice-Hall, 1993.
13. G. Minkler and J. Minkler, *Theory and Application of Kalman Filtering,* Palm Bay, FL: Magellan Book Co., 1993.

8

Smoothing

The discrete Kalman *filtering* algorithm was presented in Chapter 5. *Prediction* was then discussed in Chapter 6. We will now consider the recursive *smoothing* problem. Recall from Chapter 4 that the smoothing problem is where we wish to form the optimal estimate at some point back in the past, relative to the current measurement. We will first classify the different types of smoothing problems and then proceed to the recursive solutions for each of the types.

8.1 CLASSIFICATION OF SMOOTHING PROBLEMS

Smoothing seems to be inherently more difficult than either filtering or prediction. In the Wiener theory of Chapter 4, α is negative in Eq. (4.3.22) and this causes a cusp to appear in the positive-time part of the shifted time function. (At least this is true for rational spectral functions—see Problem 8.1.) This cusp, in turn, leads to a considerably more complicated expression for the Fourier transform than occurs in the corresponding filter or prediction problems. Similarly, we will see that the recursive algorithms for smoothing are considerably more complicated than those for filtering and prediction.

Meditch (1) classifies smoothing into three categories:

1. ***Fixed-interval smoothing.*** Here the time interval of measurements (i.e., the data span) is fixed, and we seek optimal estimates at some, or perhaps all, interior points. This is the typical problem encountered when processing noisy measurement data off-line.
2. ***Fixed-point smoothing.*** In this case, we seek an estimate at a single fixed point in time, and we think of the measurements continuing on indefinitely ahead of the point estimation. An example of this would be the estimation of initial conditions based on noisy trajectory observations

after $t = 0$. In fixed-point smoothing there is no loss of generality in letting the fixed point be at the beginning of the data stream, because all prior measurements can be processed with the filter algorithm.

3. ***Fixed-lag smoothing.*** In this problem, we again envision the measurement information proceeding on indefinitely with the running time variable t, and we seek an optimal estimate of the process at a fixed length of time back in the past. Clearly, the Wiener problem with α negative is fixed-lag smoothing. It is of interest to note that the Wiener formulation will not accommodate either fixed-interval or fixed-point smoothing without using multiple sweeps through the same data with different values of α. This would be a most awkward way to process measurement data.

Obviously, the three smoothing problems are related, and it is not especially difficult to devise correct, but clumsy, solutions. Thus, the central problem is one of finding reasonably efficient recursive algorithms for each of the types of smoothing. This has been studied extensively since the early 1960s, and we will present only those solutions that are generally considered to be best from a computational viewpoint. We will not attempt to derive all the algorithms. The derivations are adequately documented in the references cited. (Also, see Problem 8.2.) We now proceed to look at algorithms for the three specified categories of smoothing.

8.2 DISCRETE FIXED-INTERVAL SMOOTHING

The algorithm to be presented here is due to Rauch, Tung, and Striebel (2, 3), and its derivation is given in Meditch (1) as well as the referenced papers. Also, a new simplified derivation is presented in Problem 8.2. In the interest of brevity, the algorithm will be subsequently referred to as the RTS algorithm. Consider a fixed-length interval containing $N + 1$ measurements. These will be indexed in ascending order $\mathbf{z}_0, \mathbf{z}_1, \ldots, \mathbf{z}_N$. The assumptions relative to the process and measurement models are the same as for the filter problem. The computational procedure for the RTS algorithm consists of a forward recursive sweep followed by a backward sweep. This is illustrated in Fig. 8.1. We enter the algorithm as usual at $k = 0$ with the initial conditions $\hat{\mathbf{x}}_0^-$ and $\mathbf{P}_0^-$. We then sweep forward using the conventional filter algorithm. With each step of the forward sweep, we must save the computed a priori and a posteriori estimates and their associated $\mathbf{P}$ matrices. These are needed for the backward sweep. After completing the forward sweep, we begin the backward sweep with "initial" conditions $\hat{\mathbf{x}}(N|N)$ and $\mathbf{P}(N|N)$ obtained as the final computation in the forward sweep.* With each

* The notation used here is the same as that used in the prediction problem. See Chapter 6, Section 6.1.

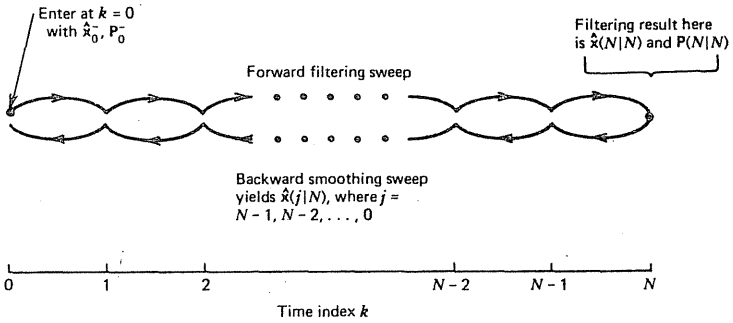


Figure 8.1 Procedure for fixed-interval smoothing.

step of the backward sweep, the old *filter* estimate is updated to yield an improved smoothed estimate, which is based on all the measurement data. The recursive equations for the backward sweep are

$$\hat{\mathbf{x}}(k|N) = \hat{\mathbf{x}}(k|k) + \mathbf{A}(k)[\hat{\mathbf{x}}(k+1|N) - \hat{\mathbf{x}}(k+1|k)] \tag{8.2.1}$$

where the smoothing gain $\mathbf{A}(k)$ is given by

$$\mathbf{A}(k) = \mathbf{P}(k|k)\boldsymbol{\Phi}^T(k+1, k)\mathbf{P}^{-1}(k+1|k) \tag{8.2.2}$$

and

$$k = N-1, N-2, \ldots, 0$$

The error covariance matrix for the smoothed estimates is given by the recursive equation

$$\mathbf{P}(k|N) = \mathbf{P}(k|k) + \mathbf{A}(k)[\mathbf{P}(k+1|N) - \mathbf{P}(k+1|k)]\mathbf{A}^T(k) \tag{8.2.3}$$

It is of interest to note that the *smoothing* error covariance matrix is not needed for the computation of the estimates in the backward sweep. This is in contrast to the situation in the filter (forward) sweep where the **P**-matrix sequence is needed for the gain and associated estimate computations. An example illustrating the use of the RTS algorithm is now in order.

EXAMPLE 8.1

Let us consider the same Gauss–Markov process used previously in Example 5.8, Section 5.6. Recall that the process has an autocorrelation function

$$R_x(\tau) = 1 \cdot e^{-|\tau|}$$

and we have a sequence of noisy measurements of this process taken every .02 sec. The measurement sequence begins at $t = 0$ and ends at $t = 1$ sec giving a

total of 51 discrete measurements. The measurement errors are white and have unity variance. A sample of this situation was simulated using Gaussian random numbers, and the filtering results were described previously, in Section 5.6. We now continue this same example with a backward sweep and obtain smoothed estimates.

A partial listing of the results from the filtering solution is repeated in Table 8.1 for reference purposes. It can be seen from the error covariances that the filter has reached a steady-state condition by the end of the forward sweep.

We enter the backward sweep at the end point where $k = 50$. Here we have

$$\hat{x}(50|50) = -.539 \tag{8.2.4}$$

$$P(50|50) = .1653 \tag{8.2.5}$$

Since the filter solution at this point is conditioned on *all* the measurement data, it is also the smoothed estimate at $k = 50$. We are now ready to compute the smoothed estimate one step back at $k = 49$. From Eqs. (8.2.1) and (8.2.2) we have

$$\hat{x}(49|50) = \hat{x}(49|49) + A(49)[\hat{x}(50|50) - \hat{x}(50|49)] \tag{8.2.6}$$

where

$$A(49) = P(49|49)\phi P^{-1}(50|49) \tag{8.2.7}$$

Using the filtering results from Table 8.1 and noting that the transition matrix is $e^{-.02} \approx .98020$ lead to

$$A(49) = .8183$$

$$\hat{x}(49|50) = -.525$$

This may now be repeated at $k = 48, 47, \ldots$, etc., with the results

Table 8.1 Filtering Results

k	$\hat{x}(k\|k)$	$P(k\|k)$	$\hat{x}(k\|k-1)$	$P(k\|k-1)$	$z(k)$
⋮	⋮	⋮	⋮	⋮	⋮
46	−.615	.1653	−.577	.1980	−.811
47	−.359	.1653	−.603	.1980	.871
48	−.511	.1653	−.352	.1980	−1.310
49	−.428	.1653	−.501	.1980	−.064
50	−.539	.1653	−.420	.1980	−1.138

$$\hat{x}(48|50) = -.531$$
$$\hat{x}(47|50) = -.506$$
$$\hat{x}(46|50) = -.536$$
$$\vdots$$

etc.

The smoothing results for the entire 50 steps are summarized in Fig. 8.2 along with the true x for comparison. The smoothed estimate plot can also be compared with the corresponding filter estimates shown in Fig. 5.11. It is evident from the plots that the smoothed plot is, in fact, smoother than the filter plot. This is as expected, because the smoother uses both past and future data in the makeup of its estimate. The smoothed estimate is not conspicuously better than the filter estimate for this simulation. However, we should not expect to be able to draw firm conclusions from such a small sample.

It is also of interest in this example to compare the error variances of the filter and smoothed estimates. These are shown in Fig. 8.3. Both the filter and smoother converge to a steady-state condition after about 15 steps. The steady-state values are

$$P_{\text{filter}} \approx .1653 \quad (\sigma_{\text{filter}} = .406)$$
$$P_{\text{smoother}} \approx .099 \quad (\sigma_{\text{smoother}} = .315)$$

Note that when the comparison is made on the basis of rms value, there is not a dramatic difference in the errors. In addition, the smoothed error curve is minimum in the middle and then gets larger as either end point is approached.

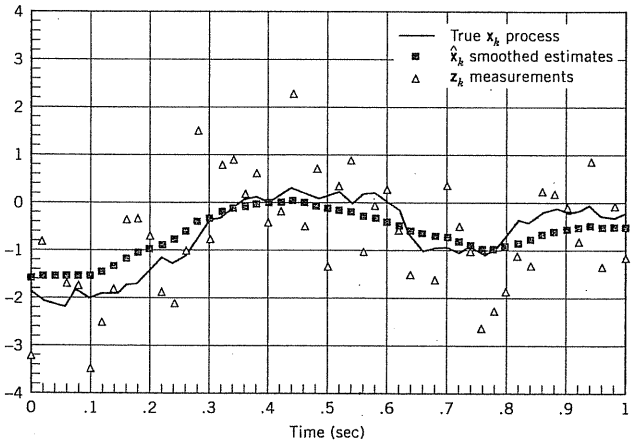


Figure 8.2 Smoothed estimate and true x-plots for Example 8.1.

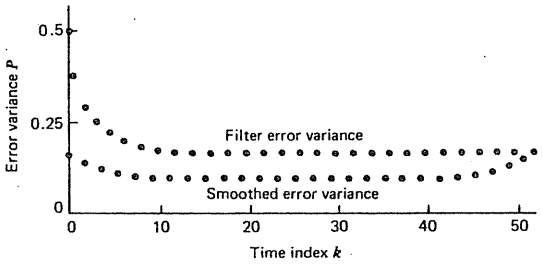


Figure 8.3 Sketch of error variances for filter and smoothed estimates of Example 8.1.

This indicates that the best situation for estimation occurs in the interior region where there is an abundance of measurement data in either direction from the point of estimation. This is exactly what we would expect intuitively.

Before we leave this example, the steady-state behavior in the middle of the variance plot of Fig. 8.3 should be noted. This frequently happens when the data span is large and the process is stationary and observable. In the steady-stage region, the filter and smoother gains and error covariance matrices are constant, and considerable computational simplification occurs, both in terms of the number of arithmetic operations and the amount of storage required for the algorithm. Thus, sometimes a problem that appears to be quite formidable at first glance works out to be feasible because a large amount of the measurement data can be processed in the steady-state region.

For future reference purposes, a partial listing of the fixed-interval smoothing solution is given in Table 8.2. ■

8.3 DISCRETE FIXED-POINT SMOOTHING

The fixed-point smoothing algorithm to be presented here is taken directly from Meditch (1). Alternative algorithms that, under certain circumstances, may be better computationally are also given by Meditch. In order to keep the discussion here as simple as possible, we present just one algorithm, which was chosen because of its simplicity and similarity to the fixed-interval algorithm. The algorithm is

$$\hat{\mathbf{x}}(k|j) = \hat{\mathbf{x}}(k|j-1) + \mathbf{B}(j)[\hat{\mathbf{x}}(j|j) - \hat{\mathbf{x}}(j|j-1)] \tag{8.3.1}$$

where

$$k \text{ is fixed} \quad (\text{usually } k = 0)$$
$$j = k+1, k+2, \text{ etc.}$$

and

Table 8.2 Fixed-Interval Smoothing Solution

k	$\hat{x}(k\|50)$	$P(k\|50)$
0	−1.592	.1653
1	−1.559	.1434
2	−1.556	.1287
3	−1.556	.1189
4	−1.552	.1123
5	−1.540	.1079
⋮	⋮	⋮
25	−.123	.0990
⋮	⋮	⋮
45	−.555	.1079
46	−.536	.1123
47	−.506	.1189
48	−.531	.1287
49	−.525	.1434
50	−.539	.1653

$$\mathbf{B}(j) = \prod_{i=k}^{j-1} \mathbf{A}(i) \tag{8.3.2}$$

$$\mathbf{A}(i) = \mathbf{P}(i|i)\boldsymbol{\phi}^T(i+1|i)\mathbf{P}^{-1}(i+1|i) \tag{8.3.3}$$

(Note that the equation for $\mathbf{A}$ is the same as for the fixed-interval algorithm. The error covariance associated with the smoothed estimate is given by

$$\mathbf{P}(k|j) = \mathbf{P}(k|j-1) + \mathbf{B}(j)[\mathbf{P}(j|j) - \mathbf{P}(j|j-1)]\mathbf{B}^T(j) \tag{8.3.4}$$

Note that the filter estimates and their error covariances are needed in this algorithm, just as in the fixed-interval case. The computational procedure is summarized in Fig. 8.4. We enter the algorithm at the beginning of the data stream with the usual a priori $\hat{\mathbf{x}}_0^-$ and $\mathbf{P}_0^-$. (We have let $k = 0$.) These initial parameters may come from prior knowledge of the process in the event there are no prior measurements, or $\hat{\mathbf{x}}_0^-$ and $\mathbf{P}_0^-$ may come from processing previous data via the filter algorithm. In either event, the first step is to update the a priori $\hat{\mathbf{x}}_0^-$ and $\mathbf{P}_0^-$ with the measurement $\mathbf{z}_0$ and obtain $\hat{\mathbf{x}}(0|0)$ and $\mathbf{P}(0|0)$. This is done with the usual filter equations and, of course, the result can be interpreted as either a filter estimate or the zero-stage smoothed estimate—they are one and the same. We next let $j = 1$, and we are ready to solve the one-stage smoothing

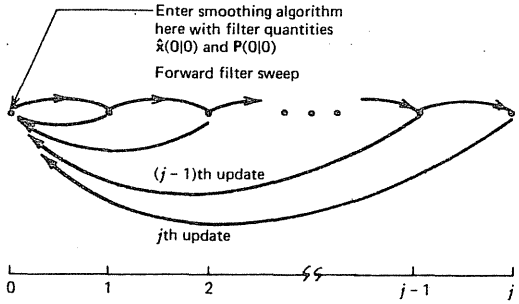


Figure 8.4 Procedure for fixed-point smoothing.

problem using Eqs. (8.3.1) to (8.3.4). This may then be continued indefinitely, theoretically, at least. Of course, as with any recursive algorithm, roundoff error may eventually lead to difficulties. We now look at an example of fixed-point smoothing.

EXAMPLE 8.2

To illustrate the application of the fixed-point algorithm, we use the same process and simulated data used in Example 8.1. We let the fixed point be at $k = 0$ and the data for the first 5 steps are given in Table 8.3. Since the filtering results are needed for the solution of the smoothing problem, they are also listed in the table.

The first step is to compute $\hat{x}(0|0)$ and $P(0|0)$. This was done in the previous filter solution with the result

$$\hat{x}(0|0) = -1.610$$
$$P(0|0) = .5$$

We are now ready to find $\hat{x}(0|1)$ by letting $k = 0$ and $j = 1$ in Eqs. (8.3.1), (8.3.2), and (8.3.3). The results are

Table 8.3 Data for Fixed-Point Example

j	$\hat{x}(j\|j)$	$\hat{x}(j\|j-1)$	$A(j)$	$B(j)$	$\hat{x}(0\|j)$	$P(j\|j)$	$P(j\|j-1)$	$P(0\|j)$
0	−1.610		.9432		−1.610	.5		.5
1	−1.317	−1.578	.9114	.9432	−1.364	.3419	.5196	.3419
2	−1.350	−1.291	.8860	.8596	−1.414	.2689	.3677	.2689
3	−1.406	−1.323	.8661	.7616	−1.477	.2293	.2975	.2293
4	−1.454	−1.378	.8513	.6597	−1.528	.2060	.2595	.2060
5	−1.815	−1.425	.8411	.5616	−1.747	.1917	.2372	.1917

$$B(1) = A(0) = P(0|0)\phi P^{-1}(1|0)$$
$$= .9432$$
$$\hat{x}(0|1) = \hat{x}(0|0) + B(1)[\hat{x}(1|1) - \hat{x}(1|0)]$$
$$= -1.364$$

We now have the needed information to go on to the next step and compute $\hat{x}(0|2)$, and so on. Note that it is not necessary to compute the smoothed error covariance in order to find the smoothed estimates. It was computed for checking purposes, though, and is listed in Table 8.3. In this example the fixed-point estimate converges nicely in 50 steps, and for $j = 50$ the solution is the same as that obtained from the fixed-interval algorithm. (See the $k = 0$ entry in Table 8.2.) ■

8.4 FIXED-LAG SMOOTHING

The fixed-lag smoothing problem was originally solved by Wiener in the 1940s. However, he considered only the stationary case, that is, the smoother was assumed to have the entire past history of the input available for weighting in its determination of the estimate. Of the three smoothing categories mentioned in Section 8.1, the fixed-lag problem is the most complicated. One reason for this is the start-up problem. For example, let us say we wish to estimate our process at a point three steps back from the current point of measurement. If we enter the problem with the first measurement at $t = 0$, we have only the one measurement on which to base the estimate, and there is a gap in the measurement sequence between the current time and point of estimation. This gap becomes filled as time progresses, but we have to be watchful of this little detail for the first few steps.

Meditch (1) gives an algorithm for fixed-lag smoothing that is considerably more complicated than the algorithms for the other two smoothing categories. We take a somewhat simpler approach here. Knowing the RTS algorithm for fixed-interval smoothing, we can always do fixed-lag smoothing by first filtering up to the current measurement and then sweeping back a fixed number of steps with the RTS algorithm. If the number of backward steps is small, this is a simple and effective way of doing fixed-lag smoothing. If, however, the number of backward steps is large, this method becomes cumbersome and one should look for more efficient algorithms. To illustrate the sweeping back on a running time basis and the start-up problem, we return to the Gauss–Markov example previously considered in Examples 8.1 and 8.2.

EXAMPLE 8.3

The needed filtering data for our Gauss–Markov example is given in Example 8.2 Let us say that we wish to estimate the process three steps back from the current point of measurement, and that the first measurement occurs at $k = 0$.

Table 8.4 Forward and Backward Sweeps (Current Measurement Is at $k = 0$)

	Forward Sweep				*Backward Sweep*		
k	$\hat{x}(k\|k-1)$	$P(k\|k-1)$	$\hat{x}(k\|k)$	$P(k\|k)$	k	$A(k)$	$\hat{x}(k\|0)$
−3	0	1	0	1	0	—	−1.610
−2	0	1	0	1	−1	.9802	−1.578
−1	0	1	0	1	−2	.9802	−1.547
0	0	1	−1.610	.5	−3	.9802	−1.516

Index k is the discrete running time variable. Our first estimate is then to be $\hat{x}(-3|0)$. Anticipating that we will use the Rauch backward sweep from $k = 0$ to $k = -3$, we see that we will need all the filter estimates for this interval. Now, even though we have no measurements prior to $k = 0$, we can still form a trivial forward sweep beginning at $k = -3$. Recall that filter updating without the benefit of a measurement simply amounts to setting the a posteriori $\hat{x}$ and P equal to the a priori $\hat{x}^-$ and P^-. The result of this trivial sweep is shown on the left in Table 8.4. We now have all the filtering data needed for the backward sweep. The desired estimate $\hat{x}(-3|0)$ is obtained from Eqs. (8.2.1) and (8.2.2), and the results are given on the right side of Table 8.4. Presumably, the only estimate of interest here is the last entry in the table, $\hat{x}(-3|0) = -1.516$.

Next, suppose we get a measurement at $k = 1$, and we wish the smoothed estimate three steps back at $k = -2$. The forward sweep that began at $k = -3$ may now be extended one step, and the results are shown at the left in Table 8.5. Note that the only new entries are in the bottom row. The results of the backward sweep are also given in Table 8.5, and all the entries here are new; that is, the backward sweep must be completely redone with each new measurement. Note that each time we increment the filter forward one step, the oldest estimate and its error covariance may be discarded, because we need only the four most recent filter results for the backward sweep. Thus, we have sort of a "sliding window" of data that we must store as time increments forward.

The procedure just described may now be repeated indefinitely. For reference purposes, the solutions for the next two steps are given in Tables 8.6 and

Table 8.5 Forward and Backward Sweeps (Current Measurement Is at $k = 1$)

	Forward Sweep				*Backward Sweep*		
k	$\hat{x}(k\|k-1)$	$P(k\|k-1)$	$\hat{x}(k\|k)$	$P(k\|k)$	k	$A(k)$	$\hat{x}(k\|1)$
−2	0	1	0	1	1	—	−1.317
−1	0	1	0	1	0	.9432	−1.364
0	0	1	−1.610	.5	−1	.9802	−1.337
1	−1.578	.5196	−1.317	.3419	−2	.9802	−1.310

Table 8.6 Forward and Backward Sweeps (Current Measurement Is at $k = 2$)

Forward Sweep					*Backward Sweep*		
k	$\hat{x}(k\|k-1)$	$P(k\|k-1)$	$\hat{x}(k\|k)$	$P(k\|k)$	k	$A(k)$	$\hat{x}(k\|2)$
−1	0	1	0	1	2	—	−1.350
0	0	1	−1.610	.5	1	.9114	−1.371
1	−1.578	.5196	−1.317	.3419	0	.9432	−1.415
2	−1.291	.3677	−1.350	.2689	−1	.9802	−1.387

8.7. The procedure has been demonstrated with tables at each step for the purpose of clarity, but it should be recognized that much of the data in the tables is either repetitious or not necessary for succeeding steps. Thus, the actual programming of the running forward-plus-backward algorithm is not as complicated as would appear from the tabular listings. With each new measurement, it simply involves moving one step forward and M steps back, where M is the desired fixed lag. ■

8.5 FORWARD–BACKWARD FILTER APPROACH TO SMOOTHING

In 1969, D. C. Fraser and J. E. Potter presented a novel solution to the smoothing problem (4). With reference to the fixed-interval problem, their approach was to *filter* the measurement data from both ends to the interior point of interest, and then the two filter estimates were combined to obtain the optimal smoothed estimate. This is illustrated in Fig. 8.5. The equations for the continuous problem will be developed first following the derivation given in Gelb (5). We will then deviate from both Gelb and the Fraser–Potter paper and present a new simplified derivation for the discrete case.

Table 8.7 Forward and Backward Sweeps (Current Measurement Is at $k = 3$)

Forward Sweep					*Backward Sweep*		
k	$\hat{x}(k\|k-1)$	$P(k\|k-1)$	$\hat{x}(k\|k)$	$P(k\|k)$	k	$A(k)$	$\hat{x}(k\|3)$
0	0	1	−1.610	.5	3	—	−1.406
1	−1.578	.5196	−1.317	.3419	2	.8860	−1.423
2	−1.291	.3677	−1.350	.2689	1	.9114	−1.438
3	−1.323	.2975	−1.406	.2293	0	.9432	−1.478

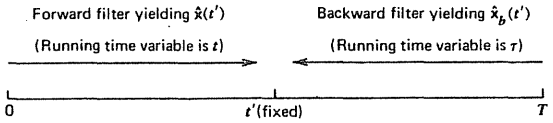


Figure 8.5 Forward–backward filtering. Smoothed estimate $\hat{x}(t'|T)$ is a linear combination of $\hat{x}(t')$ and $\hat{x}_b(t')$.

The Continuous Problem

Suppose we have two independent, unbiased estimates of some parameter x. Call these $\hat{x}_1$ and $\hat{x}_2$ and let the corresponding estimation errors have variances σ_1^2 and σ_2^2. We wish to form a new estimate $\hat{x}$ as a linear combination of $\hat{x}_1$ ad $\hat{x}_2$, where $\hat{x}$ is to have minimum mean-square error. Thus, we write $\hat{x}$ as

$$\hat{x} = k_1\hat{x}_1 + k_2\hat{x}_2 \tag{8.5.1}$$

If the new estimate is to be unbiased,

$$k_1 + k_2 = 1 \tag{8.5.2}$$

Thus, Eq. (8.5.1) may be rewritten as

$$\hat{x} = k_1\hat{x}_1 + (1 - k_1)\hat{x}_2 \tag{8.5.3}$$

The mean-square error for $\hat{x}$ is then

$$E[(x - \hat{x})^2] = E\left\{[x - k_1\hat{x}_1 - (1 - k_1)\hat{x}_2]^2\right\} \tag{8.5.4}$$

or

$$E[e^2] = E\left\{[k_1(e_1 - e_2) + e_2]^2\right\} \tag{8.5.5}$$

where e, e_1, and e_2 are the errors in $\hat{x}$, $\hat{x}_1$, and $\hat{x}_2$, respectively. Equation (8.5.5) may now be differentiated with respect to k_1 and set equal to zero to find the optimal k_1. Performing this operation and noting that e_1 and e_2 are independent yield

$$k_1 = \frac{E[e_2^2]}{E[e_1^2] + E[e_2^2]} = \frac{\sigma_2^2}{\sigma_1^2 + \sigma_2^2} \tag{8.5.6}$$

and

$$k_2 = \frac{E[e_1^2]}{E[e_1^2] + E[e_2^2]} = \frac{\sigma_1^2}{\sigma_1^2 + \sigma_2^2} \tag{8.5.7}$$

The corresponding variance of the optimal estimate $\hat{x}$ is obtained from Eq. (8.5.5) and is

$$\sigma^2 = \frac{\sigma_1^2\sigma_2^2}{\sigma_1^2 + \sigma_2^2} \tag{8.5.8}$$

Notice that the weighing factors k_1 and k_2 used for blending $\hat{x}_1$ and $\hat{x}_2$ are inversely proportional to the variances of the individual estimates. (Electrical engineers usually notice the obvious analogy between this and the division of electric current between two parallel resistors. Variance in the statistics problem plays the same role as resistance in the circuit problem.) We note in passing that Eqs. (8.5.6), (8.5.7), and (8.5.8) may also be written in terms of reciprocals as

$$\sigma^2 = [(\sigma_1^2)^{-1} + (\sigma_2^2)^{-1}]^{-1} \tag{8.5.9}$$

$$k_1 = \sigma^2(\sigma_1^2)^{-1} \tag{8.5.10}$$

$$k_2 = \sigma^2(\sigma_2^2)^{-1} \tag{8.5.11}$$

We now return to the forward–backward filter problem depicted in Fig. 8.5. The forward filter is the usual continuous Kalman filter, as discussed in Chapter 7. Thus, the process model and filter equations are

Forward process model:

$$\left.\begin{aligned}\dot{\mathbf{x}} &= \mathbf{F}\mathbf{x} + \mathbf{G}\mathbf{u} \\ \mathbf{z} &= \mathbf{H}\mathbf{x} + \mathbf{v}\end{aligned}\right\} \begin{array}{l}\mathbf{u} \text{ and } \mathbf{v} \text{ are white-noise processes} \\ \text{with zero crosscorrelation}\end{array} \tag{8.5.12}$$

(8.5.13)

Forward filter equations:

$$\dot{\hat{\mathbf{x}}} = \mathbf{F}\hat{\mathbf{x}} + \mathbf{P}\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}) \tag{8.5.14}$$

$$\dot{\mathbf{P}} = \mathbf{F}\mathbf{P} + \mathbf{P}\mathbf{F}^T - \mathbf{P}\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{P} + \mathbf{G}\mathbf{Q}\mathbf{G}^T \tag{8.5.15}$$

Initial conditions:

$$\hat{\mathbf{x}}(0) = \mathbf{0}, \qquad \mathbf{P}(0) = E[\mathbf{x}(0)\mathbf{x}^T(0)] \tag{8.5.16}$$

The resulting forward estimate and its associated error covariance will be denoted without subscripts as $\hat{\mathbf{x}}(t')$ and $\mathbf{P}(t')$.

For the backward filter, it is convenient to define a new running time variable τ that proceeds backward in time. The backward process model is then obtained by replacing the time derivative in Eq. (8.5.12) with $-d/d\tau$. This leads to the process equation.

Backward process model:

$$\frac{d\mathbf{x}}{d\tau} = -\mathbf{F}\mathbf{x} - \mathbf{G}\mathbf{u} \tag{8.5.17}$$

The corresponding filter equations are then obtained by replacing $\mathbf{F}$ and $\mathbf{G}$ in Eqs. (8.5.14) and (8.5.15) with $-\mathbf{F}$ and $-\mathbf{G}$.

Backward filter equations:

$$\frac{d\hat{\mathbf{x}}_b}{d\tau} = -\mathbf{F}\hat{\mathbf{x}}_b + \mathbf{P}_b\mathbf{H}^T\mathbf{R}^{-1}(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}_b) \tag{8.5.18}$$

$$\frac{d\mathbf{P}_b}{d\tau} = -\mathbf{F}\mathbf{P}_b - \mathbf{P}_b\mathbf{F}^T - \mathbf{P}_b\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H}\mathbf{P}_b + \mathbf{G}\mathbf{Q}\mathbf{G}^T \tag{8.5.19}$$

The boundary conditions for the backward filter equations are awkward, to say the least. Note that we have already used our a priori knowledge about the process in the forward filter. Thus, this information must not be used again in the backward filter. (If it were used in both filters, the two estimates, $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}_b$, would not be independent when they meet at the interior point t'.) This forces us to demand that $\mathbf{P}_b(0) = \infty$. Note that $\tau = 0$ corresponds to $t = T$. The correspondng initial estimate of $\hat{\mathbf{x}}_b$ is arbitrary and immaterial, because the initial $\hat{\mathbf{x}}_b$ is given zero weight due to the infinite initial $\mathbf{P}_b$. Both Gelb and Fraser–Potter give a procedure for avoiding the "infinite-P" problem. Their procedure involves propagating $\mathbf{P}_b^{-1}$ rather than $\mathbf{P}_b$. The boundary conditions, while awkward in the continuous problem, do not present similar problems in the discrete counterpart, so we will be content here to beg the question in the continuous problem and simply say that the initial conditions are awkward. (See Problem 8.6 for a further discussion of this.)

Assume now that forward and backward filters have been solved for $\hat{\mathbf{x}}$, $\mathbf{P}$ and $\hat{\mathbf{x}}_b$, $\mathbf{P}_b$ at the meeting point t'. The appropriate blend of $\hat{\mathbf{x}}$ and $\hat{\mathbf{x}}_b$ and its error covariance are then obtained from Eqs. (8.5.1) and (8.5.9) to (8.5.11) by simply replacing scalar variances with covariance matrices. The final result is then

$$\mathbf{P}(t'|T) = (\mathbf{P}^{-1} + \mathbf{P}_b^{-1})^{-1} \tag{8.5.20}$$

$$\hat{\mathbf{x}}(t'|T) = \mathbf{P}(t'|T)[\mathbf{P}^{-1}\hat{\mathbf{x}} + \mathbf{P}_b^{-1}\hat{\mathbf{x}}_b] \tag{8.5.21}$$

The Discrete Problem

When experimental measurements are recorded for later processing off-line, they are usually sampled and recorded in digital form. Thus, the discrete smoothing problem is often of more practical interest than its continuous counterpart. It can be seen that the Fraser–Potter forward–backward filter approach has considerable potential for computational savings, as compared with the RTS algorithm.

First, by filtering from both ends to the middle, there is no need to store any of the intermediate calculations along the way. The only ones needed for calculating the smoothed estimate are those obtained at the meeting point. Also, filtering does not involve inverting a $\mathbf{P}$ matrix with each step, as is required on the backward sweep of the RTS algorithm. Since the $\mathbf{P}$ matrix often has a much higher dimension than $\mathbf{R}$, having to form $\mathbf{P}^{-1}$ with each step is a time-consuming computation. Thus, a discrete version of the Fraser–Potter approach is attractive and will now be pursued further.

The discrete forward filter has been discussed in some detail in Chapters 5 and 6, and there is no change in either the model or the recursive equations relative to the forward part of the estimator. As before, the a priori knowledge about the process will be incorporated into the forward filter and not the backward one. It is important to note, though, that the discrete forward filter equations are *not* approximations of the continuous filter equations. They are exact in their own right, provided that $\boldsymbol{\phi}_k$, $\mathbf{Q}_k$, $\mathbf{H}_k$, and $\mathbf{R}_k$ are exact. We now seek a similar exact set of equations for the backward filter.

Once we have obtained an a priori estimate and its $\mathbf{P}$ matrix at any point in the backward process, the updating is the same for the forward filter. This should be apparent from the derivation given in Chapter 5, because there is nothing in the derivation that depends on how we got the a priori estimate and its associated error covariance. Thus, the only different aspect in the backward procedure has to do with the projection of $\mathbf{x}_b$ and $\mathbf{P}_b$ from one step to the next. The projection equations can be obtained from the continuous backward filter equations, Eqs. (8.5.18) and (8.5.19). We do this by thinking of having arrived at some point i in the backward process, and then we proceed through the $(i, i + 1)$ interval without benefit of measurements. Saying that we have no measurements is the same as letting $\mathbf{R}^{-1} \to 0$, which indicates worthless measurements. If we let $\mathbf{R}^{-1} = 0$ in Eqs. (8.5.18) and (8.5.19), they reduce to

$$\frac{d\hat{\mathbf{x}}_b}{d\tau} = -\mathbf{F}\hat{\mathbf{x}}_b \tag{8.5.22}$$

$$\frac{d\mathbf{P}_b}{d\tau} = -\mathbf{F}\mathbf{P}_b - \mathbf{P}_b\mathbf{F}^T + \mathbf{G}\mathbf{Q}\mathbf{G}^T \tag{8.5.23}$$

These equations must now be solved for the $(i, i + 1)$ interval, subject to the initial conditions $\hat{\mathbf{x}}_b(\tau_i)$ and $\mathbf{P}_b(\tau_i)$. (We assume here that i increments in unit steps in the same direction as τ.) The solution of Eq. (8.5.22) is obtained from linear system theory, just as in the case of the forward filter. In the interest of simplicity, we assume $\mathbf{F}$, $\mathbf{G}$, and $\mathbf{Q}$ are constant over the interval. Then

$$\hat{\mathbf{x}}_b^*(\tau_{i+1}) = e^{-F\Delta\tau}\hat{\mathbf{x}}_b(\tau_i) \tag{8.5.24}$$

or

$$\hat{\mathbf{x}}_{b(i+1)}^* = \boldsymbol{\phi}^{-1}(\Delta\tau)\hat{\mathbf{x}}_{bi} \tag{8.5.25}$$

where the asterisk * indicates "a priori" and $\boldsymbol{\phi}^{-1}(\Delta\tau)$ is the inverse of the forward transition matrix for the interval. This is exactly what we would expect intuitively.

Since the backward projection of $\mathbf{P}_b$ is not as intuitively obvious as that for $\hat{\mathbf{x}}_b$, we will rely on the formal solution of Eq. (8.5.23) and not try to justify the answer intuitively. We will show that if $\mathbf{P}_b$ is projected according to

$$\mathbf{P}_{b(i+1)}^* = \boldsymbol{\phi}^{-1}(\Delta\tau)[\mathbf{P}_{bi} + \mathbf{Q}_i]\boldsymbol{\phi}^{-1}(\Delta\tau)^T \tag{8.5.26}$$

the differential equation is satisfied. Let the right end of the projection interval be fixed at $\tau = \tau_0$. The left end is then the negatively running time variable τ, and $\tau > \tau_0$. We speculate that

$$\mathbf{P}_b(\tau) = \boldsymbol{\phi}^{-1}(\tau - \tau_0)[\mathbf{P}_b(\tau_0) + \mathbf{Q}_i(\tau)]\boldsymbol{\phi}^{-1^T}(\tau - \tau_0) \tag{8.5.27}$$

will satisfy Eq. (8.5.23). Note that the discrete $\mathbf{Q}_i$ is written as a function of τ, because it varies as the interval increases. Differentiation of the assumed solution, Eq. (8.5.27), leads to

$$\frac{d\mathbf{P}_b}{d\tau} = \frac{d\boldsymbol{\phi}^{-1}}{d\tau}[\mathbf{P}_b(\tau_0) + \mathbf{Q}_i]\boldsymbol{\phi}^{-1^T} + \boldsymbol{\phi}^{-1}\frac{d\mathbf{Q}_i}{d\tau}\boldsymbol{\phi}^{-1^T} + \boldsymbol{\phi}^{-1}[\mathbf{P}_b(\tau_0) + \mathbf{Q}_i]\frac{d\boldsymbol{\phi}^{-1^T}}{d\tau} \tag{8.5.28}$$

where the parenthetical dependence has been omitted in places to save writing. We first note from linear system theory that

$$\frac{d\boldsymbol{\phi}^{-1}}{d\tau} = -\mathbf{F}\boldsymbol{\phi}^{-1} \tag{8.5.29}$$

Thus, the first and third terms of Eq. (8.5.28) combine to yield $(-\mathbf{F}\mathbf{P}_b - \mathbf{P}_b\mathbf{F}^T)$, which is identical to the first two terms on the right side of Eq. (8.5.23). Looking next at the $\boldsymbol{\phi}^{-1}(d\mathbf{Q}_i/d\tau)\boldsymbol{\phi}^{-1^T}$ term in Eq. (8.5.28), we see that we need to form the derivative of $\mathbf{Q}_i$. Equation (5.3.6) is useful here. Let $t_k = T - \tau$, $t_{k+1} = T - \tau_0$, and note that $E[\mathbf{u}(\xi)\mathbf{u}^T(\eta)] = \mathbf{Q}\,\delta(\xi - \eta)$. After performing the inner integration, we have

$$\mathbf{Q}_i = \int_{T-\tau}^{T-\tau_0} \boldsymbol{\phi}(T - \tau_0, \eta)\mathbf{G}\mathbf{Q}\mathbf{G}^T\boldsymbol{\phi}^T(T - \tau_0, \eta)\, d\eta \tag{8.5.30}$$

Differentiating this with respect to τ leads to

$$\frac{d\mathbf{Q}_i}{d\tau} = \boldsymbol{\phi}(\tau - \tau_0)\mathbf{G}\mathbf{Q}\mathbf{G}^T\boldsymbol{\phi}^T(\tau - \tau_0) \tag{8.5.31}$$

This may now be pre- and postmultiplied by $\boldsymbol{\phi}^{-1}(\tau - \tau_0)$ and $\boldsymbol{\phi}^{-1^T}(\tau - \tau_0)$, as indicated in Eq. (8.5.28), and the result is obviously $\mathbf{G}\mathbf{Q}\mathbf{G}^T$, which equals the third term on the right side of Eq. (8.5.23). Thus, the discrete covariance projection given by Eq. (8.5.26) is seen to satisfy the continuous differential equa-

tion for $\mathbf{P}_b$ with $\mathbf{R}^{-1}$ set equal to zero. It is important to note that the $\boldsymbol{\phi}$ and $\mathbf{Q}_i$ in the projection equations are computed from the *forward* process equations. That is, they are the same parameters that would have been used in the forward filter for the interval, had the forward recursive steps been continued up through that interval of time.

The boundary conditions for the backward filter are awkward, but not impossible. One way of implementing the infinite initial $\mathbf{P}$ matrix in off-line processing is to make all terms along its major diagonal very large, say, about 10 orders of magnitude larger than their initial counterparts in the forward filter. The off-diagonal terms may then be set equal to zero. For all practical purposes, this completely de-weights the prior information about the process relative to the weight that it is given in the forward filter. This is not a very elegant approach, but it is an effective, practical solution in many applications. In simple models where the order of the state vector is small, the boundary-condition problem may be handled analytically by using the alternative Kalman filter algorithm for the first step in the backward filter. This is illustrated in the one-state example presented at the end of this section, and it is also discussed further in Problem 8.6. If all else fails, we can resort to propagating $\mathbf{P}$ inverse as suggested in the Fraser–Potter paper cited earlier.

We assume now that the forward filter has been stopped ahead recursively at the estimation point, say, k, and the end result is an a posteriori estimate $\mathbf{x}_k$ and an associated $\mathbf{P}_k$. The backward filter steps backward from the end point N, and it stops at $k+1$ where it assimilates the $\mathbf{z}^{k+1}$ measurement. It then projects this estimate one more step to obtain an a priori estimate $\mathbf{x}_{bk}^*$ and its associated $\mathbf{P}_{bk}^*$. It does not assimilate the $\mathbf{z}_k$ measurement, because this has already been used in the forward filter. Finally, the forward and backward estimates are blended together in accordance with the equation

$$\hat{\mathbf{x}}(k|N) = \mathbf{P}(k|N)[\mathbf{P}_k^{-1}\hat{\mathbf{x}}_k + \mathbf{P}_{bk}^{*-1}\hat{\mathbf{x}}_{bk}^*] \tag{8.5.32}$$

where

$$\mathbf{P}(k|N) = [\mathbf{P}_k^{-1} + \mathbf{P}_{bk}^{*-1}]^{-1} \tag{8.5.33}$$

An example will now illustrate the procedure.

EXAMPLE 8.4

Again, we consider the same Gauss–Markov process used for the previous examples of this chapter. Let us say we want to find the smoothed estimate at $k = 48$, that is, $\hat{x}(48|50)$. We first forward-filter up through z_{48}. This has already been done in previous examples and the results are given in Table 8.1. The pertinent forward-filter results are

$$\hat{x}_{48} = -.511$$
$$P_{48} = .165$$

We now need to generate a similar estimate at $k = 48$ with the backward

filter. We begin at the end point where $k = 50$. The initial conditions for the backward filter are

$$\hat{x}_{b0}^* = 0, \qquad P_{b0}^* = \infty \tag{8.5.34}$$

where the asterisk indicates "a priori" in the backward filter, and the subscript indicates the backward index i, rather than k. We update the initial estimate with the alternative Kalman filter algorithm in order to avoid the P-equal-infinity problem. Thus, we have at $k = 50$ $(i = 0)$

$$\begin{aligned} P_{b0}^{-1} &= P_{b0}^{*-1} + H_0^T R_0^{-1} H_0 \\ &= 0 + 1 \end{aligned} \tag{8.5.35}$$

or

$$P_{b0} = 1 \tag{8.5.36}$$

The gain is then

$$K_{b0} = P_{b0} H_0^T R_0^{-1} = 1 \tag{8.5.37}$$

and the updated estimate is

$$\hat{x}_{b0} = \hat{x}_{b0}^* + K_{b0}[z_0 - H_0\hat{x}_{b0}^*] = z_0 = -1.138$$
$$\text{(from } k = 50 \text{ entry of Table 8.1)} \tag{8.5.38}$$

Note that we could have taken a more heuristic approach here and simply said: "Initially, we knew nothing about the backward process at the end point, and hence we must accept the measurement at $k = 50$ at face value. The resulting estimation error is then just the measurement error." (This philosophy can also be extended to the vector case, as shown in Problem 8.6.)

Continuing the backward filter, we next project x_b and P_b to $k = 49$ $(i = 1)$ using Eqs. (8.5.25) and (8.5.26). The results are

$$\hat{x}_{b1}^* = e^{.02}\hat{x}_{b0} = e^{.02}(-1.138) = -1.161 \tag{8.5.39}$$

$$P_{b1}^* = e^{.02}(P_{b0} + Q_i)e^{.02} = (e^{.02})^2[1 + .0392] = 1.0816 \tag{8.5.40}$$

Next, we compute the gain and update the a priori estimate using the regular Kalman filter algorithm:

$$K_{b1} = \frac{P_{b1}^*}{P_{b1}^* + 1} = .5196 \tag{8.5.41}$$

$$\hat{x}_{b1} = \hat{x}_{b1}^* + K_{b1}(z_1 - \hat{x}_{b1}^*) = -.591 \tag{8.5.42}$$

(Note from Table 8.1 that the measurement at $k = 49$ is $-.064$). The a posteriori error covariance matrix is then

$$P_{b1} = (1 - K_{b1})P_{b1}^* = .5196 \tag{8.5.43}$$

Now we project $\hat{x}_{b1}$ and P_{b1} to $k = 48$ ($i = 2$).

$$\hat{x}_{b2}^* = e^{.02}\hat{x}_{b1} = -.603 \tag{8.5.44}$$

$$P_{b2}^* = e^{.02}(P_{b1} + Q_i)e^{.02} = .5816 \tag{8.5.45}$$

The backward filter is stopped at this point, because the measurement at $k = 48$ has already been assimilated in the forward filter.

We are now ready to blend together the forward and backward estimates in accordance with Eqs. (8.5.32) and (8.5.33). The results are

$$P(48|50) = [(.1653)^{-1} + (.5816)^{-1}]^{-1} = .1287$$

and

$$\hat{x}(48|50) = P(48|50)[P_{48}^{-1}\hat{x}_{48} + P_{b2}^{*-1}\hat{x}_{b2}^*] = -.531$$

Notice that these results are the same as those given in Table 8.2, which were obtained using the RTS algorithm. ▩

PROBLEMS

8.1 Consider the same signal and noise situation used in Example 4.3 We found there that the causal Wiener filter was a simple first-order low-pass filter. Using Eq. (4.3.22), find the stationary, fixed-lag smoothing solution for $\alpha = -1$. Recall that the signal and noise are independent processes with autocorrelation functions

$$R_s(\tau) = e^{-|\tau|}$$

$$R_n(\tau) = \delta(\tau)$$

The solution may be given as a transfer function, that is, find $G(s)$ rather than $g(t)$. Note that the smoothing solution is considerably more complicated than the corresponding filter solution.

8.2 (a) Consider the one-step back, fixed-interval smoothing problem. Its solution may be obtained from the usual Kalman filter equations by batching together the last two measurements $\mathbf{z}_{N-1}$ and $\mathbf{z}_N$ and considering them as one measurement occurring at $N - 1$. Note that $\mathbf{z}_N$ can be linearly related to $\mathbf{x}_{N-1}$ as follows:

$$\begin{aligned}\mathbf{z}_N &= \mathbf{H}_N\mathbf{x}_N + \mathbf{v}_N \\ &= \mathbf{H}_N(\boldsymbol{\phi}_{N-1}\mathbf{x}_{N-1} + \mathbf{w}_{N-1}) + \mathbf{v}_N \\ &= (\mathbf{H}_N\boldsymbol{\phi}_{N-1})\mathbf{x}_{N-1} + (\mathbf{H}_N\mathbf{w}_{N-1} + \mathbf{v}_N)\end{aligned}$$

The batched measurement relationship at $N - 1$ is then

$$\begin{bmatrix} \mathbf{z}_{N-1} \\ \hline \mathbf{z}_N \end{bmatrix} = \begin{bmatrix} \mathbf{H}_{N-1} \\ \hline \mathbf{H}_N\boldsymbol{\phi}_{N-1} \end{bmatrix} \mathbf{x}_{N-1} + \begin{bmatrix} \mathbf{v}_{n-1} \\ \hline \mathbf{H}_N\mathbf{w}_{N-1} + \mathbf{v}_N \end{bmatrix}$$

Note that the upper and lower components of the batched measurement can be processed sequentially because their errors are uncorrelated. Assume now that we have an a priori estimate $\hat{\mathbf{x}}(N - 1|N - 2)$ and its error covariance matrix $\mathbf{P}(N - 1|N - 2)$. Proceed to update the estimate by assimilating the two components of the measurement *separately* in two steps, and show the final result is the same as that obtained using the RTS algorithm.

(*Hint*: The first sequential step yields $\hat{\mathbf{x}}(N - 1|N - 1)$. Next, show that the gain for the second sequential step is $\mathbf{P}(N - 1|N - 1)\boldsymbol{\phi}_{N-1}^T\mathbf{P}(N|N - 1)^{-1}\mathbf{K}_N$, where $\mathbf{K}_N$ is the usual Kalman filter gain for the Nth stage. Finally, replace $\hat{\mathbf{x}}(N|N)$ in the RTS formula with $[\hat{\mathbf{x}}(N|N - 1) + \mathbf{K}_N(\mathbf{z}_N - \mathbf{H}_N\hat{\mathbf{x}}(N|N - 1))]$ and show the equivalence.)

(b) The exercise of part (a) can be generalized to justify the RTS algorithm for any interior point within the fixed interval from 0 to N. To do this, let the interior point be denoted k and batch together all subsequent measurements $\mathbf{z}_{k+1}, \mathbf{z}_{k+2}, \ldots, \mathbf{z}_N$. Call the batched measurement $\mathbf{y}_{k+1}$, that is,

$$\mathbf{y}_{k+1} = \begin{bmatrix} \mathbf{z}_{k+1} \\ \hline \mathbf{z}_{k+2} \\ \hline \vdots \\ \mathbf{z}_N \end{bmatrix}$$

We can now form a linear connection between $\mathbf{x}_{k+1}$ and $\mathbf{y}_{k+1}$ as

$$\mathbf{y}_{k+1} = \mathbf{M}_{k+1}\mathbf{x}_{k+1} + \mathbf{v}'_{k+1}$$

The batched measurement $\mathbf{y}_{k+1}$ now plays the same role as $\mathbf{z}_N$ in part (a). (We do not actually have to write out $\mathbf{M}_{k+1}$ and $\mathbf{v}'_{k+1}$ explicitly. We simply need to know that such a relationship exists.) We can now consider the interior-point smoothing problem in terms of an equivalent filter problem that terminates at $k + 1$. This is the same as the problem considered in (a) except for notation. Proceed through the steps of exercise (a) again with appropriate changes in notation and show that the generalization is valid.

8.3 Consider a stationary, Gauss–Markov process with an autocorrelation function

$$R_x(\tau) = e^{-|\tau|}$$

Assume that we have two noisy measurements of this process that were made at $t = 0$ and $t = 1$. Call these z_0 and z_1. The measurement errors associated with

z_0 and z_1 are white and have a variance of unity. We wish to find the optimal estimate of x at $t = 0$, given z_0 and z_1, that is, we desire $\hat{x}(0|1)$. Write an expression for $\hat{x}(0|1)$ explicitly in terms of the measurements z_0 and z_1 using:

(a) The RTS algorithm.
(b) The Fraser–Potter forward–backward filter method.
(c) The weight factor method (see Section 4.7).
(d) The fixed-point algorithm of Section 8.3.

8.4 Show that the continuous version of the RTS algorithm is as follows:

$$\dot{\hat{\mathbf{x}}}(t|T) = \mathbf{F}\hat{\mathbf{x}}(t|T) + \mathbf{G}\mathbf{Q}\mathbf{G}^T\mathbf{P}^{-1}(t|t)[\hat{\mathbf{x}}(t|T) - \hat{\mathbf{x}}(t|t)]$$

$$\dot{\mathbf{P}}(t|T) = [\mathbf{F} + \mathbf{G}\mathbf{Q}\mathbf{G}^T\mathbf{P}^{-1}(t|t)]\mathbf{P}(t|T) + \mathbf{P}(t|T)[\mathbf{F} + \mathbf{G}\mathbf{Q}\mathbf{G}^T\mathbf{P}^{-1}(t|t)]^T - \mathbf{G}\mathbf{Q}\mathbf{G}^T$$

Boundary conditions: $\hat{\mathbf{x}}(T|T)$ and $\mathbf{P}(T|T)$ obtained from filter solution. (*Hint*: Begin with the discrete RTS algorithm and then let the step size approach zero, just as was done in deriving the filter equations in Chapter 7. Recall that $\mathbf{Q}_k$ in the discrete model approaches $\mathbf{G}\mathbf{Q}\mathbf{G}^T\,\Delta t$ for small Δt. Also note that $[\mathbf{I} - \mathbf{A}(k)]$ is of the order of Δt in the smoothing algorithm; thus, the gain $\mathbf{A}(k)$ approaches $\mathbf{I}$ in the limit as $\Delta t \to 0$.)

8.5 Consider a Wiener process and measurement situation as follows:

$$\dot{x} = u(t), \qquad x(0) = 0$$

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

where $u(t)$ and $v(t)$ are independent Gaussian white-noise processes with spectral amplitudes q and r, respectively.

(a) Assume a fixed interval T that is sufficiently large to allow the filter solution to reach a steady-state condition on the forward sweep. This then becoms the boundary condition for the backward sweep. Using the continuous RTS algorithm given in Problem 8.4, show that $P(t|T)$ for the terminal region of the $[0, T]$ interval is approximated by

$$P(t|T) \approx \frac{\alpha}{2}[1 + e^{-2\beta(T-t)}]$$

where $\alpha = \sqrt{rq}$ and $\beta = \sqrt{q/r}$.

(b) Note that the solution of part (a) approaches $\alpha/2$ as $t \to 0$. This is obviously not compatible with the known a priori boundary condition for a Wiener process; that is, $P(0) = 0$. Explain this discrepancy.

8.6 Consider the two-state system

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -1 & 0 \\ 0 & -2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}$$

where u_1 and u_2 are independent Gaussian white-noise inputs with unity-amplitude spectral functions. We have a sequence of scalar measurements of this process $z_0, z_1, \ldots, z_N$ that are related to the process by the equation

$$z_k = [1 \quad 1]\mathbf{x}_k + v_k, \qquad k = 0, 1, \ldots, N$$

where v_k has unity variance. That is, we are allowed to observe only the sum of the state variables at each step and not their individual values. Suppose we want an estimate of the process at some interior point, and we wish to get it using the Fraser–Potter forward–backward-filter method. In this case, the single measurement z_N does not provide enough information to yield a finite error-covariance estimate of x at $t = N$. Thus, we cannot start the backward filter quite as easily as was done in Example 8.4. In this problem, let the Δt interval be unity and show that the backward filter may be started at $t = N - 1$ by batching together z_N and z_{N-1} into an equivalent vector measurement at $t = N - 1$. The same technique used in Problem 8.2 will be helpful here. In effect, you need to show that the error covariance after assimilating z_N and z_{N-1} is finite and nonsingular, and that the estimate is the same as would be obtained by deterministic methods. (The extension of this technique to higher-order systems is fairly obvious, provided the system is observable. We simply batch together an appropriate number of measurements at the end of the interval and then solve for the system state, just as if the measurements were noise-free. This then becomes the initial state estimate for the backward filter, and we start the backward filtering an appropriate number of steps back from the end point.)

8.7 Table 8.3 sumarizes the results of the fixed-point smoothing simulation of Example 8.2. Notice that the filter error variances listed under the column $P(j|j)$ are identical to the smoothing error variances $P(0|j)$. Give an intuitive explanation of this coincidence.

8.8 A rough sketch of the error variances for the forward and backward sweeps for Example 8.1 is shown in Fig. 8.3. Using MATLAB (or other suitable software), calculate and plot the 51 variances for each of the forward and backward sweeps. Note the symmetry in the backward-sweep variances as seen from the midpoint of the measurement stream. Give an intuitive explanation for the symmetry.

8.9 A two-state model for the GPS selective availability process was given in Example 6.1, Chapter 6. The model is repeated here for convenience.

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega_0^2 & -\sqrt{2}\omega_0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ \sqrt{c} \end{bmatrix} u(t)$$

$$z_k = [1 \quad 0]\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}_k + v_k$$

where

$$\omega_o = .012 \text{ rad/sec}$$

$$c = .0043987 \text{ m}^2/(\text{rad/sec})^3$$

$$u(t) = \text{unity white noise}$$

In this model, x_1 and x_2 are phase variables and both have finite variances. Consider the off-line fixed-interval smoothing problem where we have 101 noisy samples of this process. The sampling interval is 20 sec, and the rms noise

associated with each sample is 20 m. Calculate the rms errors associated with each of the smoothed estimates for this situation. Note (a) the improvement in the state estimates in the midrange of the data span as compared with the estimate at the end point (i.e., the best *filtered* estimate), and (b) the symmetry in the rms error sequence relative to the midpoint of the measurement span.

REFERENCES CITED IN CHAPTER 8

1. S. Meditch, *Stochastic Optimal Linear Estimation and Control*, New York: McGraw-Hill, 1969.
2. H. E. Rauch, "Solutions to the Linear Smoothing Problem," *IEEE Trans. Auto. Control, AC-8:* 371 (1963).
3. H. E. Rauch, F. Tung, and C. T. Striebel, "Maximum Likelihood Estimates of Linear Dynamic Systems," *AIAA J., 3:* 1445 (1965).
4. D. C. Fraser and J. E. Potter, "The Optimum Linear Smoother as a Combination of Two Optimum Linear Filters," *IEEE Trans. Auto. Control, AC-14*(4): 387 (Aug. 1969).
5. A. Gelb (ed.), *Applied Optimal Estimation*, Cambridge, MA: MIT Press, 1974.

Additional References on Smoothing

6. A. P. Sage and J. L. Melsa, *Estimation Theory with Applications to Communications and Control*, New York: McGraw-Hill, 1971.
7. B. D. O. Anderson and J. B. Moore, *Optimal Filtering*, Englewood Cliffs, NJ: Prentice-Hall, 1979.
8. M. S. Grewal and A. P. Andrews, *Kalman Filtering Theory and Practice*, Englewood Cliffs, NJ: Prentice-Hall, 1993.
9. G. Minkler and J. Minkler, *Theory and Application of Kalman Filtering*, Palm Bay, FL: Magellan Book Co., 1993.

9

Linearization and Additional Intermediate-Level Topics on Applied Kalman Filtering

Kalman's papers of the early 1960s (1, 2) were recognized almost immediately as new and important contributions to least-squares filtering. As a result, there was a renewal of research interest in this area, and a flurry of papers expanding on Kalman's original work followed during the next decade or so. Kailath (3) gives an especially comprehensive bibliography of papers for this period. Research work in this area still continues (although perhaps at a somewhat reduced rate), and new applications and extensions continue to appear regularly in the technical literature. A few of the more significant extensions and related topics have been selected for comment here. The list is by no means comprehensive. However, there is a hierarchy of importance, and it is the authors' recommendation that the reader begin with the first topic on linearization. This is, by far, the most important section in this chapter. The other sections may be studied in any desired order as time permits.

9.1 LINEARIZATION

Some of the most successful applications of Kalman filtering have been in situations with nonlinear dynamics and/or nonlinear measurement relationships. We now examine two basic ways of linearizing the problem. One is to linearize about some nominal trajectory in state space that does not depend on the measurement data. The resulting filter is usually referred to as simply a *linearized Kalman filter*. The other method is to linearize about a trajectory that is continually updated with the state estimates resulting from the measurements. When this is done, the filter is called an *extended Kalman filter*. A brief discussion of each will now be presented.

Linearized Kalman Filter

We begin by assuming the process to be estimated and the associated measurement relationship may be written in the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}_d, t) + \mathbf{u}(t) \tag{9.1.1}$$

$$\mathbf{z} = \mathbf{h}(\mathbf{x}, t) + \mathbf{v}(t) \tag{9.1.2}$$

where $\mathbf{f}$ and $\mathbf{h}$ are known functions, $\mathbf{u}_d$ is a deterministic forcing function, and $\mathbf{u}$ and $\mathbf{v}$ are white-noise processes with zero crosscorrelation as before. Note that nonlinearity may enter into the problem either in the dynamics of the process or in the measurement relationship. Also, note that the forms of Eqs. (9.1.1) and (9.1.2) are somewhat restrictive in that $\mathbf{u}$ and $\mathbf{v}$ are assumed to be separate additive terms and are not included with the $\mathbf{f}$ and $\mathbf{h}$ terms. However, to do otherwise complicates the problem considerably, and thus we will stay with these restrictive forms.

Let us now assume that an approximate trajectory $\mathbf{x}^*(t)$ may be determined by some means. This will be referred to as the nominal or reference trajectory, and it is illustrated along with the actual trajectory in Fig. 9.1. The actual trajectory $\mathbf{x}(t)$ may then be written as

$$\mathbf{x}(t) = \mathbf{x}^*(t) + \Delta\mathbf{x}(t) \tag{9.1.3}$$

Equations (9.1.1) and (9.1.2) then become

$$\dot{\mathbf{x}}^* + \Delta\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}^* + \Delta\mathbf{x}, \mathbf{u}_d, t) + \mathbf{u}(t) \tag{9.1.4}$$

$$\mathbf{z} = \mathbf{h}(\mathbf{x}^* + \Delta\mathbf{x}, t) + \mathbf{v}(t) \tag{9.1.5}$$

We now assume $\Delta\mathbf{x}$ is small and approximate the $\mathbf{f}$ and $\mathbf{h}$ functions with Taylor's series expansions, retaining only first-order terms. The result is

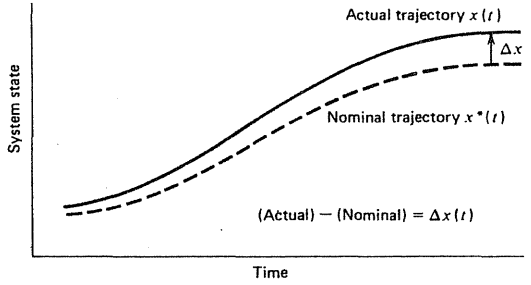


Figure 9.1 Nominal and actual trajectories for a linearized Kalman filter.

$$\dot{\mathbf{x}}^* + \Delta\dot{\mathbf{x}} \approx \mathbf{f}(\mathbf{x}^*, \mathbf{u}_d, t) + \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right]_{\mathbf{x}=\mathbf{x}^*} \cdot \Delta\mathbf{x} + \mathbf{u}(t) \tag{9.1.6}$$

$$\mathbf{z} \approx \mathbf{h}(\mathbf{x}^*, t) + \left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right]_{\mathbf{x}=\mathbf{x}^*} \cdot \Delta\mathbf{x} + \mathbf{v}(t) \tag{9.1.7}$$

where

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots \\ \vdots & \vdots & \end{bmatrix}; \qquad \frac{\partial \mathbf{h}}{\partial \mathbf{x}} = \begin{bmatrix} \frac{\partial h_1}{\partial x_1} & \frac{\partial h_1}{\partial x_2} & \cdots \\ \frac{\partial h_2}{\partial x_1} & \frac{\partial h_2}{\partial x_2} & \cdots \\ \vdots & \vdots & \end{bmatrix} \tag{9.1.8}$$

It is customary to choose the nominal trajectory $\mathbf{x}^*(t)$ to satisfy the deterministic differential equation

$$\dot{\mathbf{x}}^* = \mathbf{f}(\mathbf{x}^*, \mathbf{u}_d, t) \tag{9.1.9}$$

Substituting this into (9.1.6) then leads to the linearized model

$$\Delta\dot{\mathbf{x}} = \left[\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right]_{\mathbf{x}=\mathbf{x}^*} \cdot \Delta\mathbf{x} + \mathbf{u}(t) \quad \text{(linearized dynamics)} \tag{9.1.10}$$

$$[\mathbf{z} - \mathbf{h}(\mathbf{x}^*, t)] = \left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right]_{\mathbf{x}=\mathbf{x}^*} \cdot \Delta\mathbf{x} + \mathbf{v}(t) \quad \text{(linearized measurement equation)} \tag{9.1.11}$$

Note that the "measurement" in the linear model is the actual measurement less that predicted by the nominal trajectory in the absence of noise. Also the equivalent $\mathbf{F}$ and $\mathbf{H}$ matrices are obtained by evaluating the partial derivative matrices (Eqs. 9.1.8) along the *nominal* trajectory. We will now look at two examples that illustrate the linearization procedure. In the first example the nonlinearity appears only in the measurement relationship, so it is relatively simple. In the second, nonlinearity occurs in both the measurement and process dynamics, so it is somewhat more involved than the first.

EXAMPLE 9.1

In many electronic navigation systems the basic observable is a noisy measurement of range (distance) from the vehicle to a known location. One such system that has enjoyed wide use in aviation is distance-measuring equipment (DME) (4). We do not need to go into detail here as to how the equipment works. It suffices to say that the airborne equipment transmits a pulse that is returned by the ground station, and then the aircraft equipment interprets the transit time in

terms of distance. In our example we will simplify the geometric situation by assuming that the aircraft and the two DME stations are all in a horizontal plane as shown in Fig. 9.2 (slant range $\approx$ horizontal range). The coordinates of the two DME stations are assumed to be known, and the aircraft coordinates are unknown and to be estimated.

We will look at the aircraft dynamics first. This, in turn, will determine the process state model. To keep things as simple as possible, we will assume a nominal straight-and-level flight condition with constant velocity. The true trajectory will be assumed to be the nominal one plus small perturbations due to random horizontal accelerations, which will be assumed to be white. This leads to random walk in velocity and integrated random walk in position. This is probably unrealistic for long time spans because of the control applied by the pilot (or autopilot). However, this would be a reasonable model for short intervals of time. The basic differential equations of motion in the x and y directions are then

$$\begin{aligned} \ddot{x} &= 0 + u_x \\ \ddot{y} &= \underbrace{0}_{\text{Deterministic forcing function}} + \underbrace{u_y}_{\text{Random forcing function}} \end{aligned} \tag{9.1.12}$$

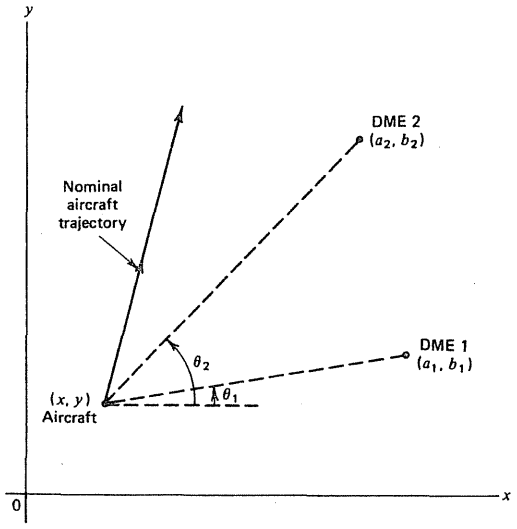


Figure 9.2 Geometry for DME example.

The dynamical equations are seen to be linear in this case, so the differential equations for the incremental quantities are the same as for the total x and y, that is,

$$\begin{aligned} \Delta\ddot{x} &= u_x \\ \Delta\ddot{y} &= u_y \end{aligned} \tag{9.1.13}$$

We now define filter state variables in terms of the incremental positions and velocities:

$$\begin{aligned} x_1 &= \Delta x, \qquad x_2 = \Delta\dot{x} \\ x_3 &= \Delta y, \qquad x_4 = \Delta\dot{y} \end{aligned} \tag{9.1.14}$$

The state equations are then

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} 0 \\ u_x \\ 0 \\ u_y \end{bmatrix} \tag{9.1.15}$$

The state variables are driven by the white-noise processes u_x and u_y, so we are assured that the corresponding discrete equations will be in the appropriate form for a Kalman filter.

We now turn to the measurement relationships. We will assume that we have two simultaneous range measurements, one to DME_1 and the other to DME_2. The two measurement equations in terms of the total x and y are then

$$\begin{aligned} z_1 &= \sqrt{(x - a_1)^2 + (y - b_1)^2} + v_1 \\ z_2 &= \sqrt{(x - a_2)^2 + (y - b_2)^2} + v_2 \end{aligned} \tag{9.1.16}$$

where v_1 and v_2 are additive white measurement noises. We see immediately that the connection between the observables (z_1 and z_2) and the quantities to be estimated (x and y) is nonlinear. Thus, linearization about the nominal trajectory is in order. We assume that an approximate nominal position is known at the time of the measurement, and that the locations of the two DME stations are known exactly. We now need to form the $\partial\mathbf{h}/\partial\mathbf{x}$ matrix as specified by Eq. (9.1.8). [We note a small notational problem here. The variables x_1, x_2, x_3, and x_4 are used in Eq. (9.1.8) to indicate total state variables, and then the same symbols are used again to indicate incremental state variables as defined by Eqs. (9.1.14). However, the meanings of the symbols are never mixed in any one set of equations, so this should not lead to confusion.] We now note that the x and y position variables are the first and third elements of the state vector. Thus, evaluation of the partial derivatives indicated in Eq. (9.1.8) leads to

$$\frac{\partial \mathbf{h}}{\partial \mathbf{x}} = \begin{bmatrix} \dfrac{(x_1 - a_1)}{\sqrt{(x_1 - a_1)^2 + (x_3 - b_1)^2}} & 0 & \dfrac{(x_3 - b_1)}{\sqrt{(x_1 - a_1)^2 + (x_3 - b_1)^2}} & 0 \\ \dfrac{(x_1 - a_2)}{\sqrt{(x_1 - a_2)^2 + (x_3 - b_2)^2}} & 0 & \dfrac{(x_3 - b_2)}{\sqrt{(x_1 - a_2)^2 + (x_3 - b_2)^2}} & 0 \end{bmatrix} \tag{9.1.17}$$

or

$$\frac{\partial \mathbf{h}}{\partial \mathbf{x}} = \begin{bmatrix} -\cos\theta_1 & 0 & -\sin\theta_1 & 0 \\ -\cos\theta_2 & 0 & -\sin\theta_2 & 0 \end{bmatrix} \tag{9.1.18}$$

Finally, we note that Eq. (9.1.18) can be generalized even further, since the sine and cosine terms are actually direction cosines between the x and y axes and the respective lines of sight to the two DME stations. Therefore, we will write the linearized $\mathbf{H}$ matrix in its final form as

$$\mathbf{H} = \left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\mathbf{x}=\mathbf{x}^*} = \begin{bmatrix} -\cos\theta_{x1} & 0 & -\cos\theta_{y1} & 0 \\ -\cos\theta_{x2} & 0 & -\cos\theta_{y2} & 0 \end{bmatrix} \tag{9.1.19}$$

where the subscripts on θ indicate the respective axes and lines of sight to the DME stations. Note that $\mathbf{H}$ is evaluated at a point on the *nominal* trajectory. (The true trajectory is not known to the filter.) The nominal aircraft position will change with each step of the recursive process, so the terms of $\mathbf{H}$ are time-variable and must be recomputed with each recursive step. Also, recall from Eq. (9.1.11) that the measurement presented to the linearized filter is the total $\mathbf{z}$ minus the predicted $\mathbf{z}$ based on the nominal position $\mathbf{x}^*$.

Strictly speaking, the linearized filter is always estimating incremental quantities, and then the total quantity is reconstructed by adding the incremental estimate to the nominal part. However, we will see later that when it comes to the actual mechanics of handling the arithmetic on the computer, we can avoid working with incremental quantities if we choose to do so. This is discussed further in the section on the extended Kalman filter. We will now proceed to a second linearization example, where the process dynamics as well as the measurement relationship has to be linearized. ■

EXAMPLE 9.2

"Kalman Filtering Techniques" in C.T. Leondes (Ed) Advances in Control Systems, Vol. 3, Academic Press, 1966

This example is taken from Sorenson (5) and is a classic example of linearization of a nonlinear problem. Consider a near-earth space vehicle in a nearly circular orbit. It is desired to estimate the vehicle's position and velocity on the basis of a sequence of angular measurements made with a horizon sensor. With reference to Fig. 9.3, the horizon sensor is capable of measuring:

1. The angle γ between the earth's horizon and the local vertical.
2. The angle α between the local vertical and a known reference line (say, to a celestial object).

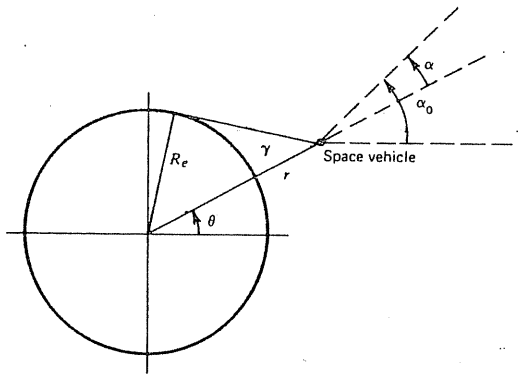


Figure 9.3 Coordinates for space vehicle example.

In the interest of simplicity, we assume all motion and measurements to be within a plane as shown in Fig. 9.3. Thus, the motion of the vehicle can be described with the usual polar coordinates r and θ.

The equations of motion for the space vehicle may be obtained from either Newtonian or Lagrangian mechanics. They are (see Section 2-10, ref. 6):

[R.G. Brown & J.W. Nilsson, Introduction to Linear Systems Analysis, J. Wiley, 1962]

Radial dir.:

$$\ddot{r} - r\dot{\theta}^2 = -\frac{K}{r^2} + u_r(t) \quad \longrightarrow \quad \ddot{r} - r\dot{\theta}^2 + \frac{K}{r^2} = u_r(t) \tag{9.1.20}$$

Tangential dir.:

$$r\ddot{\theta} + 2\dot{r}\dot{\theta} = u_\theta(t) \tag{9.1.21}$$

where K is a constant proportional to the universal gravitational constant, and u_r and u_θ are small random forcing functions in the r and θ directions (due mainly to gravitational anomalies unaccounted for in the K/r^2 term). It can be seen that the constant K must be equal to gR_e^2 if the gravitational forcing function is to match the earth's gravity constant g at the surface. The random forcing functions u_r and u_θ will be assumed to be white. We will look at the linearized process dynamics first and then consider the nonlinear measurement situation later.

The equations of motion, Eqs. (9.1.20) and (9.1.21), are clearly nonlinear, so we must linearize the dynamics if we are to apply Kalman filter methods. We have assumed that random forcing functions u_r and u_θ are small, so the corresponding perturbations from a circular orbit will also be small. By direct substitution into Eqs. (9.1.20) and (9.1.21), it can be verified that

$$r^* = R_0 \quad \text{(a constant radius)} \tag{9.1.22}$$

$$\theta^* = \omega_0 t \quad \left(\omega_0 = \sqrt{\frac{K}{R_0^3}}\right) \tag{9.1.23}$$

will satisfy the differential equations. Thus, this will be the reference trajectory that we linearize about.

We note that we have two second-order differential equations describing the dynamics. Therefore, we must have four state variables in our state model. We choose the usual phase variables as state variables as follows:

$$x_1 = r, \qquad x_2 = \dot{r}$$
$$x_3 = \theta, \qquad x_4 = \dot{\theta} \tag{9.1.24}$$

The nonlinear state equations are then

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = x_1 x_4^2 - \frac{K}{x_1^2} + u_r(t)$$
$$\dot{x}_3 = x_4$$
$$\dot{x}_4 = -\frac{2x_2x_4}{x_1} + \frac{u_\theta(t)}{x_1} \tag{9.1.25}$$

We must now form the $\partial \mathbf{f}/\partial \mathbf{x}$ matrix indicated in Eq. (9.1.8) to get the linearized $\mathbf{F}$ matrix.

$$\frac{\partial \mathbf{f}}{\partial \mathbf{x}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ \left(x_4^2 + \dfrac{2K}{x_1^3}\right) & 0 & 0 & 2x_1x_4 \\ 0 & 0 & 0 & 1 \\ \dfrac{2x_2x_4}{x_1^2} & \dfrac{-2x_4}{x_1} & 0 & \dfrac{-2x_2}{x_1} \end{bmatrix}$$
$$= \begin{bmatrix} 0 & 1 & 0 & 0 \\ \left(\dot{\theta}^2 + \dfrac{2K}{r^3}\right) & 0 & 0 & 2r\dot{\theta} \\ 0 & 0 & 0 & 1 \\ \dfrac{2\dot{r}\dot{\theta}}{r^2} & \dfrac{-2\dot{\theta}}{r} & 0 & \dfrac{-2\dot{r}}{r} \end{bmatrix} \tag{9.1.26}$$

Next, we evaluate $\partial \mathbf{f}/\partial \mathbf{x}$ along the reference trajectory.

$$\left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\substack{r=R_0 \\ \theta=\omega_0 t}} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 3\omega_0^2 & 0 & 0 & 2R_0\omega_0 \\ 0 & 0 & 0 & 1 \\ 0 & \dfrac{-2\omega_0}{R_0} & 0 & 0 \end{bmatrix} \tag{9.1.27}$$

Equation (9.1.27) then defines the $\mathbf{F}$ matrix that characterizes the linearized dynamics. Note that in the linear equations, Δr, $\dot{\Delta} r$, $\Delta\theta$, and $\dot{\Delta}\theta$ become the four state variables.

We now turn to the measurement model. The idealized (no noise) relationships are given by

$$\begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = \begin{bmatrix} \gamma \\ \alpha \end{bmatrix} = \begin{bmatrix} \sin^{-1}\left(\dfrac{R_e}{r}\right) \\ \alpha_0 - \theta \end{bmatrix} \tag{9.1.28}$$

We next replace r with x_1 and θ with x_3, and then perform the partial derivatives indicated by Eq. (9.1.8). The result is

$$\left[\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right] = \begin{bmatrix} -\dfrac{R_e}{r\sqrt{r^2 - R_e^2}} & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix} \tag{9.1.29}$$

Finally, we evaluate $\partial \mathbf{h}/\partial \mathbf{x}$ along the reference trajectory

$$\left.\frac{\partial \mathbf{h}}{\partial \mathbf{x}}\right|_{\substack{r=R_0 \\ \theta=\omega_0 t}} = \begin{bmatrix} -\dfrac{R_e}{R_0\sqrt{R_0^2 - R_e^2}} & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix} \tag{9.1.30}$$

This then becomes the linearized $\mathbf{H}$ matrix of the Kalman filter. The linearized model is now complete with the determination of the $\mathbf{F}$ and $\mathbf{H}$ matrices. Before we leave this example, though, it is worth noting that the forcing function $u_\theta(t)$ must be scaled by $1/R_0$ in the linear model because of the $1/x_1$ factor in Eq. (9.1.25). ■

The Extended Kalman Filter

The extended Kalman filter is similar to a linearized Kalman filter except that the linearization takes place about the filter's estimated trajectory, as shown in

Fig. 9.4, rather than a precomputed nominal trajectory. That is, the partial derivatives of Eq. (9.1.8) are evaluated along a trajectory that has been updated with the filter's estimates; these, in turn, depend on the measurements, so the filter gain sequence will depend on the sample measurement sequence realized on a particular run of the experiment. Thus, the gain sequence is not predetermined by the process model assumptions as in the usual Kalman filter.

A general analysis of the extended Kalman filter is difficult because of the feedback of the measurement sequence into the process model. However, qualitatively it would seem to make sense to update the trajectory that is used for the linearization—after all, why use the old trajectory when a better one is available? The flaw in this argument is this: The "better" trajectory is only better in a *statistical* sense. There is a chance (and maybe a good one) that the updated trajectory will be poorer than the nominal one. In that event, the estimates may be poorer; this, in turn, leads to further error in the trajectory, which causes further errors in the estimates, and so forth and so forth, leading to eventual divergence of the filter. The net result is that the extended Kalman filter is a somewhat riskier filter than the regular linearized filter, especially in situations where the initial uncertainty and measurement errors are large. It may be better *on the average* than the linearized filter, but it is also more likely to diverge in unusual situations.

Both the regular linearized Kalman filter and the extended Kalman filter have been used in a variety of applications. Each has its advantages and disadvantages, and no general statement can be made as to which is best because it depends on the particular situation at hand. Aided inertial navigation systems serve as good examples of both methods of linearization, and this is discussed further in Chapter 10.

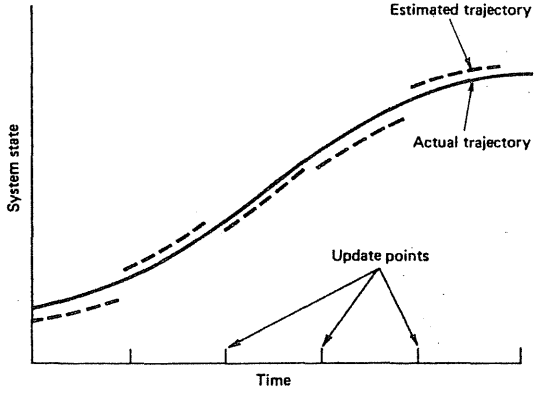


Figure 9.4 Reference and actual trajectories for an extended Kalman filter.

Keeping Track of Total Estimates in an Extended Kalman Filter

It should be remembered that the basic state variables in a linearized Kalman filter are incremental quantities, and not the total quantities such as position, velocity, and so forth. However, in an extended Kalman filter it is usually more convenient to keep track of the total estimates rather than the incremental ones, so we will now proceed to show how this is done and why it is valid to do so.

We begin with the basic linearized measurement equation, Eq. (9.1.11)

$$\mathbf{z} - \mathbf{h}(\mathbf{x}^*) = \mathbf{H}\Delta\mathbf{x} + \mathbf{v} \tag{9.1.31}$$

Note that when working with incremental state variables, the measurement presented to the Kalman filter is $[\mathbf{z} - \mathbf{h}(\mathbf{x}^*)]$ rather than the total measurement $\mathbf{z}$. Next, consider the incremental estimate update equation at time t_k.

$$\Delta\hat{\mathbf{x}}_k = \Delta\hat{\mathbf{x}}_k^- + \mathbf{K}_k \underbrace{[\mathbf{z}_k - \mathbf{h}(\mathbf{x}_k^*) - \mathbf{H}_k\Delta\hat{\mathbf{x}}_k^-]}_{\text{Inc. meas.}} \tag{9.1.32}$$

Now, in forming the measurement residual in Eq. (9.1.32), suppose we associate the $\mathbf{h}(\mathbf{x}_k^*)$ term with $\mathbf{H}_k\Delta\hat{\mathbf{x}}_k^-$ rather than $\mathbf{z}_k$. This measurement residual can then be written as

$$\text{Measurement residual} = (\mathbf{z}_k - \hat{\mathbf{z}}_k^-) \tag{9.1.33}$$

because the predictive estimate of the measurement is just the sum of $\mathbf{h}(\mathbf{x}_k^*)$ and $\mathbf{H}_k\Delta\hat{\mathbf{x}}_k^-$. Note that the measurement residual as given by Eq. (9.1.33) is formed exactly as would be done in an extended Kalman filter, that is, it is the noisy measurement minus the predictive measurement based on the corrected trajectory rather than the nominal one.

We now return to the update equation, Eq. (9.1.32), and add $\mathbf{x}_k^*$ to both sides of the equation:

$$\underbrace{\mathbf{x}_k^* + \Delta\hat{\mathbf{x}}_k}_{\hat{\mathbf{x}}_k} = \underbrace{\mathbf{x}_k^* + \Delta\hat{\mathbf{x}}_k^-}_{\hat{\mathbf{x}}_k^-} + \mathbf{K}_k(\mathbf{z}_k - \hat{\mathbf{z}}_k^-) \tag{9.1.34}$$

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \hat{\mathbf{z}}_k^-) \tag{9.1.35}$$

Equation (9.1.35) is, of course, the familiar linear estimate update equation written in terms of *total* rather than incremental quantities. It simply says that we correct the a priori estimate by adding the measurement residual appropriately weighted by the Kalman gain $\mathbf{K}_k$. Note that after the update is made in the extended Kalman filter, the incremental $\Delta\hat{\mathbf{x}}_k$ is reduced to zero. Its projection to

the next step is then trivial. The only nontrivial projection is to project $\hat{\mathbf{x}}_k$ (which has become the reference $\mathbf{x}$ at t_k) to $\hat{\mathbf{x}}_{k+1}^-$. This must be done through the nonlinear dynamics as dictated by Eq. (9.1.1). That is,

$$\hat{\mathbf{x}}_{k+1}^- = \left\{ \begin{array}{l} \text{Solution of the nonlinear differential equation} \\ \dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}_d, t) \text{ at } t = t_{k+1}, \text{ subject to the} \\ \text{initial condition } \mathbf{x} = \hat{\mathbf{x}}_k \text{ at } t_k \end{array} \right\} \tag{9.1.36}$$

Note that the additive white-noise forcing function $\mathbf{u}(t)$ is zero in the projection step, but the deterministic $\mathbf{u}_d$ is included in the $\mathbf{f}$ function. Once $\hat{\mathbf{x}}_{k+1}^-$ is determined, the predictive measurement $\hat{\mathbf{z}}_{k+1}^-$ can be formed as $\mathbf{h}(\hat{\mathbf{x}}_{k+1}^-)$, and the measurement residual at t_{k+1} is formed as the difference $(\mathbf{z}_{k+1} - \hat{\mathbf{z}}_{k+1}^-)$. The filter is then ready to go through another recursive loop.

For completeness, we repeat the familiar error covariance update and projection equations:

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^- \tag{9.1.37}$$

$$\mathbf{P}_{k+1}^- = \boldsymbol{\phi}_k\mathbf{P}_k\boldsymbol{\phi}_k^T + \mathbf{Q}_k \tag{9.1.38}$$

where $\boldsymbol{\phi}_k$, $\mathbf{H}_k$, and $\mathbf{Q}_k$ come from the linearized model. Equations (9.1.37) and (9.1.38) and the gain equation (which is the same as in the linear Kalman filter) should serve as a reminder that the extended Kalman filter is still working in the world of linear dynamics, even though it keeps track of total estimates rather than incremental ones.

Getting the Extended Kalman Filter Started

It was mentioned previously that the extended Kalman filter can diverge if the reference about which the linearization takes place is poor. The most common situation of this type occurs at the initial starting point of the recursive process. Frequently, the a priori information about the true state of the system is poor. This causes a large error in $\hat{\mathbf{x}}_0^-$ and forces $\mathbf{P}_0^-$ to be large. Thus, two problems can arise in getting the extended filter started:

1. A very large $\mathbf{P}_0^-$ combined with low-noise measurements at the first step will cause the $\mathbf{P}$ matrix to "jump" from a very large value to a small value in one step. In principle, this is permissible. However, this can lead to numerical problems due to roundoff. A non-positive-definite $\mathbf{P}$ matrix at any point in the recursive process usually leads to divergence.
2. The initial $\hat{\mathbf{x}}_0^-$ is presumably the best estimate of $\mathbf{x}$ prior to receiving any measurement information, and thus it is used as the reference for linearization. If the error in $\hat{\mathbf{x}}_0^-$ is large, the first-order approximation used in the linearization will be poor, and divergence may occur, even with perfect arithmetic.

With respect to problem 1, the filter designer should be especially careful to use all the usual numerical precautions to preserve the symmetry and positive definiteness of the $\mathbf{P}$ matrix on the first step. In some cases, simply using the symmetric form of the $\mathbf{P}$-update equation is sufficient to ward off divergence. This form, Eq. (5.5.11), is repeated here for convenience (sometimes called the Joseph form; see ref. 7):

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T \tag{9.1.39}$$

Another way of mitigating the numerical problem is to let $\mathbf{P}_0^-$ be considerably smaller than would normally be dictated by the true a priori uncertainty in $\hat{\mathbf{x}}_0^-$. This will cause suboptimal operation for the first few steps, but this is better than divergence! A similar result can be accomplished by letting $\mathbf{R}_k$ be abnormally large for the first few steps. There is no one single cure for all numerical problems. Each case must be considered on its own merits.

Problem 2 is more subtle than problem 1. Even with perfect arithmetic, poor linearization can cause a poor $\hat{\mathbf{x}}_0^-$ to be updated into an even poorer a posteriori estimate, which in turn gets projected on ahead, and so forth. Various "fixes" have been suggested for the poor-linearization problem, and it is difficult to generalize about them (7, 8, 9, 10). All are ad hoc procedures. This should come as no surprise, because the extended Kalman filter is, itself, an ad hoc procedure. One remedy that works quite well when the information contained in $\mathbf{z}_0$ is sufficient to determine $\mathbf{x}$ algebraically is to use $\mathbf{z}_0$ to solve for $\mathbf{x}$, just as if there were no measurement error. This is usually done with some tried-and-true numerical algorithm such as the Newton–Raphson method of solving algebraic equations (11). It is hoped this will yield a better estimate of $\mathbf{x}$ than the original coarse $\hat{\mathbf{x}}_0^-$. The filter can then be linearized about the new estimate (and a smaller $\mathbf{P}_0^-$ than the original $\mathbf{P}_0^-$ can be used), and the filter is then run as usual beginning with $\mathbf{z}_0$ and with proper accounting for the measurement noise. Another ad hoc procedure that has been used is to let the filter itself iterate on the estimate at the first step. The procedure is fairly obvious. The linearized filter parameters that depend on the reference $\mathbf{x}$ are simply relinearized with each iteration until convergence is reached within some predetermined tolerance. $\mathbf{P}_0^-$ may be held fixed during the iteration, but this need not be the case. Also, if $\mathbf{x}$ is not observable on the basis of just one measurement, iteration may also have to be carried out at a few subsequent steps in order to converge on good estimates of all the elements of $\mathbf{x}$. There is no guarantee that iteration will work in all cases, but it is worth trying.

Before leaving the subject of getting the filter started, it should be noted that neither the algebraic solution nor the iteration remedies just mentioned play any role in the basic "filtering" process. Their sole purpose is simply to provide a good reference for linearization, so that the linearized Kalman filter can do its job of optimal estimation.

9.2 CORRELATED PROCESS AND MEASUREMENT NOISE FOR THE DISCRETE FILTER. DELAYED-STATE FILTER EXAMPLE

It was shown in Section 7.3 that the continuous Kalman filter equations could be modified to accommodate correlated process and measurement noise. We will now consider the corresponding problem for the discrete filter.

Discrete Filter—Correlated Process and Measurement Noise

Just as for the continuous filter, we first define the process and measurement models. They are as follows:

$$\left.\begin{aligned}\mathbf{x}_{k+1} &= \boldsymbol{\phi}_k\mathbf{x}_k + \mathbf{w}_k\\ \mathbf{z}_k &= \mathbf{H}_k\mathbf{x}_k + \mathbf{v}_k\end{aligned}\right\} \quad \text{(as before in Chapter 5)} \qquad \begin{matrix}(9.2.1)\\(9.2.2)\end{matrix}$$

where

$$E[\mathbf{w}_k\mathbf{w}_i^T] = \begin{cases}\mathbf{Q}_k, & i = k\\ 0, & i \neq k\end{cases} \quad \text{(as before in Chapter 5)} \qquad (9.2.3)$$

$$E[\mathbf{v}_k\mathbf{v}_i^T] = \begin{cases}\mathbf{R}_k, & i = k\\ 0, & i \neq k\end{cases} \quad \text{(as before in Chapter 5)} \qquad (9.2.4)$$

and

$$E[\mathbf{w}_{k-1}\mathbf{v}_k^T] = \mathbf{C}_k \quad \text{(new)} \qquad (9.2.5)$$

Before we proceed, an explanation is in order as to why we are concerned with the crosscorrelation of $\mathbf{v}_k$ with $\mathbf{w}_{k-1}$ rather than $\mathbf{w}_k$, which one might expect from just a casual look at the problem. Rewriting Eq. (9.2.1) with k retarded one step should help in this regard:

$$\mathbf{x}_k = \boldsymbol{\phi}_{k-1}\mathbf{x}_{k-1} + \mathbf{w}_{k-1} \qquad (9.2.6)$$

Note that it is $\mathbf{w}_{k-1}$ (and not $\mathbf{w}_k$) that represents the cumulative effect of the white forcing function in the continuous model in the interval (t_{k-1}, t_k). Similarly, $\mathbf{v}_k$ represents the cumulative effect of the white measurement noise in the continuous model when averaged over the same interval (t_{k-1}, t_k) (see Eq. 7.1.10). Therefore, if we wish to have a correspondence between the continuous and discrete models for small Δt, it is the crosscorrelation between $\mathbf{v}_k$ and $\mathbf{w}_{k-1}$ that

we need to include in the discrete model. This is, of course, largely a matter of notation, but an important one.

Now, we could go backward from the continuous equations given in Section 7.3 to get the corresponding discrete equations. However, it is somewhat easier in this case to start over and rederive the discrete recursive equations with proper accounting for the crosscorrelation in the process. We begin with the general update equation.

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-) \qquad (9.2.7)$$

Next, we form the expression for the estimation error.

$$\begin{aligned}\mathbf{e}_k &= \mathbf{x}_k - \hat{\mathbf{x}}_k\\ &= \mathbf{x}_k - [\hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-)]\\ &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{e}_k^- - \mathbf{K}_k\mathbf{v}_k\end{aligned} \qquad (9.2.8)$$

We anticipate now that $\mathbf{e}_k^-$ and $\mathbf{v}_k$ will be correlated, so we will work this out as a side problem:

$$\begin{aligned}E[\mathbf{e}_k^-\mathbf{v}_k^T] &= E[(\mathbf{x}_k - \hat{\mathbf{x}}_k^-)\mathbf{v}_k^T]\\ &= E[(\boldsymbol{\phi}_{k-1}\mathbf{x}_{k-1} + \mathbf{w}_{k-1} - \boldsymbol{\phi}_{k-1}\hat{\mathbf{x}}_{k-1})\mathbf{v}_k^T]\end{aligned} \qquad (9.2.9)$$

Note that $\mathbf{v}_k$ will not be correlated with either $\mathbf{x}_{k-1}$ or $\hat{\mathbf{x}}_{k-1}$ because of its whiteness. Therefore, Eq. (9.2.9) reduces to

$$E[\mathbf{e}_k^-\mathbf{v}_k^T] = E[\mathbf{w}_{k-1}\mathbf{v}_k^T] = \mathbf{C}_k \qquad (9.2.10)$$

We now return to the main derivation. By using Eq. (9.2.8), we form the expression for the $\mathbf{P}_k$ matrix:

$$\begin{aligned}\mathbf{P}_k &= E[\mathbf{e}_k\mathbf{e}_k^T]\\ &= E\{[(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{e}_k^- - \mathbf{K}_k\mathbf{v}_k][(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{e}_k^- - \mathbf{K}_k\mathbf{v}_k]^T\}\end{aligned} \qquad (9.2.11)$$

Now, expanding Eq. (9.2.11) and taking advantage of Eq. (9.2.10) lead to

$$\begin{aligned}\mathbf{P}_k &= (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{P}_k^-(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^T + \mathbf{K}_k\mathbf{R}_k\mathbf{K}_k^T\\ &\quad - (\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)\mathbf{C}_k\mathbf{K}_k^T - \mathbf{K}_k\mathbf{C}_k^T(\mathbf{I} - \mathbf{K}_k\mathbf{H}_k)^T\end{aligned} \qquad (9.2.12)$$

This is a perfectly general expression for the error covariance and is valid for any gain $\mathbf{K}_k$. The last two terms in Eq. (9.2.12) are "new" and involve the crosscorrelation parameter $\mathbf{C}_k$.

We now follow the same procedure used in Section 5.5 to find the optimal gain. We differentiate trace $\mathbf{P}_k$ with respect to $\mathbf{K}_k$ and set the result equal to zero.

The necessary matrix differentiation formulas are given in Section 5.5, and the resulting optimal gain is

$$\mathbf{K}_k = (\mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{C}_k)[\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k + \mathbf{H}_k \mathbf{C}_k + \mathbf{C}_k^T \mathbf{H}_k^T]^{-1} \tag{9.2.13}$$

Note that this expression is similar to the gain formula of Chapter 5 except for the additional terms involving $\mathbf{C}_k$. Let $\mathbf{C}_k$ go to zero, and Eq. (9.2.13) reduces to the same gain as in the zero crosscorrelation model, which is as it should be.

We can now substitute the optimal gain expression, Eq. (9.2.13), into the general $\mathbf{P}_k$ equation, Eq. (9.2.12), to get the a posteriori $\mathbf{P}_k$ equation. After some algebraic manipulation, this leads to either of the two forms:

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k[\mathbf{H}_k \mathbf{P}_k^- \mathbf{H}_k^T + \mathbf{R}_k + \mathbf{H}_k \mathbf{C}_k + \mathbf{C}_k^T \mathbf{H}_k^T]\mathbf{K}_k^T \tag{9.2.14}$$

or

$$\mathbf{P}_k = (\mathbf{I} - \mathbf{K}_k \mathbf{H}_k)\mathbf{P}_k^- - \mathbf{K}_k \mathbf{C}_k^T \tag{9.2.15}$$

The projection equations are not affected by the crosscorrelation between $\mathbf{w}_{k-1}$ and $\mathbf{v}_k$ because of the whiteness property of each. Therefore, the projection equations are (repeated here for completeness)

$$\hat{\mathbf{x}}_{k+1}^- = \boldsymbol{\phi}_k \hat{\mathbf{x}}_k \tag{9.2.16}$$

$$\mathbf{P}_{k+1}^- = \boldsymbol{\phi}_k \mathbf{P}_k \boldsymbol{\phi}_k^T + \mathbf{Q}_k \tag{9.2.17}$$

Equations (9.2.7), (9.2.13), (9.2.15), (9.2.16), and (9.2.17) now comprise the complete set of recursive equations for the correlated process and measurement noise case. We could go one step further at this point and show that the discrete recursive equations will, in fact, go over into the continuous-model differential equations as $\Delta t \to 0$. This is fairly routine, though, so it is left as an exercise (see Problem 9.8).

Delayed-State Measurement Problem

There are numerous dynamical applications where position and velocity are chosen as state variables. It is also common to have *integrated* velocity over some Δt interval as one of the measurements. In some applications the integration is an intrinsic part of the measurement mechanism, and an associated accumulative "count" is the actual measurement that is available to the Kalman filter (e.g., integrated doppler in a digital GPS receiver—see Chapter 11; also see refs. 12 and 13). Other times, integration may be performed on the velocity measurement to presmooth the high-frequency noise. In either case, these measurement situations are described by (in words):

(Discrete measurement observed at time t_k)

$$= \int_{t_{k-1}}^{t_k} (\text{velocity})dt + (\text{discrete noise})$$

$$= (\text{position at } t_k) - (\text{position at } t_{k-1}) + (\text{discrete noise}) \tag{9.2.18}$$

Or, in general mathematical terms, the measurement equation is of the form

$$\mathbf{z}_k = \mathbf{H}_k \mathbf{x}_k + \mathbf{J}_k \mathbf{x}_{k-1} + \mathbf{v}_k \tag{9.2.19}$$

This, of course, does not fit the required format for the usual Kalman filter because of the $\mathbf{x}_{k-1}$ term. In practice, various approximations have been used to accommodate the delayed-state term: some good, some not so good. (One of the poorest approximations is simply to consider the integral of velocity divided by Δt to be a measure of the instantaneous velocity at the end point of the Δt interval.) The correct way to handle the delayed-state measurement problem, though, is to modify the recursive equations so as to accommodate the $\mathbf{x}_{k-1}$ term exactly (14). This can be done with only a modest increase in complexity, as will be seen presently.

We begin by noting that the recursive equation for $\mathbf{x}_k$ can be shifted back one step, that is,

$$\mathbf{x}_k = \boldsymbol{\phi}_{k-1} \mathbf{x}_{k-1} + \mathbf{w}_{k-1} \tag{9.2.20}$$

Equation (9.2.20) can now be rewritten as

$$\mathbf{x}_{k-1} = \boldsymbol{\phi}_{k-1}^{-1} \mathbf{x}_k - \boldsymbol{\phi}_{k-1}^{-1} \mathbf{w}_{k-1} \tag{9.2.21}$$

and this can be substituted into the measurement equation, Eq. (9.2.19), that yields

$$\mathbf{z}_k = \underbrace{(\mathbf{H}_k + \mathbf{J}_k \boldsymbol{\phi}_{k-1}^{-1})\mathbf{x}_k}_{\text{New } \mathbf{H}_k} + \underbrace{(-\mathbf{J}_k \boldsymbol{\phi}_{k-1}^{-1} \mathbf{w}_{k-1} + \mathbf{v}_k)}_{\text{New } \mathbf{v}_k} \tag{9.2.22}$$

Equation (9.2.22) now has the proper form for a Kalman filter, but the new $\mathbf{v}_k$ term is obviously correlated with the process $\mathbf{w}_{k-1}$ term. We can now take advantage of the correlated measurement-process noise equations that were derived in the first part of this section. Before doing so, though, we need to work out the covariance expression for the new $\mathbf{v}_k$ term and also evaluate $\mathbf{C}_k$ for this application.

We will temporarily let the covariance associated with new $\mathbf{v}_k$ be denoted as "New $\mathbf{R}_k$", and it is

$$\text{New } \mathbf{R}_k = E[(-\mathbf{J}_k \boldsymbol{\phi}_{k-1}^{-1} \mathbf{w}_{k-1} + \mathbf{v}_k)(-\mathbf{J}_k \boldsymbol{\phi}_{k-1}^{-1} \mathbf{w}_{k-1} + \mathbf{v}_k)^T] \tag{9.2.23}$$

We note now that $\mathbf{v}_k$ and $\mathbf{w}_{k-1}$ are uncorrelated. Therefore,

$$\text{New } \mathbf{R}_k = \mathbf{J}_k\boldsymbol{\Phi}_{k-1}^{-1}\mathbf{Q}_{k-1}\boldsymbol{\Phi}_{k-1}^{-1T}\mathbf{J}_k^T + \mathbf{R}_k \tag{9.2.24}$$

Also, with reference to Eq. (9.2.5), we can write $\mathbf{C}_k$ as

$$\mathbf{C}_k = E[\mathbf{w}_{k-1}(-\mathbf{J}_k\boldsymbol{\Phi}_{k-1}^{-1}\mathbf{w}_{k-1} + \mathbf{v}_k)^T] = -\mathbf{Q}_{k-1}\boldsymbol{\Phi}_{k-1}^{-1T}\mathbf{J}_k^T \tag{9.2.25}$$

In this application, we can now make the following replacements in Eqs. (9.2.7), (9.2.13), (9.2.14), (9.2.16), and (9.2.17):

$$\mathbf{H}_k \rightarrow \mathbf{H}_k + \mathbf{J}_k\boldsymbol{\Phi}_{k-1}^{-1} \tag{9.2.26}$$

$$\mathbf{R}_k \rightarrow \mathbf{R}_k + \mathbf{J}_k\boldsymbol{\Phi}_{k-1}^{-1}\mathbf{Q}_{k-1}\boldsymbol{\Phi}_{k-1}^{-1T}\mathbf{J}_k^T \tag{9.2.27}$$

$$\mathbf{C}_k \rightarrow -\mathbf{Q}_{k-1}\boldsymbol{\Phi}_{k-1}^{-1T}\mathbf{J}_k^T \tag{9.2.28}$$

where $\rightarrow$ means "is replaced by." After the indicated replacements are made in the recursive equations, the result is a relatively complicated set of equations that involve, among other things, the inverse of $\boldsymbol{\Phi}_{k-1}$. This is a computation that is not required in the usual recursive equations, and it can be eliminated with appropriate algebraic substitutions. The key step is to eliminate $\mathbf{Q}_{k-1}$ by noting that

$$\mathbf{Q}_{k-1} = \mathbf{P}_k^- - \boldsymbol{\Phi}_{k-1}\mathbf{P}_{k-1}\boldsymbol{\Phi}_{k-1}^T \tag{9.2.29}$$

and that the inverse of the transpose is the transpose of the inverse. The final resulting recursive equations for the delayed-state measurement situation can then be written in the form:

Estimate update:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \hat{\mathbf{z}}_k^-) \tag{9.2.30}$$

where

$$\hat{\mathbf{z}}_k^- = \mathbf{H}_k\hat{\mathbf{x}}_k^- + \mathbf{J}_k\hat{\mathbf{x}}_{k-1} \tag{9.2.31}$$

Gain:

$$\mathbf{K}_k = [\mathbf{P}_k^-\mathbf{H}_k^T + \boldsymbol{\Phi}_{k-1}\mathbf{P}_{k-1}\mathbf{J}_k^T][\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k + \mathbf{J}_k\mathbf{P}_{k-1}\boldsymbol{\Phi}_{k-1}^T\mathbf{H}_k^T + \mathbf{H}_k\boldsymbol{\Phi}_{k-1}\mathbf{P}_{k-1}\mathbf{J}_k^T + \mathbf{J}_k\mathbf{P}_{k-1}\mathbf{J}_k^T]^{-1} \tag{9.2.32}$$

Error covariance update:

$$\mathbf{P}_k = \mathbf{P}_k^- - \mathbf{K}_k\mathbf{L}_k\mathbf{K}_k^T \tag{9.2.33}$$

where

$$\mathbf{L}_k = \mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + \mathbf{R}_k + \mathbf{J}_k\mathbf{P}_{k-1}\boldsymbol{\Phi}_{k-1}^T\mathbf{H}_k^T + \mathbf{H}_k\boldsymbol{\Phi}_{k-1}\mathbf{P}_{k-1}\mathbf{J}_k^T + \mathbf{J}_k\mathbf{P}_{k-1}\mathbf{J}_k^T \tag{9.2.34}$$

Projection:

$$\hat{\mathbf{x}}_{k+1}^- = \boldsymbol{\Phi}_k\hat{\mathbf{x}}_k \tag{9.2.35}$$

$$\mathbf{P}_{k+1}^- = \boldsymbol{\Phi}_k\mathbf{P}_k\boldsymbol{\Phi}_k^T + \mathbf{Q}_k \tag{9.2.36}$$

Equations (9.2.30) through (9.2.36) comprise the complete set of recursive equations that must be implemented for the exact (i.e., optimal) solution for the delayed-state measurement problem.* Note that the general form of the equations is the same as for the usual Kalman filter equations. It is just that there are a few additional terms that have to be calculated in the gain and **P**-update expressions. Thus, the extra effort in programming the exact equations is quite modest. (See Problem 9.5 for a demonstration that these recursive equations really do what they purport to do.)

9.3 ADAPTIVE KALMAN FILTER (MULTIPLE MODEL ADAPTIVE ESTIMATOR)

In the usual Kalman filter we assume that all the process parameters, that is, $\boldsymbol{\Phi}_k$, $\mathbf{H}_k$, $\mathbf{R}_k$, and $\mathbf{Q}_k$, are known. They may vary with time (index k) but, if so, the nature of the variation is assumed to be known. In physical problems this is often a rash assumption. There may be large uncertainty in some parameters because of inadequate prior test data about the process. Or some parameter might be expected to change slowly with time, but the exact nature of the change is not predictable. In such cases, it is highly desirable to design the filter to be self-learning, so that it can adapt itself to the situation at hand, whatever that might be. This problem has received considerable attention since Kalman's original papers of the early 1960s. However, it is not an easy problem with one simple solution. This is evidenced by the fact that 35 years later we still see many papers on the subject in current control system journals. (Reference 21 gives a good list of recent papers on adaptive control.)

We will concentrate our attention here on an adaptive filter scheme that was first presented by D. T. Magill (16). Also a more intuitive scheme is discussed in Problem 9.3. We will see presently that Magill's adaptive filter is not just one filter but, instead, is a whole bank of Kalman filters running in parallel. At the time that this scheme was first suggested in 1965, it was considered to be impractical for implementation on-line. However, the spectacular advances in computer technology over the past few decades have made Magill's parallel-filter scheme quite feasible in a number of applications (17, 18, 19, 20, 21, 22).

* It is of interest to note that Eqs. (9.2.30) through (9.2.36) can also be derived by a completely different method. See Section 9.4 of (15).

Because of the parallel bank of filters, this scheme is usually referred to as the multiple model adaptive estimator (MMAE). In the interest of simplicity, we will confine our attention here to Magill's original MMAE scheme in its primitive form. It is worth mentioning that there have been many extensions and variations on the original scheme since 1965, including recent papers by Caputi (23) and Blair and Bar-Shalom (22). (These are interesting papers, both for their technical content and the references contained therein.) We will now proceed to the derivation that leads to the bank of parallel filters.

We begin with the simple statement that the desired estimator is to be the conditional mean given by

$$\hat{\mathbf{x}}_k = \int_{\mathbf{x}} \mathbf{x}\, p(\mathbf{x}|\mathbf{z}_k^*)\, d\mathbf{x} \tag{9.3.1}$$

where $\mathbf{z}_k^*$ denotes all the measurements up to and including time t_k (i.e., $\mathbf{z}_1$, $\mathbf{z}_2$, . . . , $\mathbf{z}_k$), and $p(\mathbf{x}|\mathbf{z}_k^*)$ is the probability density function of $\mathbf{x}_k$ with the conditioning shown in parentheses.* The indicated integration is over the entire $\mathbf{x}$ space. If the $\mathbf{x}$ and $\mathbf{z}$ processes are Gaussian, we are assured that the estimate given by Eq. (9.3.1) will be optimal by almost any reasonable criterion of optimality, least-mean-square or otherwise (24). We also wish to assume that some parameter of the process, say, α, is unknown to the observer, and that this parameter is a random variable (not necessarily Gaussian). Thus, on any particular sample run it will be an unknown constant, but with a known statistical distribution. Hence, rather than beginning with $p(\mathbf{x}|\mathbf{z}_k^*)$, we really need to begin with the joint density $p(\mathbf{x}, \alpha|\mathbf{z}_k^*)$ and sum out on α to get $p(\mathbf{x}|\mathbf{z}_k^*)$. Thus, we will rewrite Eq. (9.3.1) in the form

$$\hat{\mathbf{x}}_k = \int_{\mathbf{x}} \mathbf{x} \int_{\alpha} p((\mathbf{x}, \alpha)|\mathbf{z}_k^*)\, d\alpha\, d\mathbf{x} \tag{9.3.2}$$

But the joint density in Eq. (9.3.2) can be written as

$$p(\mathbf{x}, \alpha|\mathbf{z}_k^*) = p(\mathbf{x}|\alpha, \mathbf{z}_k^*)p(\alpha|\mathbf{z}_k^*) \tag{9.3.3}$$

Substituting Eq. (9.3.3) into (9.3.2) and interchanging the order of integration lead to

$$\hat{\mathbf{x}}_k = \int_{\alpha} p(\alpha|\mathbf{z}_k^*) \int_{\mathbf{x}} \mathbf{x}\, p(\mathbf{x}|\alpha, \mathbf{z}_k^*)\, d\mathbf{x}\, d\alpha \tag{9.3.4}$$

The inner integral will be recognized as just the usual Kalman filter estimate for

* Throughout this section we will use a looser notation than that used in Chapter 1 in that p will be used for both probability density and discrete probability. In this way we avoid the multitudinous subscripts that would otherwise be required for conditioned multivariate random variables. However, this means that the student must use a little imagination and interpret the symbol p properly within the context of its use in any particular derivation.

a given α. This is denoted as $\hat{\mathbf{x}}_k(\alpha)$ where α shown in parentheses is intended as a reminder that there is α dependence. Equation (9.3.4) may now be rewritten as

$$\hat{\mathbf{x}}_k = \int_{\alpha} \hat{\mathbf{x}}_k(\alpha)p(\alpha|\mathbf{z}_k^*)\, d\alpha \tag{9.3.5}$$

Or the discrete random variable equivalent to Eq. (9.3.5) would be

$$\hat{\mathbf{x}}_k = \sum_{i=1}^{L} \hat{\mathbf{x}}_k(\alpha_i)p(\alpha_i|\mathbf{z}_k^*) \tag{9.3.6}$$

where $p(\alpha_i|\mathbf{z}_k^*)$ is the discrete probability for α_i, conditioned on the measurement sequence $\mathbf{z}_k^*$. We will concentrate on the discrete form from this point on in our discussion.

Equation (9.3.6) simply says that the optimal estimate is a weighted sum of Kalman filter estimates with each Kalman filter operating with a separate assumed value of α. This is shown in Fig. 9.5. The problem now reduces to one of determining the weight factors $p(\alpha_1|\mathbf{z}_k^*)$, $p(\alpha_2|\mathbf{z}_k^*)$, etc. These, of course, change with each recursive step as the measurement process evolves in time. Presumably, as more and more measurements become available, we learn more about the state of the process and the unknown parameter α. (Note that it is constant for any particular sample run of the process.)

We now turn to the matter of finding the weight factors indicated in Fig. 9.5. Toward this end we use Bayes' rule:

$$p(\alpha_i|\mathbf{z}_k^*) = \frac{p(\mathbf{z}_k^*|\alpha_i)p(\alpha_i)}{p(\mathbf{z}_k^*)} \tag{9.3.7}$$

But

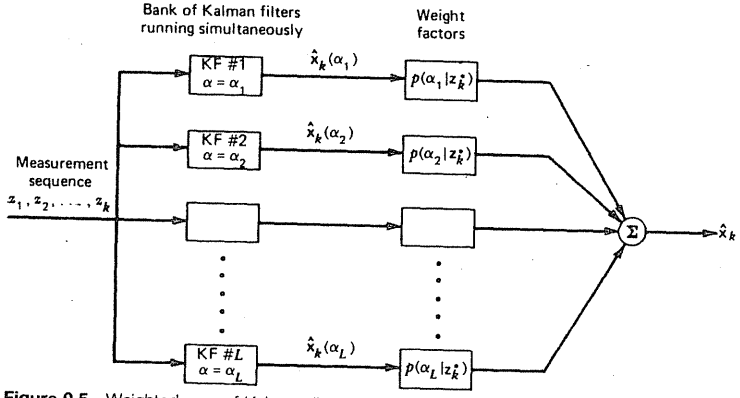


Figure 9.5 Weighted sum of Kalman filter estimates.

$$p(\mathbf{z}_k^*) = \sum_{j=1}^{L} p(\mathbf{z}_k^*, \alpha_j)$$

$$= \sum_{j=1}^{L} p(\mathbf{z}_k^*|\alpha_j)p(\alpha_j) \tag{9.3.8}$$

Equation (9.3.8) may now be substituted into Eq. (9.3.7) with the result

$$p(\alpha_i|\mathbf{z}_k^*) = \left[\frac{p(\mathbf{z}_k^*|\alpha_i)p(\alpha_i)}{\sum_{j=1}^{L} p(\mathbf{z}_k^*|\alpha_j)p(\alpha_j)}\right], \qquad i = 1, 2, \ldots, L \tag{9.3.9}$$

The distribution $p(\alpha_i)$ is presumed to be known, so it remains to determine $p(\mathbf{z}_k^*|\alpha_i)$ in Eq. (9.3.9). Toward this end we will write $p(\mathbf{z}_k^*|\alpha_i)$ as a product of conditional density functions. Temporarily omitting the α_i conditioning (just to save writing), we have

$$\begin{aligned} p(\mathbf{z}_k^*) &= p(\mathbf{z}_k, \mathbf{z}_{k-1}, \ldots \mathbf{z}_0) \\ &= p(\mathbf{z}_k, \mathbf{z}_{k-1}, \ldots \mathbf{z}_1|\mathbf{z}_0)p(\mathbf{z}_0) \\ &= p(\mathbf{z}_k, \mathbf{z}_{k-1}, \ldots \mathbf{z}_2|\mathbf{z}_1, \mathbf{z}_0)p(\mathbf{z}_1|\mathbf{z}_0)p(\mathbf{z}_0) \\ &\vdots \\ &= p(\mathbf{z}_k|\mathbf{z}_{k-1}, \mathbf{z}_{k-2}, \ldots \mathbf{z}_0)p(\mathbf{z}_{k-1}|\mathbf{z}_{k-2}, \mathbf{z}_{k-3}, \ldots \mathbf{z}_0) \ldots p(\mathbf{z}_1|\mathbf{z}_0)p(\mathbf{z}_0), \\ &\qquad k = 1, 2, \ldots \end{aligned} \tag{9.3.10}$$

We now note that the first term in the product string of Eq. (9.3.10) is just $p(\hat{\mathbf{z}}_k^-)$, and that the remaining product is just $p(\mathbf{z}_{k-1}^*)$. Thus, we can rewrite Eq. (9.3.10) in the form

$$p(\mathbf{z}_k^*) = p(\hat{\mathbf{z}}_k^-)p(\mathbf{z}_{k-1}^*) \tag{9.3.11}$$

We now make the Gaussian assumption for the $\mathbf{x}$ and $\mathbf{z}$ processes (but not for α). Also, to simplify matters, we will assume $\mathbf{z}_k^*$ to be a sequence of scalar measurements $z_0, z_1, \ldots z_k$. Equation (9.3.11) then becomes

$$p(\mathbf{z}_k^*) = \frac{1}{(2\pi)^{1/2}(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + R_k)^{1/2}} \exp\left[-\frac{1}{2}\frac{(z_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-)^2}{(\mathbf{H}_k\mathbf{P}_k^-\mathbf{H}_k^T + R_k)}\right] p(\mathbf{z}_{k-1}^*), \qquad k = 1, 2, \ldots \tag{9.3.12}$$

Bear in mind that $p(\mathbf{z}_k^*)$ will, in general, be different for each α_i. For example, if the unknown parameter is R_k, each filter in the bank of filters will be modeled around a different value for R_k.

It should be helpful at this point to go through an example step by step (in words, at least) to see how the parallel bank of filters works.

1. We begin with the prior distribution of α and set the filter weight factors accordingly. Frequently, we have very little prior knowledge of the unknown parameter α. In this case we would assume a uniform probability distribution and set all the weight factors equal initially. This does not have to be the case in general, though.
2. The initial prior estimates $\hat{\mathbf{x}}_0^-$ for each filter are set in accordance with whatever prior information is available. Usually, if $\mathbf{x}$ is a zero-mean process, $\hat{\mathbf{x}}_0^-$ is simply set equal to zero for each filter. We will assume that this is true in this example.
3. Usually, the initial estimate uncertainty does not depend on α, so the initial $\mathbf{P}_0^-$ for each filter will just be the covariance matrix of the $\mathbf{x}$ process.
4. Initially then, before any measurements are received, the prior estimates from each of the filters are weighted by the initial weight factors and summed to give the optimal prior estimate from the bank of filters. In the present example this is trivial, because the initial estimates for each filter were set to zero.
5. At $k = 0$ the bank of filters receives the first measurement z_0, and the unconditional $p(z_0)$ must be computed for each permissible α_i. We note that $z = \mathbf{H}\mathbf{x} + v$, so $p(z_0)$ may be written as

$$p(z_0) = \frac{1}{(2\pi)^{1/2}(\mathbf{H}_0\mathbf{C}_x\mathbf{H}_0^T + R_0)^{1/2}} \exp\left[-\frac{1}{2}\frac{z_0^2}{(\mathbf{H}_0\mathbf{C}_x\mathbf{H}_0^T + R_0)}\right] \tag{9.3.13}$$

   where $\mathbf{C}_x$ is the covariance matrix of $\mathbf{x}$. We note again that one or more of the parameters in Eq. (9.3.13) may have α dependence; thus, in general, $p(z_0)$ will be different for each α_i.
6. Once $p(z_0)$ for each α_i has been determined, Eq. (9.3.9) may be used to find $p(\alpha_i|z_0)$. These are the weight factors to be used in summing the updated estimates $\hat{\mathbf{x}}_0$ that come out of each of the filters in the bank of filters. This then yields the optimal adaptive estimate, given the measurement z_0, and we are ready to project on to the next step.
7. Each of the individual Kalman filter estimates and their error covariances is projected ahead to $k = 1$ in the usual manner. The adaptive filter must now compute $p(\mathbf{z}_1^*)$ for each α_i, and it uses the recursive formula, Eq. (9.3.12), in doing so. Therefore, for $p(\mathbf{z}_1^*)$ we have

$$p(\mathbf{z}_1^*) = \frac{1}{(2\pi)^{1/2}(\mathbf{H}_1\mathbf{P}_1^-\mathbf{H}_1^T + R_1)^{1/2}} \exp\left[-\frac{1}{2}\frac{(z_1 - \mathbf{H}_1\hat{\mathbf{x}}_1^-)^2}{(\mathbf{H}_1\mathbf{P}_1^-\mathbf{H}_1^T + R_1)}\right] p(\mathbf{z}_0^*) \tag{9.3.14}$$

   Note that $p(\mathbf{z}_0^*)$ was computed in the previous step, and the prior $\hat{\mathbf{x}}_1^-$ and $\mathbf{P}_1^-$ for each α_i are obtained from the projection step.

8. Now, the $p(\mathbf{z}_1^*)$ determined in step 7 can be used in Bayes' formula, Eq. (9.3.9), and the weight factors $p(\alpha_i|\mathbf{z}_1^*)$ for $k = 1$ are thus determined. It should be clear now that this recursive procedure can be carried on *ad infinitum*.

We are now in a position to reflect on the whole adaptive filter in perspective. At each recursive step the adaptive filter does three things: (1) Each filter in the bank of filters computes its own estimate, which is hypothesized on its own model; (2) the system computes the a posteriori probabilities for each of the hypotheses; and (3) the scheme forms the adaptive optimal estimate of $\mathbf{x}$ as a weighted sum of the estimates produced by each of the individual Kalman filters. As the measurements evolve with time, the adaptive scheme learns which of the filters is the correct one, and its weight factor approaches unity while the others are going to zero. The bank of filters accomplishes this, in effect, by looking at sums of the weighted squared measurement residuals. The filter with the smallest residuals "wins," so to speak.

The Magill scheme just described is not without some problems and limitations (23). It is still important, though, because it is optimum (within the various assumptions that were made), and it serves as a point of departure for other less rigorous schemes. One of the problems of this technique has to do with numerical behavior as the number of steps becomes large. Clearly, as the sum of the squared measurement residuals becomes large, there is a possibility of computer underflow. Also, note that the unknown parameter was assumed to be constant with time. Thus, there is no way this kind of adaptive filter can readjust if the parameter actually varies slowly with time. This adaptive scheme, in its purest form, never forgets; it tends to give early measurements just as much weight as later ones in its determination of the a posteriori probabilities. Some ad hoc procedure, such as periodic reinitialization, has to be used if the scheme is to adapt to a slowly varying parameter situation.

By its very nature, the Magill adaptive filter is a transient scheme, and it converges to the correct α_i in an optimal manner (provided, of course, that the various assumptions are valid). This is one of the scheme's strong points, and it gives rise to a class of applications that are usually not thought of as adaptive filter problems. These are applications where we are more interested in the a posteriori probabilities than in the optimal estimate of the vector $\mathbf{x}$ process. The Magill scheme is an excellent multiple-hypothesis testor in the presence of Gauss–Markov noise. In this setting the main objective of the bank of filters is to determine which hypothesized model in the bank of filters is the correct one, and there have been a number of applications of this type reported in the literature (18, 19, 20). We will discuss one of these briefly at the end of this section. Before doing so, though, we wish to emphasize the assumptions that were used in deriving Magill's parallel filter scheme. These assumptions are most important in understanding the limitations of the scheme. The key assumptions are as follows:

1. We assume Gaussian statistics for the $\mathbf{x}$ and $\mathbf{z}$ processes, but not for α.
2. The unknown parameter α is assumed to be a discrete random variable with a known distribution. Thus, for any sample run, α will be constant with time. If, in fact, α is an unknown *deterministic* parameter (in contrast to being random), then we must be cautious about our conclusions relative of optimality. (See Example 6.6, Chapter 6, for further discussion of this.)
3. It is tacitly assumed that the process and measurement models are of the proper form for a linear Kalman filter for each allowable α_i. (Note that finite-order state models will not exist for all random processes, band-limited white noise being a good example.)

We will now look at an application of the Magill adaptive filter where both the hypothesis and the optimal estimate are of interest.

EXAMPLE 9.3

We will now return to the power system relaying application that was discussed in Section 6.4. Recall that the problem there was one of estimating post-fault currents and voltages on a transmission line. The objective of the Kalman filter was to estimate the steady-state components based on a short observation span during the transient period immediately after the fault occurrence. We will now look at fault classification, which is a different facet of the overall power system relaying problem.

Classification of a fault on a three-phase transmission line within a few milliseconds after the fault is not as simple as it may seem at first glance. This is because a fault on any one line induces an imbalance that reflects into the unfaulted phases as well as the faulted one. However, the statistics of the voltage noises after the fault are significantly different for the faulted and unfaulted phases. For example, Girgis and Brown (18) found by extensive simulation that the measurement R_k parameters for the two cases could be approximated by

$$R_k(\text{faulted phase}) = 0.6e^{-k\Delta t/T_1}$$

$$R_k(\text{unfaulted phase}) = 0.1e^{-k\Delta t/T_1}$$

where R_k is in units of (per unit voltage)^2. Clearly, this is a significant difference. The measurement-noise differences, along with differences in initial conditions, can then be used to discriminate between the faulted and unfaulted conditions. A block diagram for accomplishing this with a Magill adaptive filter is shown in Fig. 9.6. In the scheme described by Girgis and Brown (18) there would be three such adaptive filters, one for each of the three phases. The results coming out of each of the filters are then combined to determine the type of fault (e.g., line a to ground, line b to ground, etc.) and the optimal voltage estimates.

Most of the details of the referenced scheme will be omitted here because they are adequately covered in the reference. We will simply say that in all of the simulations conducted, the adaptive scheme worked quite well in identifying the faulted phase (or phases); and in all instances a firm identification was made within about 2 msec. Figure 9.7 shows the probability calculations for a typical run with a three-phase fault applied to the line. Only the probabilities associated with the "correct" Kalman filters are shown in the figure. The corresponding

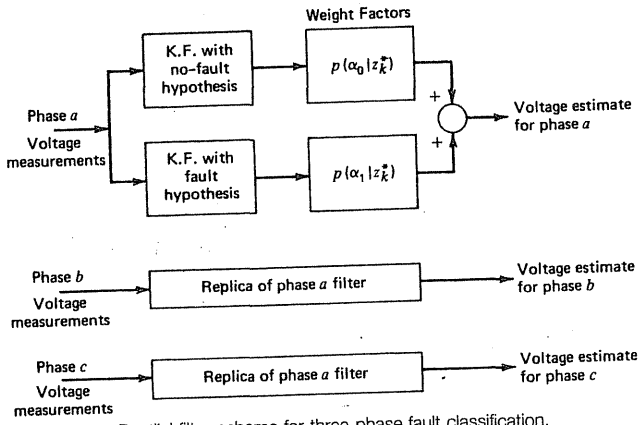


Figure 9.6 Parallel filter scheme for three-phase fault classification.

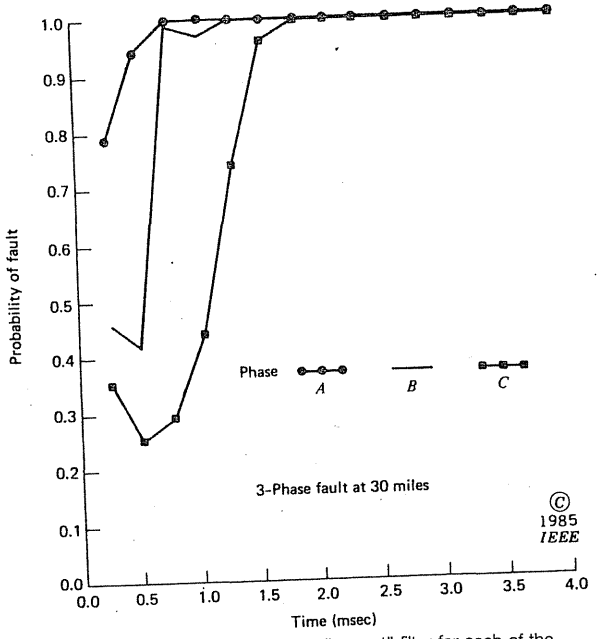


Figure 9.7 Probability profiles for the "correct" filter for each of the three phases.

probabilities for the no-fault filters would just be the complements of the ones shown. Note that the plots for each of the phases are not the same, even though the fault is symmetric in this case. This is to be expected, though, because the instantaneous voltages on the three lines at the time of the fault are not the same. In effect, some of the resulting voltage profiles match the assumed fault model better than others; thus, the better the match, the faster the convergence. However, in spite of the differences in the probability responses for the three phases, all of them converge to the correct decision within 2 msec. This yields very fast fault classification.

Not only does the adaptive filter shown in Fig. 9.6 make a fast decision as to fault or no-fault, but it also estimates the post-fault voltage in the process. Of course, the estimate produced by the "winning" filter is the one of primary interest. After the decision is made as to the correct hypothesis, the other estimate can be ignored. The advantage of this type of operation over the nonadaptive scheme presented in Section 6.4 lies in the individualized modeling that is permitted in the adaptive scheme. There, each of the two filter models can be tailored to fit its own hypothesis, whereas in a nonadaptive scheme the model must be an all-purpose model that fits, to a degree at least, a wider class of situations. ■

9.4 SCHMIDT–KALMAN FILTER. REDUCING THE ORDER OF THE STATE VECTOR

In many real-life applications the size of the state vector can be a problem, and it becomes necessary to eliminate some of the elements to make the filter computationally feasible in real time. This is true for integrated navigation system applications, more often than not. Recall that when colored measurement noises are present, they must be absorbed as elements of the state vector to fit the required format for a Kalman filter (see Section 5.7). Estimates of these noise components are usually not of primary interest, and they are often only weakly observable. Thus, they immediately become candidates for elimination if the need arises. This cannot be done and maintain true optimality, but there is a way of at least partially compensating for the missing state variables in the resulting reduced-order filter. The technique that we will look at here was first suggested by S. F. Schmidt (25), and the resulting filter algorithm is often referred to as the Schmidt–Kalman filter. It is also known as a *consider* filter, because the missing states are "considered" but not actually implemented in the filter. We will concentrate our attention here only on the case where the state variables to be eliminated are colored measurement noises (including true biases).

We will begin the development of the Schmidt–Kalman filter by writing the total state vector for the truth model in partitioned form.

$$\mathbf{X}_k = \begin{bmatrix} \mathbf{x} \\ \hline \mathbf{y} \end{bmatrix}_k = \left[\begin{array}{c} x_1 \\ x_2 \\ \vdots \\ x_n \\ \hline y_1 \\ y_2 \\ \vdots \\ y_r \end{array}\right]_k \begin{array}{l} \left.\begin{array}{l} \\ \\ \\ \\ \end{array}\right\} \text{Portion of the state vector to be retained and estimated} \\ \left.\begin{array}{l} \\ \\ \\ \\ \end{array}\right\} \text{Part due to colored measurement noises that is to be eliminated} \end{array} \tag{9.4.1}$$

Next, we write the process and measurement equations for the truth model in partitioned form.

Process model:

$$\begin{bmatrix} \mathbf{x} \\ \hline \mathbf{y} \end{bmatrix}_{k+1} = \begin{bmatrix} \boldsymbol{\Phi}_x & \mathbf{0} \\ \hline \mathbf{0} & \boldsymbol{\Phi}_y \end{bmatrix}_k \begin{bmatrix} \mathbf{x} \\ \hline \mathbf{y} \end{bmatrix}_k + \begin{bmatrix} \mathbf{w}_x \\ \hline \mathbf{w}_y \end{bmatrix}_k \tag{9.4.2}$$

Measurement model:

$$\mathbf{z}_k = [\mathbf{H} \mid \mathbf{J}]_k \begin{bmatrix} \mathbf{x} \\ \hline \mathbf{y} \end{bmatrix}_k + \mathbf{v}_k \tag{9.4.3}$$

The error covariance associated with $\mathbf{X}$ will also be written in partitioned form as

$$[\mathbf{P}_X]_k = \begin{bmatrix} \mathbf{P}_x & \mathbf{P}_{xy} \\ \hline \mathbf{P}_{yx} & \mathbf{P}_y \end{bmatrix}_k \tag{9.4.4}$$

Note from Eq. (9.4.2) that $\mathbf{x}$ and $\mathbf{y}$ are completely decoupled. Also, as usual we will assume that $(\mathbf{w}_x)_k$, $(\mathbf{w}_y)_k$, and $\mathbf{v}_k$ are white sequences with zero crosscorrelation and with covariances $(\mathbf{Q}_x)_k$, $(\mathbf{Q}_y)_k$, and $\mathbf{R}_k$.

We now need to write the expression for the optimal gain at step k in partitioned form, and at this point we will temporarily drop the k subscripts just to save writing. We use the usual optimal gain formula:

$$\begin{bmatrix} \mathbf{K}_x \\ \hline \mathbf{K}_y \end{bmatrix} = \begin{bmatrix} \mathbf{P}_x^- & \mathbf{P}_{xy}^- \\ \hline \mathbf{P}_{yx}^- & \mathbf{P}_y^- \end{bmatrix} \begin{bmatrix} \mathbf{H}^T \\ \hline \mathbf{J}^T \end{bmatrix} \left[[\mathbf{H} \mid \mathbf{J}] \begin{bmatrix} \mathbf{P}_x^- & \mathbf{P}_{xy}^- \\ \hline \mathbf{P}_{yx}^- & \mathbf{P}_y^- \end{bmatrix} \begin{bmatrix} \mathbf{H}^T \\ \hline \mathbf{J}^T \end{bmatrix} + \mathbf{R} \right]^{-1} \tag{9.4.5}$$

It is a routine matter now to expand out Eq. (9.4.5) and obtain explicit expressions for $\mathbf{K}_x$ and $\mathbf{K}_y$. However, we anticipate that $\mathbf{K}_y$ will not be needed in the reduced-order filter because the $\mathbf{y}$ variables will not be present. Hence, we will arbitrarily assume that the estimates of the $\mathbf{y}$ components are always zero (and invisible) in the reduced-order filter. Furthermore, we will use $\mathbf{K}_x$ in Eq. (9.4.5) as the gain for the remaining $\mathbf{x}$ state variables. This is, of course, the optimal gain, subject to the optimality of $\mathbf{P}_x^-$, $\mathbf{P}_{xy}^-$, and $\mathbf{P}_y^-$, which came from the previous step. Now, setting the $\mathbf{y}$ estimates equal to zero can be effected in the full-order filter by arbitrarily letting $\mathbf{K}_y$ be zero at each step, rather than the optimum value as determined by Eq. (9.4.5). If we do this, we then have a suboptimal full-order, truth-model filter, and we can use suboptimal error analysis to evaluate the error covariance (see Section 6.7). In brief, this involves substituting the suboptimal gain into the general expression for the error covariance, Eq. (5.5.11) (repeated here in the notation of Chapter 5):

$$\mathbf{P} = (\mathbf{I} - \mathbf{KH})\mathbf{P}^-(\mathbf{I} - \mathbf{KH})^T + \mathbf{KRK}^T \tag{9.4.6}$$

Or, for the problem at hand, we have for the full-order suboptimal error covariance:

$$\begin{aligned} \begin{bmatrix} \mathbf{P}_x & \mathbf{P}_{xy} \\ \hline \mathbf{P}_{yx} & \mathbf{P}_y \end{bmatrix} = & \left[\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{I} \end{bmatrix} - \begin{bmatrix} \mathbf{K}_x \\ \hline \mathbf{0} \end{bmatrix} [\mathbf{H} \mid \mathbf{J}] \right] \\ & \times \begin{bmatrix} \mathbf{P}_x^- & \mathbf{P}_{xy}^- \\ \hline \mathbf{P}_{yx}^- & \mathbf{P}_y^- \end{bmatrix} \left[\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \hline \mathbf{0} & \mathbf{I} \end{bmatrix} - \begin{bmatrix} \mathbf{K}_x \\ \hline \mathbf{0} \end{bmatrix} [\mathbf{H} \mid \mathbf{J}] \right]^T \\ & + \begin{bmatrix} \mathbf{K}_x \\ \hline \mathbf{0} \end{bmatrix} \mathbf{R} [\mathbf{K}_x^T \mid \mathbf{0}] \end{aligned} \tag{9.4.7}$$

This equation can now be expanded out, and explicit expressions for updating $\mathbf{P}_x^-$, $\mathbf{P}_{xy}^-$, and $\mathbf{P}_y^-$ can be obtained.

We will summarize the recursive equations for the reduced-order (Schmidt–Kalman) filter.

1. The gain expression (from Eq. 9.4.5):

$$\mathbf{K}_x = (\mathbf{P}_x^-\mathbf{H}^T + \mathbf{P}_{xy}^-\mathbf{J}^T)\boldsymbol{\alpha}^{-1} \tag{9.4.8}$$

where

$$\boldsymbol{\alpha} = \mathbf{HP}_x^-\mathbf{H}^T + \mathbf{HP}_{xy}^-\mathbf{J}^T + \mathbf{JP}_{xy}^-\mathbf{H}^T + \mathbf{JP}_y^-\mathbf{J}^T + \mathbf{R} \tag{9.4.9}$$

2. Update $\hat{\mathbf{x}}^-$ (the usual equation because $\hat{\mathbf{y}}$ is zero):

$$\hat{\mathbf{x}} = \hat{\mathbf{x}}^- + \mathbf{K}_x(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}^-) \tag{9.4.10}$$

3. Covariance updates (from Eq. 9.4.7):

$$\begin{aligned}
\mathbf{P}_x &= (\mathbf{I} - \mathbf{K}_x\mathbf{H})\mathbf{P}_x^- - \mathbf{K}_x\mathbf{J}\mathbf{P}_{yx}^- \\
\mathbf{P}_{xy} &= (\mathbf{I} - \mathbf{K}_x\mathbf{H})\mathbf{P}_{xy}^- - \mathbf{K}_x\mathbf{J}\mathbf{P}_y^- \\
\mathbf{P}_{yx} &= \mathbf{P}_{xy}^T \\
\mathbf{P}_y &= \mathbf{P}_y^-
\end{aligned} \tag{9.4.11}$$

4. Projection equations (with the k subscripts reinserted):
   (a) The state estimate (recall that x and y are decoupled):

$$\hat{\mathbf{x}}_{k+1}^- = (\boldsymbol{\phi}_x)_k\hat{\mathbf{x}}_k \tag{9.4.12}$$

   (b) The error covariances (from the usual $\boldsymbol{\phi}\mathbf{P}\boldsymbol{\phi}^T + \mathbf{Q}$ expression written in partitioned form):

$$\begin{aligned}
(\mathbf{P}_x^-)_{k+1} &= (\boldsymbol{\phi}_x)_k(\mathbf{P}_x)_k(\boldsymbol{\phi}_x)_k^T + (\mathbf{Q}_x)_k \\
(\mathbf{P}_{xy}^-)_{k+1} &= (\boldsymbol{\phi}_x)_k(\mathbf{P}_{xy})_k(\boldsymbol{\phi}_y)_k^T \\
(\mathbf{P}_{yx}^-)_{k+1} &= (\mathbf{P}_{xy}^-)_{k+1}^T \\
(\mathbf{P}_y^-)_{k+1} &= (\boldsymbol{\phi}_y)_k(\mathbf{P}_y)_k(\boldsymbol{\phi}_y)_k^T + (\mathbf{Q}_y)_k
\end{aligned} \tag{9.4.13}$$

It should be apparent now that the recursive equations for the Schmidt–Kalman filter are more complicated than for an "ordinary" Kalman filter. Hence, one might logically ask: Where is the saving? The answer lies in the simplification that comes from setting $\mathbf{K}_y = 0$ and not having to carry $\mathbf{y}$ along in the recursive equations. The gain and update equations, bad as they look, would be even more complicated were $\mathbf{K}_y$ not equal to zero. The projection equations are really not simplified by setting $\mathbf{K}_y$ equal to zero. The simplification there comes from the block diagonal character of the transition matrix and not from anything that was done in the gain computation. A simple example will now be presented that compares the effectiveness of the Schmidt–Kalman filter with other ways of reducing the state vector.

EXAMPLE 9.4

Consider a simple example where we wish to estimate a random walk process on the basis of a sequence of noisy measurements that are uniformly spaced by 1 sec. In order to establish common initial conditions for the three filters that are to be compared, we will assume that an optimal 1-state filter with white measurement noise has been operating and is in steady state prior to $t = 0$ (i.e., $k = 0$). The $\mathbf{R}_k$ and $\mathbf{Q}_k$ parameters during this period are

$$\mathbf{R}_k = 1, \qquad \mathbf{Q}_k = 1$$

In the steady-state condition, it is easily verified that $\mathbf{P}_k^-$ and $\mathbf{P}_k$ are

$$\mathbf{P}_k^- = 1.618034, \qquad \mathbf{P}_k = .618034, \quad \text{for } k < 0$$

Now at $k = 0$ we will assume that an additional bias component that is $N(0, 1)$ is added to the already present white measurement noise. This changes the truth model from a 1-state process to a 2-state process. (The bias must be added to the state vector to account for it exactly—see Section 5.7.) Thus, for $k = 0, 1, 2, \ldots$, the truth model becomes

$$\boldsymbol{\phi}_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{9.4.14}$$

$$\mathbf{Q}_k = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \tag{9.4.15}$$

$$\mathbf{H}_k = [1 \quad 1] \tag{9.4.16}$$

$$\mathbf{R}_k = [1] \tag{9.4.17}$$

$$\mathbf{P}_0^- = \begin{bmatrix} 1.618034 & 0 \\ 0 & 1.0 \end{bmatrix} \tag{9.4.18}$$

We will now compare the estimation errors for three different filters for 20 steps, beginning at $k = 0$.

Optimal 2-State Filter Error covariance analysis for the optimal 2-state filter is routine, because the usual Kalman filter equations developed in Chapter 5 can be used for the analysis. A new steady-state condition after $k = 0$ is reached in about 10 steps, and the resulting error variance plot for state variable 1 (i.e., the random walk process) is shown in Fig. 9.8. Presumably, this is the best that we can hope to do under the circumstances.

Schmidt–Kalman Filter Here, we wish to "consider" the bias state (state variable 2 in the truth model), but we do not want to actually implement it in the filter. We can use the methods just developed in this section to analyze such a reduced-order filter. We will assume that a Schmidt–Kalman filter is implemented. In this example, the following parameters are applicable (in the notation used earlier in this section):

$$\boldsymbol{\phi}_x = 1.0, \qquad \boldsymbol{\phi}_y = 1.0 \tag{9.4.19}$$

$$\mathbf{Q}_x = 1.0, \qquad \mathbf{Q}_y = 0 \tag{9.4.20}$$

$$\mathbf{H} = 1, \qquad \mathbf{J} = 1, \qquad \mathbf{R} = 1 \tag{9.4.21}$$

$$\mathbf{P}_x^- = 1.618034, \qquad \mathbf{P}_{xy}^- = 0, \qquad \mathbf{P}_y^- = 1 \tag{9.4.22}$$

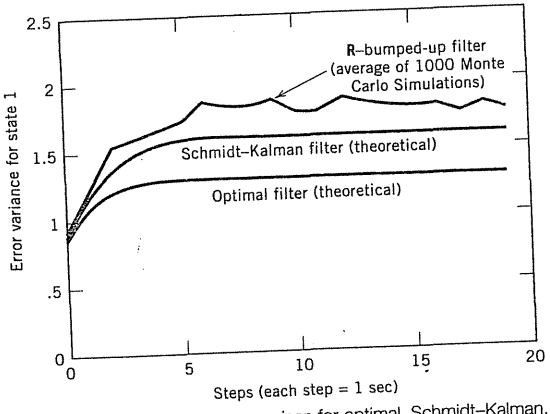


Figure 9.8 Performance comparison for optimal, Schmidt–Kalman, and **R**-bumped-up filters.

Programming Eqs. (9.4.8) through (9.4.13) is relatively easy using MATLAB, and the resulting $\mathbf{P}_x$ is plotted along with the optimal error variance in Fig. 9.8. Note that the Schmidt–Kalman error variance is somewhat larger than that for the optimal filter, but not dramatically so.

Suboptimal 1-State Filter with Bias Included in R The lazy way to account for the bias component in the measurement noise is simply to lump it in with the white component and treat the total noise as if it were white. We will call this the **R**-bumped-up filter. This, of course, is a modeling error, but such "crimes" have often been committed in applied Kalman filter work. We will now look at the seriousness of the crime (for one example, at least). The total variance of the measurement noise in this example is two units, so let us consider a usual (but suboptimal) 1-state Kalman filter with the following parameters:

$$\boldsymbol{\phi}_k = 1, \qquad \mathbf{Q}_k = 1 \tag{9.4.23}$$

$$\mathbf{H}_k = 1, \qquad \mathbf{R}_k = 2 \tag{9.4.24}$$

$$\mathbf{P}_0^- = 1.618034 \tag{9.4.25}$$

We now wish to consider the *actual* estimation error variance for such a mismodeled filter. The **P**-matrix generated by this filter will not provide a measure of the estimation error for reasons discussed in Section 6.7. Nor can we safely cycle the suboptimal gain sequence back through the truth model, because the **H**-matrix actually being implemented in this case is not the same as that in the truth model (i.e., $\mathbf{H} = [1]$ versus $\mathbf{H} = [1 \quad 1]$). We can, however, resort to

Monte Carlo simulation.* To do so, we need to generate an ensemble of sample realizations of the true x process with a random bias plus a white component added to x to form the measurement process. This process is then input into the implemented filter that yields a corresponding ensemble of suboptimal estimates. These can then be differenced with the corresponding true x's, and an ensemble of actual error trajectories is obtained. The average of the squared errors then becomes an empirically derived measure of the mean-square error of the suboptimal filter. This may seem like a crude way to perform systems analysis, but it is effective and relatively easy to do with software such as MATLAB.

The results of averaging over 1000 Monte Carlo runs for the bumped-up-**R** filter are shown in Fig. 9.8 (along with the error plots for the other two filters). Clearly, it is the poorest of the three filters that were analyzed. For this particular example, the Schmidt–Kalman filter's performance is about half-way in between the optimal and the bumped-up-**R** filter. This indicates that proper accounting for the missing state variable results in considerable improvement over the more casual method of accounting for the bias. ▩

9.5 U-D FACTORIZATION

Computational problems associated with Kalman filtering were mentioned briefly in Chapter 6. Such problems are probably not as worrisome today as they were in the early 1960s because of the spectacular progress in computer technology. Also, problems of divergence are better understood now than they were in the early days of Kalman filtering. Even so, there are occasional applications where roundoff errors can be a problem, and one must take all possible precautions against divergence. A class of Kalman filter algorithms known as square-root filtering have been developed, and they have somewhat better numerical behavior than the "usual" algorithm given in Chapter 5. (At least, with low-precision arithmetic this is so.) The basic idea is to propagate something analogous to $\sqrt{\mathbf{P}}$ (standard deviation) rather than **P** (variance). One of the critical situations is where the elements of **P** go through an extremely wide dynamical range in the course of the filter operation. For example, if the dynamical range of **P** is 20 orders of magnitude, the corresponding range of $\sqrt{\mathbf{P}}$ will be 10 orders of magnitude; and it does not take much imagination to see that we would be better off, numerically, manipulating $\sqrt{\mathbf{P}}$ rather than **P**.

Both Bierman (26) and Maybeck (27) give good accounts of the development of square-root filtering, which apparently dates back to a 1964 paper by J. E. Potter [see Battin (28)]. We will be content here to look at just the **U-D** factorization algorithm, which is due to Bierman (26). It is now the favored

* There are some special cases where recycling suboptimal gains through the truth model will yield valid error-covariance results, even though the suboptimal $\mathbf{H}_k$ (or $\boldsymbol{\phi}_k$) is not identical with that of the truth model. We will not go into the details here. It suffices to say that Monte Carlo simulation is an easy and effective means of analyzing stochastic systems when rigorous analytical means are in doubt.

algorithm in applications where numerical stability is of special concern. We will see presently that, technically, it is not really a square-root algorithm, because a square-rooting operation never appears anywhere in the algorithm. Nevertheless, it is usually considered in the square-root class because of the factorization of the $\mathbf{P}$ matrix. Our treatment of **U-D** factorization will be brief. The reader can consult either Bierman (26) or Maybeck (27) for more details. Maybeck gives an especially readable treatment of the subject.

We begin by defining that is meant by **U-D** factorization. Suppose we have a symmetric, positive definite $\mathbf{P}$ matrix. We assert that it can always be decomposed into the factored form

$$\mathbf{P} = \mathbf{U}\mathbf{D}\mathbf{U}^T \tag{9.5.1}$$

where $\mathbf{U}$ is upper triangular (ones along the major diagonal, nontrivial elements in the upper triangular part, and zeros in the lower triangular part), and $\mathbf{D}$ is diagonal. It might seem at first glance that carrying out the mechanics of the factorization would be difficult. However, this is not so. A simple (3×3) example will illustrate how easy it is to do the decomposition. We repeat Eq. (9.5.1) for a symmetric (3×3) $\mathbf{P}$ matrix.

$$\begin{aligned}\mathbf{P} &= \begin{bmatrix} p_{11} & p_{12} & p_{13} \\ p_{12} & p_{22} & p_{23} \\ p_{13} & p_{23} & p_{33} \end{bmatrix} = \begin{bmatrix} 1 & u_{12} & u_{13} \\ 0 & 1 & u_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} d_{11} & 0 & 0 \\ 0 & d_{22} & 0 \\ 0 & 0 & d_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ u_{12} & 1 & 0 \\ u_{13} & u_{23} & 1 \end{bmatrix} \\ &= \begin{bmatrix} d_{11} + d_{22}u_{12}^2 + d_{33}u_{13}^2 & d_{22}u_{12} + d_{33}u_{13}u_{23} & d_{33}u_{13} \\ d_{22}u_{12} + d_{33}u_{13}u_{23} & d_{22} + d_{33}u_{23}^2 & d_{33}u_{23} \\ d_{33}u_{13} & d_{33}u_{23} & d_{33} \end{bmatrix}\end{aligned} \tag{9.5.2}$$

The decomposition problem is to determine d_{11}, d_{22}, d_{33}, u_{12}, u_{13}, and u_{23}, without having to solve, it is hoped, simultaneous equations. Note that if we begin at the lower right corner, we can immediately write d_{33} as

$$d_{33} = p_{33} \tag{9.5.3}$$

Now, by inspection of the third column we can see that once we have determined d_{33}, we can write u_{13} and u_{23} as

$$u_{13} = \frac{p_{13}}{d_{33}} \quad \left(\text{or } \frac{p_{13}}{p_{33}}\right) \tag{9.5.4}$$

$$u_{23} = \frac{p_{23}}{d_{33}} \quad \left(\text{or } \frac{p_{23}}{p_{33}}\right) \tag{9.5.5}$$

Next, we continue by going to the 22 term and determine d_{22} from the equation

$$d_{22} = p_{22} - d_{33}u_{23}^2 \tag{9.5.6}$$

Finally, in a similar manner we go on to the 12 term and determine u_{12}, and

then to the 11 term to find d_{11}, and all of this without solving any simultaneous equations or performing any square-root operations. It suffices to say here that this procedure can be automated in the form of an algorithm for computer use, and it is relatively easy to decompose $\mathbf{P}$ into its $\mathbf{U}$ and $\mathbf{D}$ factors. We will now go through the recursive equations, step by step, in terms of the **U-D** factors.

We will begin with a priori $\mathbf{P}^-$ and will omit the k subscript just to save writing. We first factor $\mathbf{P}^-$ into its $\mathbf{U}$ and $\mathbf{D}$ parts.

$$\mathbf{P}^- = \mathbf{U}^-\mathbf{D}^-\mathbf{U}^{-T} \tag{9.5.7}$$

The Kalman gain is computed in the usual way, either as $\mathbf{P}^-\mathbf{H}^T(\mathbf{H}\mathbf{P}^-\mathbf{H}^T + \mathbf{R})^{-1}$ if $\mathbf{P}^-$ is used, or by the same formula with $\mathbf{P}^-$ replaced by $\mathbf{U}^-\mathbf{D}^-\mathbf{U}^{-T}$ if $\mathbf{P}^-$ is not available in reconstructed form from the previous step. After the gain is computed, the estimate is updated by using the usual equation:

$$\hat{\mathbf{x}} = \hat{\mathbf{x}}^- + \mathbf{K}(\mathbf{z} + \mathbf{H}\hat{\mathbf{x}}^-) \tag{9.5.8}$$

Next, we need to update $\mathbf{P}^-$, and this is where we use **U-D** factorization. We will begin with the $\mathbf{P}^-$ update form given by Eq. (5.5.20), which we repeat here for convenience.

$$\mathbf{P} = \mathbf{P}^- - \mathbf{P}^-\mathbf{H}^T(\mathbf{H}\mathbf{P}^-\mathbf{H}^T + \mathbf{R})^{-1}\mathbf{H}\mathbf{P}^- \tag{9.5.9}$$

We will assume that the measurement is scalar. (Note that we can always process the measurements sequentially in scalar form by forming appropriate linear combinations of the elements of $\mathbf{z}$, if necessary, and thus diagonalize the $\mathbf{R}$ matrix. See Problem 6.2.) The $(\mathbf{H}\mathbf{P}^-\mathbf{H}^T + \mathbf{R})$ term will then be scalar, and we will denote it as α, that is,

$$\alpha \triangleq \mathbf{H}\mathbf{P}^-\mathbf{H}^T + \mathbf{R} \tag{9.5.10}$$

Next, we will rewrite $\mathbf{P}$ and $\mathbf{P}^-$ in Eq. (9.5.9) in terms of their **U-D** factors. Let $\mathbf{P} = \mathbf{U}^+\mathbf{D}^+\mathbf{U}^{+T}$. Then

$$\begin{aligned}\mathbf{U}^+\mathbf{D}^+\mathbf{U}^{+T} &= \mathbf{U}^-\mathbf{D}^-\mathbf{U}^{-T} - \frac{1}{\alpha}\mathbf{U}^-\mathbf{D}^-\mathbf{U}^{-T}\mathbf{H}^T\mathbf{H}\mathbf{U}^-\mathbf{D}^-\mathbf{U}^{-T} \\ &= \mathbf{U}^-\left[\mathbf{D}^- - \frac{1}{\alpha}(\mathbf{D}^-\mathbf{U}^{-T}\mathbf{H}^T)(\mathbf{D}^-\mathbf{U}^{-T}\mathbf{H}^T)^T\right]\mathbf{U}^{-T}\end{aligned} \tag{9.5.11}$$

Now note that the bracketed term in Eq. (9.5.11) is symmetric, so it can be factored into **U-D** factors. Let

$$\left[\mathbf{D}^- - \frac{1}{\alpha}(\mathbf{D}^-\mathbf{U}^{-T}\mathbf{H}^T)(\mathbf{D}^-\mathbf{U}^{-T}\mathbf{H}^T)^T\right] = \bar{\mathbf{U}}\bar{\mathbf{D}}\bar{\mathbf{U}}^T \tag{9.5.12}$$

Then, Eq. (9.5.11) can be rewritten as

$$\begin{aligned}\mathbf{U}^+\mathbf{D}^+\mathbf{U}^{+T} &= \mathbf{U}^-\overline{\mathbf{U}}\,\overline{\mathbf{D}}\,\overline{\mathbf{U}}^T\mathbf{U}^{-T} \\ &= (\mathbf{U}^-\overline{\mathbf{U}})\overline{\mathbf{D}}(\mathbf{U}^-\overline{\mathbf{U}})^T\end{aligned} \tag{9.5.13}$$

Now note that $(\mathbf{U}^-\overline{\mathbf{U}})$ is upper triangular and $\overline{\mathbf{D}}$ is diagonal. Therefore, Eq. (9.5.13) is in **U-D** form and thus

$$\mathbf{U}^+ = \mathbf{U}^-\overline{\mathbf{U}} \tag{9.5.14}$$

$$\mathbf{D}^+ = \overline{\mathbf{D}} \tag{9.5.15}$$

Equations (9.5.14) and (9.5.15) then provide a means of updating the error covariance, but in terms of its **U-D** factors rather than in terms of $\mathbf{P}^-$.

It now remains to project the estimate and $\mathbf{U}^+$ and $\mathbf{D}^+$ ahead to the next step. The estimate is projected ahead as usual (with the subscripts reinserted).

$$\hat{\mathbf{x}}_{k+1}^- = \boldsymbol{\phi}_k\hat{\mathbf{x}}_k \tag{9.5.16}$$

The $\mathbf{U}^+$ and $\mathbf{D}^+$ factors can be projected ahead in a number of ways. The easiest way is to write the usual $\mathbf{P}$ projection equation as

$$\begin{aligned}\mathbf{P}_{k+1}^- &= \boldsymbol{\phi}_k\mathbf{P}_k\boldsymbol{\phi}_k^T + \mathbf{Q}_k \\ &= \boldsymbol{\phi}_k\mathbf{U}_k^+\mathbf{D}_k^+\mathbf{U}_k^{+T}\boldsymbol{\phi}_k^T + \mathbf{Q}_k \\ &= (\boldsymbol{\phi}_k\mathbf{U}_k^+)\mathbf{D}_k^+(\boldsymbol{\phi}_k\mathbf{U}_k^+)^T + \mathbf{Q}_k\end{aligned} \tag{9.5.17}$$

This computation is made easily, and it is especially efficient if $\boldsymbol{\phi}_k$ is upper triangular, or nearly so. (Integrators in cascade lead to an upper triangular transition matrix.) Also, the computation specified by Eq. (9.5.17) provides $\mathbf{P}_{k+1}^-$, which can then be used directly in the gain step at t_{k+1}. However, this method of projection begs the issue somewhat, because $\mathbf{U}^+$ and $\mathbf{D}^+$ have not been projected ahead individually. Rather, $\mathbf{P}_{k+1}^-$ has been reconstructed in the projection process. This technique has been used in equipment, though, and it is justified in applications where $\mathbf{Q}_k$ is "comfortably" positive definite. This is to say, equivalently, that there is a substantial amount of process noise being added to each of the state variables in the projection step. This, in turn, adds to the estimation uncertainty in all the state variables. In such cases, the $\mathbf{P}$-update step will be the critical one, and this is where the **U-D** factorization does the most good, so to speak.

If, for some reason, it is necessary to project $\mathbf{U}^+$ and $\mathbf{D}^+$ ahead individually and gain the corresponding numerical benefits, there are ways of doing so. Maybeck (27) gives one method that is straightforward, but it is somewhat more complicated and involves more computational effort than reconstructing $\mathbf{P}_{k+1}^-$ as given by Eq. (9.5.17). We will not pursue the matter further here other than to say that either Maybeck (27) or Bierman (26) may be consulted for methods of handling the projection step in terms of the **U-D** factors.

We have one final comment before leaving the subject of **U-D** factorization. Do not be unduly alarmed by the preceding remarks about numerical divergence.

In many applications where the system is completely observable and there is adequate process noise feeding into all the system states, there is no divergence problem. This assumes, of course, that the programmer has taken the usual precaution of symmetrizing the error covariance matrix after both the projection and update steps. In most instances, the usual update equation

$$\mathbf{P} = (\mathbf{I} - \mathbf{KH})\mathbf{P}^- \tag{9.5.18}$$

is perfectly adequate and is, of course, easy to program. Therefore, do not be afraid of numerical stability in Kalman filtering. In rare situations it can be a problem but, even in these cases, there are methods of coping with it.

9.6 DECENTRALIZED KALMAN FILTERING*

In our previous discussions of Kalman filtering, we always considered all of the measurements being input directly into a single filter. This mode of operation is now usually referred to as a *centralized* Kalman filter. Before we look at alternatives, it should be recognized that the centralized filter yields optimal estimates. We cannot expect to find any other filter that will produce any better MMSE (minimum mean-square error) estimates, subject, of course, to the usual assumptions of linear dynamics and measurement connections, and the validity of the state models that describe the various random processes. Occasionally, though, there are applications where a full-order centralized filter cannot be implemented, for one reason or another. Even so, the centralized filter is still important because it serves as a baseline for purposes of comparison. We will now look at some alternatives to the centralized filter architecture.

Cascaded Filters (With Preexisting Equipment Constraints)

A simplified block diagram for a decentralized filter is shown in Fig. 9.9. A basic tenet of the decentralized approach is for each of the local filters to operate autonomously. Each local filter has its own suite of measurements, and there is no sharing of measurements (directly, at least). Note that this is inherently a cascaded mode of operation, because the outputs of one or more of the local filters are acting as inputs to the master filter. A problem with cascaded filters that is often encountered in real-life engineering work can be illustrated as follows: Suppose a new piece of equipment is to be integrated into an existing suite of avionics equipment. The systems engineer is expected to blend together the outputs of the various equipments in some sensible near-optimal manner.

* The authors are especially indebted to Dr. Larry Levy, The Johns Hopkins University Applied Physics Laboratory, for much of the material in this section (29).

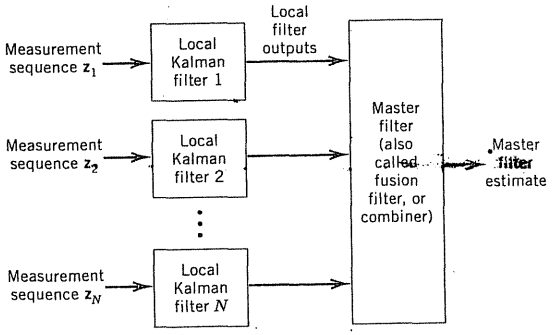


Figure 9.9 Decentralized filter—no feedback.

Each component in the suite may have its own local Kalman filter, and each of the boxes is "closed," in that the combiner (i.e., master filter) only has access to the output state estimates of the various boxes. In these circumstances it is tempting to simply treat the outputs of the local filters as measurements feeding the master filter, and then design the integration filter using the usual rules governing a Kalman filter. There are two problems with this simple approach:

1. The estimation errors in the outputs of the local filters (which translate to measurement errors in the master filter) normally have nontrivial time correlation structure. This *time* correlation is difficult to account for, even if the $\mathbf{P}$ matrix is made available to the master filter (and it usually is not in preexisting "closed" equipments). If the correlation structure of the errors in the inputs to the master filter is not properly accounted for in the master filter's model, divergence may occur.
2. If the full-order state vector is not implemented in each local filter, which is usually the case with specialized equipment, there is the possibility of loss of valuable measurement information that cannot be recovered by the master filter just by looking at local state estimates.

There is no really good theoretical solution to the constrained cascaded filter problem as just posed. One ad hoc solution that has been used successfully in special circumstances is simply to "thin out" the measurement data feeding the master filter. For example, if prefiltered estimates from a GPS receiver (see Chapter 11) are available at a 1-Hz rate, and these are to be used to update inertial equipment with large time constants, it is quite possible that sampling every 20th, or perhaps every 50th, measurement will yield satisfactory updating of the inertial equipment. In this way, the measurement errors seen by the master filter become less correlated timewise, and the white measurement noise requirement is satisfied, or at least nearly so. This solution may seem crude, but the omission of some of the measurement data may lead to only a small loss in optimality. This is certainly better than risking divergence.

Decentralized Filtering—No Feedback to the Local Filters

We will now consider various decentralized filter architectures where we relax most of the constraints on the local filters. Autonomous operation is still desired, though. We will first look at an idealistic structure where there is no feedback, and we will then use this as a point of departure for other possible architectures where there is feedback from the master filter to the local filters.

The block diagram of Fig. 9.9 applies to our no-feedback system. The autonomous operation requirement says that there is to be no sharing of measurement information among the local filters. Also, the master filter does not have direct access to the raw measurements feeding the local filters. We begin by developing a rigorous theoretical structure for our feedforward decentralized filter. The alternative form of the Kalman filter equations will be used, and the equations are repeated here for convenience (see Section 6.2 for their derivation):

1. Information matrix update:

$$\mathbf{P}_k^{-1} = (\mathbf{P}_k^-)^{-1} + \mathbf{H}_k^T\mathbf{R}_k^{-1}\mathbf{H}_k \tag{9.6.1}$$

2. Gain computation:

$$\mathbf{K}_k = \mathbf{P}_k\mathbf{H}_k^T\mathbf{R}_k^{-1} \tag{9.6.2}$$

3. Estimate update:

$$\hat{\mathbf{x}}_k = \hat{\mathbf{x}}_k^- + \mathbf{K}_k(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-) \tag{9.6.3}$$

4. Project ahead to next step:

$$\hat{\mathbf{x}}_{k+1}^- = \boldsymbol{\phi}_k\hat{\mathbf{x}}_k \tag{9.6.4}$$

$$\mathbf{P}_{k+1}^- = \boldsymbol{\phi}_k\mathbf{P}_k\boldsymbol{\phi}_k^T + \mathbf{Q}_k \tag{9.6.5}$$

Recall that $\mathbf{P}^{-1}$ is called the *information* matrix. In terms of information, Eq. (9.6.1) says that the updated information is equal to the prior information plus the additional information obtained from the measurement at time t_k. Furthermore, if $\mathbf{R}_k$ is block diagonal, the total "added" information can be divided into separate components, each representing the contribution from the respective measurement blocks. That is, we have (omitting the k subscripts for convenience)

$$\mathbf{H}^T\mathbf{R}^{-1}\mathbf{H} = \mathbf{H}_1^T\mathbf{R}_1^{-1}\mathbf{H}_1 + \mathbf{H}_2^T\mathbf{R}_2^{-1}\mathbf{H}_2 + \cdots \mathbf{H}_N^T\mathbf{R}_N^{-1}\mathbf{H}_N \tag{9.6.6}$$

We also note that the estimate update equation at time t_k can be written in a different form as follows:

$$\begin{aligned}\hat{\mathbf{x}} &= (\mathbf{I} - \mathbf{KH})\hat{\mathbf{x}}^- + \mathbf{Kz} \\ &= \mathbf{P}(\mathbf{P}^-)^{-1}\hat{\mathbf{x}}^- + \mathbf{PH}^T\mathbf{R}^{-1}\mathbf{z} \\ &= \mathbf{P}[(\mathbf{P}^-)^{-1}\hat{\mathbf{x}}^- + \mathbf{H}^T\mathbf{R}^{-1}\mathbf{z}]\end{aligned} \tag{9.6.7}$$

When written in this form, it is clear that the updated estimate is a linear blend of the old information with the new information.

For simplicity, we will start with just two local filters in our decentralized system, and we will continue to omit the k subscripts to save writing. Both filters are assumed to implement the full-order state vector, and at step k both are assumed to have available their respective prior estimates $\mathbf{m}_1$ and $\mathbf{m}_2$ and their associated error covariances $\mathbf{M}_1$ and $\mathbf{M}_2$. For Gaussian processes, $\mathbf{m}_1$ and $\mathbf{m}_2$ will be the means of $\mathbf{x}$ conditioned on their respective measurement streams up to, but not including, time t_k. The measurements presented to filters 1 and 2 at time t_k are $\mathbf{z}_1$ and $\mathbf{z}_2$, and they have the usual relationships to $\mathbf{x}$:

$$\mathbf{z}_1 = \mathbf{H}_1\mathbf{x} + \mathbf{v}_1 \tag{9.6.8}$$

$$\mathbf{z}_2 = \mathbf{H}_2\mathbf{x} + \mathbf{v}_2 \tag{9.6.9}$$

where $\mathbf{v}_1$ and $\mathbf{v}_2$ are zero mean random variables with covariances $\mathbf{R}_1$ and $\mathbf{R}_2$. The state $\mathbf{x}$ and noises $\mathbf{v}_1$ and $\mathbf{v}_2$ are assumed to be mutually uncorrelated as usual.

If we assume now that local filters 1 and 2 do not have access to each other's measurements, the filters will form their respective error covariances and estimates according to Eqs. (9.6.1) and (9.6.7).

Local filter 1:

$$\mathbf{P}_1^{-1} = \mathbf{M}_1^{-1} + \mathbf{H}_1^T\mathbf{R}_1^{-1}\mathbf{H}_1 \tag{9.6.10}$$

$$\hat{\mathbf{x}}_1 = \mathbf{P}_1(\mathbf{M}_1^{-1}\mathbf{m}_1 + \mathbf{H}_1^T\mathbf{R}_1^{-1}\mathbf{z}_1) \tag{9.6.11}$$

Local filter 2:

$$\mathbf{P}_2^{-1} = \mathbf{M}_2^{-1} + \mathbf{H}_2^T\mathbf{R}_2^{-1}\mathbf{H}_2 \tag{9.6.12}$$

$$\hat{\mathbf{x}}_2 = \mathbf{P}_2(\mathbf{M}_2^{-1}\mathbf{m}_2 + \mathbf{H}_2^T\mathbf{R}_2^{-1}\mathbf{z}_2) \tag{9.6.13}$$

Note that the local estimates will be optimal, conditioned on their *respective* measurement streams, but not with respect to the combined measurements. (Remember, the filters are operating autonomously.)

Now consider the master filter. It is looking for an optimal global estimate of $\mathbf{x}$ conditioned on both measurement streams 1 and 2. Let

$\mathbf{m}$ = optimal estimate of $\mathbf{x}$ conditioned on both measurement streams up to but not including t_k

$\mathbf{M}$ = covariance matrix associated with $\mathbf{m}$

The optimal global estimate and associated error covariance are then

$$\begin{aligned}\mathbf{P}^{-1} &= [\mathbf{H}_1^T\mathbf{H}_2^T]\begin{bmatrix}\mathbf{R}_1^{-1} & 0\\ 0 & \mathbf{R}_2^{-1}\end{bmatrix}\begin{bmatrix}\mathbf{H}_1\\ \mathbf{H}_2\end{bmatrix} + \mathbf{M}^{-1} \\ &= \mathbf{M}^{-1} + \mathbf{H}_1^T\mathbf{R}_1^{-1}\mathbf{H}_1 + \mathbf{H}_2^T\mathbf{R}_2^{-1}\mathbf{H}_2\end{aligned} \tag{9.6.14}$$

$$\hat{\mathbf{x}} = \mathbf{P}(\mathbf{M}^{-1}\mathbf{m} + \mathbf{H}_1^T\mathbf{R}_1^{-1}\mathbf{z}_1 + \mathbf{H}_2^T\mathbf{R}_2^{-1}\mathbf{z}_2) \tag{9.6.15}$$

However, the master filter does not have direct access to $\mathbf{z}_1$ and $\mathbf{z}_2$, so we will rewrite Eqs. (9.6.14) and (9.6.15) in terms of the local filter's computed estimates and covariances. The result is

$$\mathbf{P}^{-1} = (\mathbf{P}_1^{-1} - \mathbf{M}_1^{-1}) + (\mathbf{P}_2^{-1} - \mathbf{M}_2^{-1}) + \mathbf{M}^{-1} \tag{9.6.16}$$

$$\hat{\mathbf{x}} = \mathbf{P}[(\mathbf{P}_1^{-1}\hat{\mathbf{x}}_1 - \mathbf{M}_1^{-1}\mathbf{m}_1) + (\mathbf{P}_2^{-1}\hat{\mathbf{x}}_2 - \mathbf{M}_2^{-1}\mathbf{m}_2) + \mathbf{M}^{-1}\mathbf{m}] \tag{9.6.17}$$

It can now be seen that the local filters can pass their respective $\hat{\mathbf{x}}_i$, $\mathbf{P}_i^{-1}$, $\mathbf{m}_i$, $\mathbf{M}_i^{-1}$ $(i = 1, 2)$ on to the master filter, which, in turn, can then compute its global estimate. The local filters can, of course, do their own local projections and then repeat the cycle at step $k + 1$. Likewise, the master filter can project its global estimate and get a new $\mathbf{m}$ and $\mathbf{M}$ for the next step. Thus, we see that this architecture permits complete autonomy of the local filters, and it yields local optimality with respect to the respective measurement streams. The system also achieves global optimality in the master filter.

We now come to a matter of semantics. Note that the parenthetical quantities in Eq. (9.6.17) are really just $\mathbf{H}_1^T\mathbf{R}_1^{-1}\mathbf{z}_1$ and $\mathbf{H}_2^T\mathbf{R}_2^{-1}\mathbf{z}_2$. Each local filter must compute its respective $\mathbf{HR}^{-1}\mathbf{z}$ to get its local estimate (see Eqs. 9.6.11 and 9.6.13), so why not pass these quantities on to the master filter rather than the more complicated individual terms that appear in Eq. (9.6.17). Certainly, it would be simpler. However, if the local filters do pass their $\mathbf{HR}^{-1}\mathbf{z}$ quantities on to the master filter, this becomes very close to a parallel architecture where the raw measurements are fed not only to the local filters but also to a larger master filter that optimally processes all the measurements. Such parallel systems have been implemented (30). So, it might be said that our feedforward decentralized filter is really just a parallel architecture in disguise. We will not try to answer this question here—it is simply a matter of semantics.

Decentralized Filtering with Feedback

We now wish to consider a decentralized filter architecture where we permit information feedback from the master filter to the local filters. Figure 9.10 shows such an architecture for two local filters. We note that by feeding back the prior $\mathbf{m}$ and $\mathbf{M}$ to the local filters, we are allowing indirect measurement sharing between the local filters. This feedback enables the local filters to reset their respective prior estimates more accurately with each step than they would be able to do otherwise. All we need to do to get the appropriate equations for the feedback configuration is to let

$$\mathbf{m}_1 = \mathbf{m}_2 = \mathbf{m} \tag{9.6.18}$$

$$\mathbf{M}_1 = \mathbf{M}_2 \doteq \mathbf{M} \tag{9.6.19}$$

With these modifications the previously derived equations can be applied to the decentralized filter with feedback. The key equations are

Local filter 1 (with feedback):

$$\hat{\mathbf{x}}_1 = \mathbf{P}_1(\mathbf{M}^{-1}\mathbf{m} + \mathbf{H}_1^T\mathbf{R}_1^{-1}\mathbf{z}_1) \tag{9.6.20}$$

$$\mathbf{P}_1^{-1} = \mathbf{M}^{-1} + \mathbf{H}_1^T\mathbf{R}_1^{-1}\mathbf{H}_1 \tag{9.6.21}$$

Local filter 2 (with feedback):

$$\hat{\mathbf{x}}_2 = \mathbf{P}_2(\mathbf{M}^{-1}\mathbf{m} + \mathbf{H}_2^T\mathbf{R}_2^{-1}\mathbf{z}_2) \tag{9.6.22}$$

$$\mathbf{P}_2^{-1} = \mathbf{M}^{-1} + \mathbf{H}_2^T\mathbf{R}_2^{-1}\mathbf{H}_2 \tag{9.6.23}$$

Global filter:

$$\hat{\mathbf{x}} = \mathbf{P}[\mathbf{P}_1^{-1}\hat{\mathbf{x}}_1 + \mathbf{P}_2^{-1}\hat{\mathbf{x}}_2 - \mathbf{M}^{-1}\mathbf{m}] \tag{9.6.24}$$

$$\mathbf{P}^{-1} = \mathbf{P}_1^{-1} + \mathbf{P}_2^{-1} - \mathbf{M}^{-1} \tag{9.6.25}$$

Note that there is no direct communication between filters 1 and 2 with this

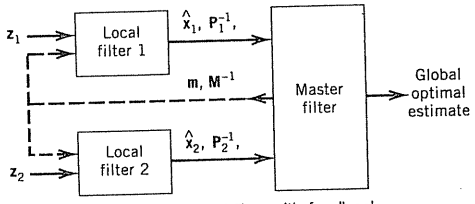


Figure 9.10 Decentralized filter with feedback.

architecture. There is indirect information sharing, though, in that the prior $\mathbf{m}$ that is fed back with each step is a linear combination of past measurements in both filters.

The master filter maintains full global optimality with the feedback architecture where $\mathbf{m}_1 = \mathbf{m}_2 = \mathbf{m}$. The local filters have better optimality than they would have without feedback, but they still do not have full optimality with respect to all measurements. For example, with the Gaussian assumption local estimate $\hat{\mathbf{x}}_1$ is the conditional mean of $\mathbf{x}$, conditioned on all past measurements and the present $\mathbf{z}_1$ (but not $\mathbf{z}_2$). A similar interpretation applies to $\hat{\mathbf{x}}_2$.

Concluding Remarks

Two rigorous architectures for filter decentralization have been presented. One involves feeding back information from the master filter to the local filters; the other does not. In both architectures global optimality is maintained in the master filter. Also, optimality is achieved in the local filters to a limited degree with respect to specific subsets of the measurements. Thus, these configurations are important as baseline architectures for purposes of comparison.

Both the no-feedback and feedback architectures considered here require that the full-order state vector (i.e., the global state vector) be implemented in each of the local filters. In many applications this is an unreasonable requirement. One of the motivations for decentralization is simplification, so having to implement the full state vector in all the filters does not help in this regard. Thus, in practice there are applications where it is necessary to consider a suite of local filters with lower-order state vectors. This complicates the theory considerably, and it nearly always leads to some degree of suboptimality relative to the two baseline architectures considered here.

One architecture that has received considerable attention as a practical means of decentralization is the *federated filter* (38, 39, 40). In the federated filter the information that is fed back to the local filters is divided, and portions of the total information are shared among the local filters. Various sharing strategies are possible (see Problem 9.7 for an example of information sharing). Dividing the total information leads to some degree of suboptimality in the local filters, but there are reports of applications where the penalty is quite modest (39). The federated filter is relatively new, and it is considerably more complicated conceptually than the straightforward parallel configuration where the $\mathbf{z}_1$, $\mathbf{z}_2$, . . . $\mathbf{z}_N$ measurements are input directly to a master filter, as well as to the respective local filters. The federated filter must compete with parallel systems (as well as other architectures), so the eventual role of the federated filter in integrated systems remains to be seen.

9.7 STOCHASTIC LINEAR REGULATOR PROBLEM AND THE SEPARATION THEOREM

The linear regulator problem is a classical problem of optimal control theory, and it can be posed (and solved) in either continuous or discrete form (24,

34–38). In the interest of brevity we will confine our remarks to the continuous version of the problem.

The *deterministic* linear regulator problem may be stated as follows: Given a linear dynamical system of the form

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{B}\mathbf{u}_d \tag{9.7.1}$$

what $\mathbf{u}_d(t)$ will minimize the quadratic performance index

$$J = \mathbf{x}^T(t_f)\mathbf{S}\mathbf{x}(t_f) + \int_{t_0}^{t_f} [\mathbf{x}^T(t)\mathbf{W}_x\mathbf{x}(t) + \mathbf{u}_d^T(t)\mathbf{W}_u\mathbf{u}_d(t)]\, dt \tag{9.7.2}$$

where $\mathbf{S}$, $\mathbf{W}_x$ and $\mathbf{W}_u$ are symmetric positive definite weighting matrices that are chosen to fit the situation at hand? The intent of the regulator is to reduce the initial state of the system to zero (or nearly so) quickly and without undue control effort. The weighting factor $\mathbf{S}$ penalizes the system for not reaching zero at the specified terminal time t_f. The weighting factors $\mathbf{W}_x$ and $\mathbf{W}_u$ apply penalties for the trajectories of the state $\mathbf{x}(t)$ and the control $\mathbf{u}_d(t)$ over the time span $[t_0, t_f]$. It can be seen that if $\mathbf{W}_x$ is large and $\mathbf{W}_u$ small, the optimal system will apply a large control effort and force the system toward zero rapidly in order to avoid a large penalty due to large $\mathbf{W}_x$. On the other hand, if $\mathbf{W}_x$ is small and $\mathbf{W}_u$ large, the system will be frugal with its control effort, and it will approach zero slowly. Clearly, a wide variety of situations can be accommodated within the structure of this formulation. We will not derive the solution of the linear regulator problem here. This is adequately covered in the mentioned references (and many others, as well). We simply state that the optimal control law is a linear feedback law that specifies $\mathbf{u}_d(t)$ to be

$$\mathbf{u}_d(t) = -\mathbf{K}_1(t)\mathbf{x}(t) \tag{9.7.3}$$

where the feedback gain $\mathbf{K}_1(t)$ is computed from the system parameters. (It is not a function of $\mathbf{x}$.) We need not be concerned with the detailed solution for $\mathbf{K}_1(t)$ (except to note parenthetically that there is a close duality between this and the optimal estimator problem.) It is important to note, though, that it is assumed that the state vector $\mathbf{x}$ is available for feedback purposes as indicated in Fig. 9.11. However, in many physical situations, we do not have the privilege of observing all the elements of $\mathbf{x}$ directly; quite to the contrary, we are usually allowed to observe $\mathbf{x}$ only through some output relationship, say,

$$\mathbf{y} = \mathbf{H}\mathbf{x} \tag{9.7.4}$$

Now, based on $\mathbf{y}$, we must somehow reconstruct $\mathbf{x}$. If the observation is essentially error-free, we can use observer theory in the reconstruction (38–40). However, if the observation of $\mathbf{x}$ is corrupted with noise, as is often the case, then

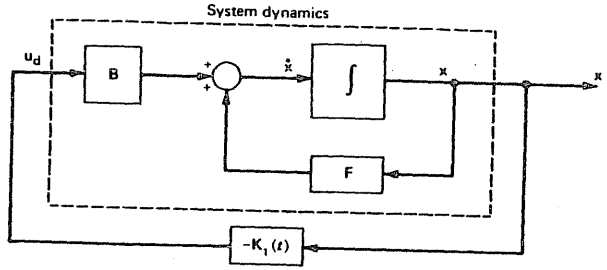


Figure 9.11 Optimal linear regulator.

the reconstruction of $\mathbf{x}$ becomes an estimation problem. This leads to the stochastic linear regulator problem,* which will now be formulated.

Let the system dynamics be specified by the linear equation

$$\dot{\mathbf{x}} = \mathbf{F}\mathbf{x} + \mathbf{B}\mathbf{u}_d + \mathbf{G}\mathbf{u} \tag{9.7.5}$$

where $\mathbf{F}$, $\mathbf{B}$, and $\mathbf{u}_d$ are as before, and $\mathbf{u}$ is an additional Gaussian white-noise forcing function that is characterized by

$$E[\mathbf{u}(t)\mathbf{u}^T(\tau)] = \mathbf{Q}\delta(t - \tau) \tag{9.7.6}$$

The system state vector is assumed to be observed via the relationship

$$\mathbf{z} = \mathbf{H}\mathbf{x} + \mathbf{v} \tag{9.7.7}$$

where $\mathbf{v}$ is Gaussian white noise and is characterized by

$$E[\mathbf{v}(t)\mathbf{v}^T(\tau)] = \mathbf{R}\delta(t - \tau) \tag{9.7.8}$$

The two white noises $\mathbf{u}$ and $\mathbf{v}$ will be assumed to be independent, and, in order to avoid any questions about singular conditions, we assume that the system is controllable with respect to the control input $\mathbf{u}_d$ and is observable with respect to $\mathbf{z}$. The optimization problem is to minimize the following cost function:

$$J = E\left\{\mathbf{x}^T(t_f)\mathbf{S}\mathbf{x}(t_f) + \int_{t_0}^{t_f} [\mathbf{x}^T(t)\mathbf{W}_x\mathbf{x}(t) + \mathbf{u}_d^T(t)\mathbf{W}_u\mathbf{u}_d(t)]\, dt\right\} \tag{9.7.9}$$

As before, we will not dwell on the details of the solution; we will simply state the results (24, 36, 37, 38). Design of the optimal stochastic linear regulator may be separated into two steps:

* This problem is also called the linear quadratic Gaussian (LQG) problem.

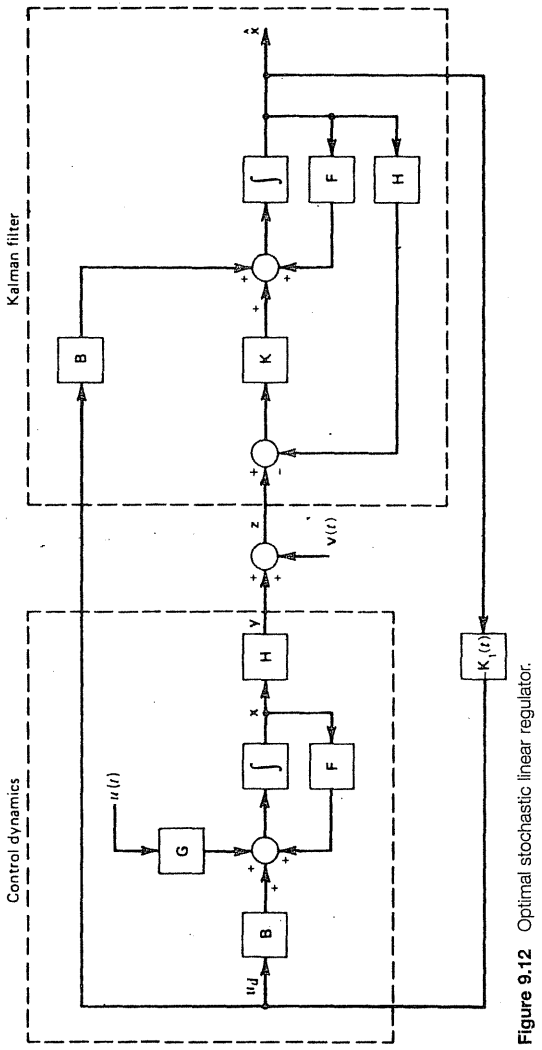


Figure 9.12 Optimal stochastic linear regulator.

1. First, design the minimum mean-square error estimator for $\mathbf{x}$, treating $\mathbf{u}_d$ just as if it were a known deterministic input. The optimal estimator is, of course, a Kalman filter with parameters $\mathbf{F}$, $\mathbf{GQG}^T$, $\mathbf{H}$, and $\mathbf{R}$. Note that the process has a known input as well as a random input; therefore, the differential equation for $\hat{\mathbf{x}}$ is

$$\dot{\hat{\mathbf{x}}} = \mathbf{F}\hat{\mathbf{x}} + \mathbf{B}\mathbf{u}_d + \mathbf{K}(\mathbf{z} - \mathbf{H}\hat{\mathbf{x}}) \tag{9.7.10}$$

   where $\mathbf{K}$ is the continuous Kalman gain (see Problem 7.8).
2. Next, solve the deterministic linear regulator problem for the optimal feedback gain $\mathbf{K}_1(t)$ just as if a perfect measurement of $\mathbf{x}$ were available and $\mathbf{u}(t)$ were absent. Then let control input $\mathbf{u}_d$ be

$$\mathbf{u}_d = -\mathbf{K}_1(t)\hat{\mathbf{x}}(t) \tag{9.7.11}$$

   and the cost function is minimized. The resulting optimal system is summarized in Fig. 9.12.

The two-step solution just described is known as the *separation theorem* or *separation principle*. It is a most remarkable result in that we would normally expect that the feedback loop would horribly complicate matters by mixing together the control and estimation problems. However, it does not; the two problems separate nicely! The superficial reason for this is the duality between the optimal control and optimal filter problems. The underlying reason for the duality in the first place, though, is not all that obvious, so we will simply say that this is a fortuitous circumstance.

It is not our intent here to teach the subject of optimal control. The subject is well developed, and many fine books have been written in this area. We simply want to point out with this limited example that estimation theory (and Kalman filtering in particular) plays an important role in optimal control theory, and it behooves the control systems engineer/scientist to understand the rudiments of estimation theory.

PROBLEMS

9.1 Consider a simple one-dimensional trajectory determination problem as follows. A small object is launched vertically in the earth's atmosphere. The initial thrust exists for a very short time, and the object "free falls" ballistically for essentially all of its straight-up, straight-down trajectory. Let y be measured in the up direction, and assume that the nominal trajectory is governed by the following dynamical equation:

$$m\ddot{y} = -mg - D\dot{y}|\dot{y}|$$

where

$$m = .05 \text{ kg} \quad \text{(mass of object)}$$
$$g = 9.087 \text{ m/sec}^2 \quad \text{(acceleration of gravity)}$$
$$D = 1.4 \times 10^{-4} \text{ n/(m/sec)}^2 \quad \text{(drag coefficient)}$$

The drag coefficient will be assumed to be constant for this relatively short

trajectory, and note that drag force is proportional to (velocity)^2, which makes the differential equation nonlinear. Let the initial conditions for the nominal trajectory be as follows:

$$y(0) = 0$$

$$\dot{y}(0) = 85 \text{ m/sec}$$

The body is tracked and noisy position measurements are obtained at intervals of .1 sec. The measurement error variance is .25 m^2. The actual trajectory will differ from the nominal one primarily because of uncertainty in the initial velocity. Assume that the initial position is known perfectly but that initial velocity is best modeled as a normal random variable described by N (85 m/sec, 1 m^2/sec^2). Work out the linearized discrete Kalman filter model for the up portion of the trajectory.
(*Hint:* An analytical solution for the nominal trajectory may be obtained by considering the differential equation as a first-order equation in velocity. Note $|\dot{y}| = \dot{y}$ during the up portion of the trajectory. Since variables are separable in the velocity equation, it can be integrated. The velocity can then be integrated to obtain position.)

9.2. (a) At the kth step of the usual nonadaptive Kalman filter, the measurement residual is $(\mathbf{z}_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-)$. Let z_k be scalar and show that the expectation of the squared residual is minimized if $\hat{\mathbf{x}}_k^-$ is the optimal a priori estimate of $\mathbf{x}_k$, that is, the one normally computed in the projection step of the Kalman filter loop.
(*Hint:* Use the measurement relationship $z_k = \mathbf{H}_k\mathbf{x}_k + v_k$ and note that v_k and the a priori estimation error have zero crosscorrelation. Also note that the a priori estimate $\hat{\mathbf{x}}_k^-$, optimal or otherwise, can only depend on the measurement sequence up through z_{k-1} and not z_k.)

(b) Show that the time sequence of residuals $(z_k - \mathbf{H}_k\hat{\mathbf{x}}_k^-)$, $(z_{k+1} - \mathbf{H}_{k+1}\hat{\mathbf{x}}_{k+1}^-)$, . . . is a white sequence if the filter is optimal. As a matter of terminology, this sequence is known as an *innovations* sequence. See (33) for a brief discussion of innovations processes and their role in optimal filter theory.

9.3 In Problem 9.2 it was stated that the sequence of measurement residuals in a Kalman filter is a white sequence. This occurs only when the filter gain is adjusted to the optimal value. Thus, we can think of the filter gain as a "tuning" parameter that we tune to make the measurement-residual sequence white. This immediately suggests an intuitive sort of adaptive filter in which the filter monitors the measurement-residual sequence, and if it detects nonwhiteness, it readjusts the gain to force the sequence to be white. This will be demonstrated with a simple continuous Kalman filter example.

Let the process and measurement models be

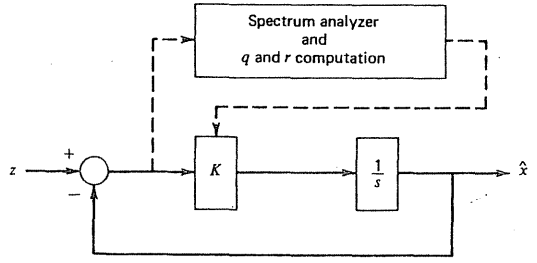
$$\dot{x} = u(t), \qquad u(t) \text{ is white with spectral amplitude } q$$

$$z = x + v(t), \qquad v(t) \text{ is white with spectral amplitude } r$$

As usual, we will assume $u(t)$ and $v(t)$ to be independent. Their spectral amplitudes q and r are presumed to be unknown and, furthermore, they might vary slowly with time. The continuous Kalman filter solution for this setting is given by (see Chapter 7)

$$\dot{\hat{x}} = K(z - \hat{x})$$

The solution is also shown in block diagram form in the accompanying figure. It is not difficult to imagine the filter performing spectral analysis on the measurement residual at the summing point in much the same manner as with any modern digital spectrum analyzer. This could be done repeatedly throughout the course of the filter operation. The result would be an experimentally derived power spectral density function (or autocorrelation function) that would be representative of the measurement-residual process over the time span of the sample.



Problem 9.3

Assume that an autocorrelation function has been determined as just described, and assume that q and r do not change appreciably during the observation span. Find q, r and the filter gain in terms of the parameters that describe the autocorrelation function. Note that once q and r are determined, the filter gain can be readjusted to the optimal condition. [If you have difficulty with this problem, see Gelb (8), pp. 319–320.]

9.4 This problem is a variation on the DME example given in Section 9.1 (Example 9.1) with a simplification in the model of the aircraft dynamics. Suppose that the two DME stations are located on the x-axis as shown in the accompanying figure, and further suppose that the aircraft follows an approximate path from south to north as shown. The aircraft has a nominal velocity of 100 m/sec in a northerly direction, but there is random motion superimposed on this in both the x and y directions. The flight duration (for our purposes) is 200 sec, and the initial coordinates at $t = 0$ are properly described as normal random variables as follows:

$$x_0 \sim N(0, 2000 \text{ m}^2)$$

$$y_0 \sim N(-10{,}000 \text{ m}, 2000 \text{ m}^2)$$

The aircraft kinematics are described by the following equations:

$$x_{k+1} = x_k + w_{1k}, \qquad k = 0, 1, 2, \ldots, 200$$

$$y_{k+1} = y_k + w_{2k} + 100\,\Delta t, \qquad k = 0, 1, 2, \ldots, 200$$

where w_{1k} and w_{2k} are independent white Gaussian sequences described by

$$w_{1k} \sim N(0, 400\ \text{m}^2)$$

$$w_{2k} \sim N(0, 400\ \text{m}^2)$$

The sampling interval Δt is 1 sec. The aircraft motion will be recognized as simple random walk in the x-coordinate, and random walk superimposed on linear motion for the y-coordinate.

The aircraft obtains simultaneous discrete range measurements at 1 sec intervals on both DME stations, and the measurement errors consist of a superposition of Markov and white components. Thus, we have (for a typical range measurement)

$$\begin{bmatrix}\text{Total}\\ \text{noisy}\\ \text{measurement}\end{bmatrix} = \begin{bmatrix}\text{true}\\ \text{total}\\ \text{range}\end{bmatrix} + \begin{bmatrix}\text{Markov}\\ \text{error}\\ \text{component}\end{bmatrix} + \begin{bmatrix}\text{white}\\ \text{error}\\ \text{component}\end{bmatrix}$$

We have two DME stations, so the measurement vector is a 2-tuple at each sample point beginning at $k = 0$ and continuing until $k = 200$. The Markov errors for each of the DME stations are independent Gaussian random processes, which are described by the autocorrelation function

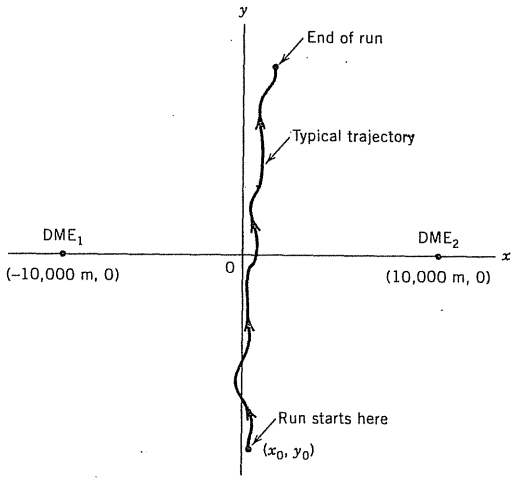
$$R_m(\tau) = 900e^{-.01|\tau|}\ \text{m}^2$$

The white measurement errors for both stations have a variance of 225 m^2.

(a) Work out the linearized Kalman filter parameters for this situation. The linearization is to be done about the nominal linear-motion trajectory exactly along the y-axis. That is, the resultant filter is to be an "ordinary" linearized Kalman filter and not an extended Kalman filter.

(b) After the key filter parameters have been worked out, run a covariance analysis for $k = 0, 1, 2, \ldots, 200$, and plot the estimation error variances for both the x and y position coordinates.

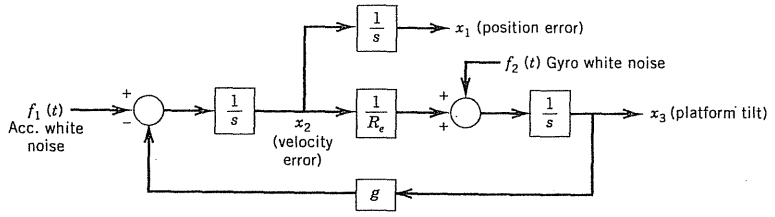
(c) You should see a pronounced peak in the y-coordinate error curve as the aircraft goes through (or near) the origin. This simply reflects a bad geometric situation when the two DME stations are 180 degrees apart relative to the aircraft. One might think that the estimation error variance should go to infinity at exactly $k = 100$. Explain qualitatively why this is not true.



Problem 9.4

9.5 It is well known that the Schuler oscillation in a pure inertial navigation system (INS) can be damped with velocity information from a noninertial measurement. This is discussed in detail in Section 10.3. There, we think of the external velocity measurements as being a sequence of instantaneous snapshot velocity measurements, so the measurement equation for the Kalman filter does not involve a delayed-state term. In this exercise, however, we wish to consider a measurement situation where the Kalman filter has to work with a sequence of measurements where they represent the *integral* of velocity over a sequence of contiguous ΔT intervals.

The single-channel INS error model given in Fig. 10.6 is repeated below for convenience, but the external velocity feedback branch has been omitted for our purposes here. Also, gyro white noise has been inserted into the Schuler loop at the appropriate point.



Problem 9.5

$R_e \approx 2.09 \times 10^7$ ft; $g \approx 32.2$ ft/sec^2, $\omega_0 = \sqrt{g/R_e} \approx 1.24 \times 10^{-3}$ rad/sec

Clearly, the true measurement situation for the problem at hand is

$$\text{Measurement} = \int_{\Delta t} x_2\, dt + \text{measurement error}$$

or, at time t_k,

$$z(t_k) = x_1(t_k) - x_1(t_{k-1}) + v(t_k) \qquad \text{(P9.5.1)}$$

Also if the measurement noise at the velocity level is white, the corresponding discrete sequence $v(t_k)$ will be white. Equation (P9.5.1) then fits the format for the delayed-state measurement model that was discussed in Section 9.2.

(a) First, refer to Fig. 10.6 and study the stability properties of the continuous system as the feedback constant K is varied. Do this by examining the roots of the characteristic polynomial $|s\mathbf{I} - \mathbf{F}|$.
[*Answer:* For $K = 0$ the three roots are at $s = 0$ and $s = \pm j\omega_0$. Then, as K is increased, the complex roots move into the left-half s-plane and converge on the negative real axis (critical damping). As K is increased further, one root goes to the left and the other to the right, which corresponds to the overdamped situation.]

(b) We now wish to demonstrate that damping can be achieved just as well with integrated velocity measurements as with instantaneous samples of velocity. Toward this end we will look at the error covariance propagation in the delayed-state filter as discussed in Section 9.2. This can be done easily using MATLAB (or other suitable software). The specific parameters to be used for this exercise are as follows (in addition to the constants shown in the accompanying figure):
(i) Integration interval: $\Delta t = 120$ sec
(ii) Perfect alignment at $t = 0$ (i.e., $k = 0$)
(iii) Integrated velocity measurement-error variance:

$$\mathbf{R}_k = (300 \text{ ft})^2$$

(iv) Power spectral density of $f_1(t) = .003$ $(\text{ft/sec}^2)^2/\text{rad/sec}$
(v) Power spectral density of $f_2(t) = 2.35e{-}11$ $(\text{rad/sec})^2/\text{rad/sec}$
(This will induce an rms position error of about 1 n.mi in 1 hour, in the absence of velocity damping.)

The dynamic scenario to be programmed is as follows:

1. Initialize the $\mathbf{P}$ matrix to be the null matrix (i.e., perfect alignment).
2. Sample points are at $k = 0, 1, 2, \ldots, 100$, corresponding to $t = 0, 120, 240, \ldots, 12{,}000$ sec. (Note, however, that the first conceptual measurement occurs at $k = 1$, not $k = 0$, because of the delayed-state measurement situation.)
3. For the first 42 steps ($k = 1, 2, \ldots, 42$), the system is to run free of velocity measurements. This allows the error covariances for all three states to build up to nontrivial values, and it demonstrates the system instability without damping. (This is easily simulated during this period by letting $\mathbf{R}_k$ be excessively large, e.g., 1.0e20.)
4. The conceptual velocity measurements begin at $k = 43$ and continue until $k = 100$. $\mathbf{R}_k$ is to be set at 90,000 during this period.

Plot the error covariances associated with each of the state variables (i.e., p_{11}, p_{22}, and p_{33}) as a function of k (or time). Note that when the external velocity measurement is "turned on," the damping effect on the INS velocity error is very rapid. The damping effect is less rapid, though, on the position and platform errors. This is to be expected—they are one integration removed from the velocity measurement. Note especially the stabilizing effect on the platform tilt.

(c) In Section 6.9, it was shown that the stability characteristics of a constant-gain Kalman filter are described by the filter's characteristic roots much the same as with the digital filters that one studies in digital signal theory (41). In the scenario considered here in part (b), the filter reaches a near-constant gain condition at the end of the run. First, adapt the derivation given in Section 6.9 to the delayed-state filter equations and show that the characteristic roots of the delayed-state filter are the eigenvalues of $(\boldsymbol{\phi} - \mathbf{K}\mathbf{H}\boldsymbol{\phi} - \mathbf{K}\mathbf{J})$, where $\boldsymbol{\phi}$, $\mathbf{K}$, $\mathbf{H}$, and $\mathbf{J}$ are steady-state values. Then, using the value of $\mathbf{K}$ obtained at $k = 100$, find the characteristic roots of the filter. [Note that the built-in MATLAB function eig(A) returns the eigenvalues of a square matrix A.] It should be noted that in the case of the discrete filter we are working with z-transforms. The unit circle in the z-domain plays the same role as the imaginary axis in the s-domain, and the entire left-half s-plane maps into the interior of the unit circle in the z-domain.
(*Answer:* For $\mathbf{R}_k = 90{,}000$ the characteristic roots are $z = 1.0$, $z \approx .7915 \pm j.1593$.)

(d) Once the computer program for parts (b) and (c) is written, it is easy to rerun the scenario using different values of $\mathbf{R}_k$. Try two more runs, one with $\mathbf{R}_k$ set larger than 90,000, and another with $\mathbf{R}_k$ less than 90,000. (Suggested values for this part are 160,000 and 40,000.) Are your results reasonable in view of how the characteristic roots vary with a change of gain in the corresponding continuous problem?

9.6 In Example 9.4 the measurement bias that was to be "considered" (but not implemented) was modeled as a true constant with time. It may be more realistic in some applications to model the "bias" as a quasibias, that is, one that is allowed to vary slowly with time in a random manner. Let us say that a Gauss–Markov process is a suitable way to model the quasibias, and its autocorrelation function is given by

$$R_y(\tau) = \sigma^2 e^{-\beta|\tau|}; \qquad \sigma^2 = 1 \quad \text{and} \quad \beta = .1 \text{ sec}^{-1}$$

Repeat the three-way comparison among the optimal, Schmidt–Kalman, and $\mathbf{R}$-bumped-up filters using the same parameters as in Example 9.4, except for the measurement bias model. In order to be assured of reaching a near steady-state condition after $k = 0$, you may wish to let the filters run through 50 recursive steps instead of 20 as was done in Example 9.4. By our choice of $\sigma^2 = 1$, the initial conditions here are the same as in Example 9.4; note, in particular, that the initial true $\mathbf{x}$ for each Monte Carlo run (for the $\mathbf{R}$-bumped-up filter) can be set to be a sample of an $N(0, \mathbf{P}_0^-)$ random variable, and $\hat{\mathbf{x}}_0^-$ can be set to zero. This corresponds physically to resetting the random walk process to zero, in accordance with the optimal filter's best estimate, for each step prior to $k = 0$. Then at the step just before $k = 0$, the projection of $\hat{\mathbf{x}}$ to $k = 0$ is zero, and its uncertainty (as measured by its variance) is just (.618034 + 1.0).

When the resulting three error variance plots are compared, you will note that all three variances are closer together than they were in Example 9.4. Give a heuristic explanation of this.

9.7 In some decentralized filters with feedback, information from the master filter is divided and shared among the local filters (see Section 9.6). We will now look at a simple example of information sharing.

(a) First, show that the two-filter equations for the feedback case (Eqs. 9.6.20 through 9.6.25) can be generalized to account for N local filters. The result for the global filter is

$$\hat{\mathbf{x}} = \mathbf{P}\left[\sum_{i=1}^{N} \mathbf{P}_i^{-1}\hat{\mathbf{x}}_i - (N-1)\mathbf{M}^{-1}\mathbf{m}\right] \tag{P9.7.1}$$

$$\mathbf{P}^{-1} = \left(\sum_{i=1}^{N} \mathbf{P}_i^{-1}\right) - (N-1)\mathbf{M}^{-1} \tag{P9.7.2}$$

(b) Now suppose that the master filter feeds the following prior quantities back to the local filters:

$\mathbf{m}_i = \mathbf{m}$ (same optimal $\mathbf{m}$ is fed back to each local filter)

$\mathbf{M}_i^{-1} = \left(\frac{1}{N}\right)\mathbf{M}^{-1}$ (optimal prior information $\mathbf{M}^{-1}$ is divided by N, and the same amount is fed back to each local filter)

Show that the global filter equations simplify to the following equations:

$$\hat{\mathbf{x}} = \mathbf{P}\left[\sum_{i=1}^{N} \mathbf{P}_i^{-1}\hat{\mathbf{x}}_i\right] \tag{P9.7.3}$$

$$\mathbf{P}^{-1} = \sum_{i=1}^{N} \mathbf{P}_i^{-1} \tag{P9.7.4}$$

(c) With the information divided and fed back as in part (b), is the master (i.e., global) filter estimate a truly optimal global estimate? Also, are the local estimates optimal with respect to their respective measurement streams? Justify your answers.

9.8 The discrete filter equations that account for the correlation between $\mathbf{w}_{k-1}$ and $\mathbf{v}_k$ should, for small step size Δt, go over into the continuous filter equations derived in Section 7.3 by other means. In order to show this, we must first derive the appropriate connection between $\mathbf{C}$ (continuous model) and $\mathbf{C}_k$ (discrete model). We do this by noting that

$$\mathbf{w}_{k-1} = \int_{t_{k-1}}^{t_k} \boldsymbol{\phi}(t_k, \xi)\mathbf{G}(\xi)\mathbf{u}(\xi)\, d\xi$$

$$\mathbf{v}_k = \frac{1}{\Delta t}\int_{t_{k-1}}^{t_k} \mathbf{v}(\eta)\, d\eta \quad \text{(small } \Delta t\text{)}$$

The $\mathbf{C}_k$ parameter is then (for small Δt)

$$\begin{aligned}\mathbf{C}_k &= E[\mathbf{w}_{k-1}\mathbf{v}_k^T] \\ &= \frac{1}{\Delta t}\int_{t_{k-1}}^{t_k}\int_{t_{k-1}}^{t_k} \boldsymbol{\phi}(t_k, \xi)\mathbf{G}(\xi)\underbrace{E[\mathbf{u}(\xi)\mathbf{v}^T(\eta)]}_{\mathbf{C}\delta(\xi-\eta)}\, d\xi\, d\eta \\ &= \frac{1}{\Delta t}\mathbf{G}\mathbf{C}\,\Delta t\end{aligned}$$

Therefore, the desired relationship is

$$\mathbf{C}_k = \mathbf{G}\mathbf{C}$$

This is then added to the other necessary correspondences for small Δt:

$$\mathbf{H}_k = \mathbf{H}$$

$$\mathbf{R}_k = \frac{\mathbf{R}}{\Delta t}$$

$$\boldsymbol{\phi}_k = \mathbf{I} + \mathbf{F}\Delta t$$

$$\mathbf{Q}_k = \mathbf{G}\mathbf{Q}\mathbf{G}^T\,\Delta t$$

Now, following the same method used in Section 7.1, show that the discrete filter equations for the correlated case go over into the continuous filter differential equations derived in Section 7.3.

REFERENCES CITED IN CHAPTER 9

1. R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," *Trans. ASME, J. Basic Engr.:* 35–45 (March 1960).
2. R. E. Kalman and R. S. Bucy, "New Results in Linear Filtering and Prediction," *Trans. ASME, J. Basic Engr., 83:* 95–108 (1961).
3. T. Kailath, "A View of Three Decades of Linear Filtering Theory," *IEEE Trans. Information Theory, IT-20* (2): 146–181 (March 1974).
4. M. Kayton and W. R. Fried (eds.), *Avionics Navigation Systems,* New York: Wiley, 1969.

5. H. W. Sorenson, "Kalman Filtering Techniques," in C. T. Leondes (ed.), *Advances in Control Systems,* Vol. 3, New York: Academic Press, 1966, pp. 219–289.
6. R. G. Brown and J. W. Nilsson, *Introduction to Linear Systems Analysis,* New York: Wiley, 1962.
7. M. S. Grewal and A. P. Andrews, *Kalman Filtering Theory and Practice,* Englewood Cliffs, NJ: Prentice-Hall, 1993 (see Section 4.2 and Chapter 6).
8. A. Gelb (ed.), *Applied Optimal Estimation,* Cambridge, MA: MIT Press, 1974.
9. C. K. Chui, G. Chen, and H. C. Chui, "Modified Extended Kalman Filtering and a Real-Time Parallel Algorithm for System Parameter Identification," *IEEE Trans. Automatic Control,* 35(1): 100–104 (Jan. 1990).
10. S. T. Park and J. G. Lee, "Comments on 'Modified Extended Kalman Filtering and a Real-Time Parallel Algorithm for System Parameter Identification,'" *IEEE Trans. Automatic Control,* 40(9): 1661–1662 (Sept. 1995).
11. R. Ellis and D. Gulick, *Calculus with Analytical Geometry,* 4th ed., San Diego: Harcourt Brace Jovanovich, 1990.
12. R. G. Brown, "Analysis of an Integrated Inertial/Doppler-Satellite Navigation System: Part I, Theory and Mathematical Model," Tech. rept. no. ERI 62600, Engineering Research Institute, Iowa State University, Ames, 1969.
13. R. G. Brown and L. L. Hagerman, "An Optimum Inertial Doppler-Satellite Navigation System," *Navigation, J. Inst. Navigation, 16*(3): 260–269 (Fall 1969).
14. R. G. Brown and G. L. Hartman, "Kalman Filter with Delayed States as Observables," *Proceedings of the National Electronics Conference,* Chicago, IL, 1968.
15. R. G. Brown and P. Y. C. Hwang, *Introduction to Random Signals and Applied Kalman Filtering,* 2nd ed., New York: Wiley, 1992.
16. D. T. Magill, "Optimal Adaptive Estimation of Sampled Stochastic Processes," *IEEE Trans. Automatic Control, AC-10*(4): 434–439 (Oct. 1965).
17. G. L. Mealy and W. Tang, "Application of Multiple Model Estimation to a Recursive Terrain Height Correlation System," *IEEE Trans. Automatic Control, AC-28:* 323–331 (March 1983).
18. A. A. Girgis and R. G. Brown, "Adaptive Kalman Filtering in Computer Relaying: Fault Classification Using Voltage Models," *IEEE Trans. Power Apparatus Systs., PAS-104*(5): 1168–1177 (May 1985).
19. R. G. Brown, "A New Look at the Magill Adaptive Filter as a Practical Means of Multiple Hypothesis Testing," *IEEE Trans. Circuits Systs., CAS-30:* 765–768 (Oct. 1983).
20. R. G. Brown and P. Y. C. Hwang, "A Kalman Filter Approach to Precision Geodesy," *Navigation, J. Inst. Navigation, 30*(4): 338–349 (Winter 1983–84).
21. H. E. Rauch, "Autonomous Control Reconfiguration," *IEEE Control Systems,* 15(6): 37–48 (Dec. 1995).
22. W. D. Blair and Y. Bar-Shalom, "Tracking Maneuvering Targets with Multiple Sensors: Does More Data Always Mean Better Estimates?", *IEEE Trans. Aerospace Electronic Systs.,* 32(1): 450–456 (Jan. 1996).
23. M. J. Caputi, "A Necessary Condition for Effective Performance of the Multiple Model Adaptive Estimator," *IEEE Trans. Aerospace Electronic Syst.,* 31(3): 1132–1138 (July 1995).
24. J. S. Meditch, *Stochastic Optimal Linear Estimation and Control,* New York: McGraw-Hill, 1969.
25. S. F. Schmidt, "Application of State-Space Methods to Navigation Problems," in C. T. Leondes (ed.), *Advances in Control Systems,* Vol. 3, New York: Academic Press, 1966.
26. G. J. Bierman, *Factorization Methods for Discrete Sequential Estimation,* New York: Academic Press, 1977.

27. P. S. Maybeck, *Stochastic Models, Estimation and Control,* Vol. 1, New York: Academic Press, 1979.
28. R. H. Battin, *Astronautical Guidance,* New York: McGraw-Hill, 1964, pp. 338–340.
29. L. Levy, *Applied Kalman Filtering,* Unpublished Notes for Course 347, Navtech Seminars, Inc., Arlington, VA, 1994.
30. R. G. Hartman, "An Integrated GPS/IRS Design Approach," *Navigation, J. Inst. Navigation,* 35(1): 121–134 (Spring 1988).
31. N. A. Carlson, "Federated Square Root Filter for Decentralized Parallel Processes," *IEEE Trans. Aerospace and Electronic Syst.,* 26(3): 517–525 (May 1990).
32. N. A. Carlson and M. P. Berarducci, "Federated Kalman Filter Simulation Results," *Navigation, J. Inst. Navigation,* 41(3): 297–321 (Fall 1994).
33. G. Minkler and J. Minkler, *Theory and Application of Kalman Filtering,* Palm Bay, FL: Magellan Book Co., 1993.
34. M. Athans and P. L. Falb, *Optimal Control,* New York: McGraw-Hill, 1966.
35. D. E. Kirk, *Optimal Control Theory,* Englewood Cliffs, NJ: Prentice-Hall, 1970.
36. A. E. Bryson, Jr., and Y. Ho, *Applied Optimal Control,* rev. ed., New York: Halsted Press, Div. of John Wiley & Sons, 1975.
37. A. P. Sage and C. C. White, *Optimum Systems Control,* 2nd ed., Englewood Cliffs, NJ: Prentice-Hall, 1977.
38. G. F. Franklin and J. D. Powell, *Digital Control of Dynamic Systems,* Reading, MA: Addison-Wesley, 1980.
39. C. T. Chen, *Linear System Theory and Design,* New York: Holt, Rinehart and Winston, 1984.
40. T. Kailath, *Linear Systems,* Englewood Cliffs, NJ: Prentice-Hall, 1980.

10

More on Modeling: Integration of Noninertial Measurements into INS

It has often been said that modeling is the "hardest" part of Kalman filtering. This is especially true when there are nonlinearities in the physical equations that must be linearized. Developing a good Kalman filter model is part art and part science. As a general rule, we look for models that are simple enough to be implementable, but yet, at the same time, still represent the physical situation with a reasonable degree of accuracy. This chapter is devoted to modeling examples from one of the most successful applications of Kalman filtering over the past three decades, namely, integrated inertial navigation systems.

10.1 COMPLEMENTARY FILTER METHODOLOGY

The use of Kalman filtering has probably benefited no other application area more so than navigation systems. The particular navigation systems problem that the Kalman filter has been very adept in solving is the integration of an inertial navigation system (INS) with navigation data from other sensors. Such data from diverse sensors provided the necessary and often times redundant data to piece together the information puzzle. However, the errors that corrupted or distorted the data from each sensor were characteristically different and varied in quality. The central problem in integration was to efficiently blend the available data together to extract the optimum navigation solution. In historical perspective, this problem had been around at least ten years before the Kalman filter solution became available, and it is still an important problem in navigation systems work.

Beneficial Features

Since the early 1960s, the complementary filter, introduced earlier in Chapter 4, has become the basis for this form of integration (1), and there are a number of good reasons for this choice; a block diagram of the general methodology using a complementary filter for such a function is shown in Fig. 10.1. First of all, the method has a degree of generality that allows for a wide variety of mixes of aiding measurement information. This is important because the combination of aiding sensors may vary during an individual mission, as well as in the broader sense over various suites of equipment. The Kalman filter readily accepts various mixes of aiding sources; this is all handled in the system software. Note that all the aiding measurements are processed in a single Kalman filter. Thus, this is an example of a *centralized* Kalman filter (see Section 9.6).

Another reason for choosing the complementary filter form of integration has to do with the restrictions placed on the Kalman filter model. Recall that the process dynamics and measurement relationship must both be linear. Frequently, the total state variables do not satisfy this requirement. For example, electronically made distance measurements are proportional to the square root of the sum of the squares of Cartesian components, and these are certainly not linear relationships. In navigation systems, measurement relationships involving multiple spatial dimensions are seldom linear. Therefore, the problem must be linearized about some reference trajectory in order to fit the format required by the Kalman filter. This reference trajectory can be a single point in vector space but is, in general, a time-varying trajectory. The details of the linearization procedure are described in Chapter 9. It should be apparent that the system integration scheme shown in Fig. 10.1 is, in fact, a regular linearized Kalman filter in every sense of the word. The process dynamics in the inertial system and the measurement relationships may be nonlinear. So be it. The nonlinearities are washed out in the differencing operation $[\mathbf{z} - \mathbf{h}(\mathbf{x}^*)]$, and the filter subproblem becomes linear, provided, of course, that the deviation of the reference trajectory from truth remains small throughout the time span of interest.

A third reason for choosing the complementary filter form of integration has to do with maintaining high dynamic response in the position, velocity, and attitude state variables. The usual price associated with filtering is time delay or sluggish response. For example, if a low-pass filter is energized with a step input, the response is exponential. The leading edge of the step input is rounded in the

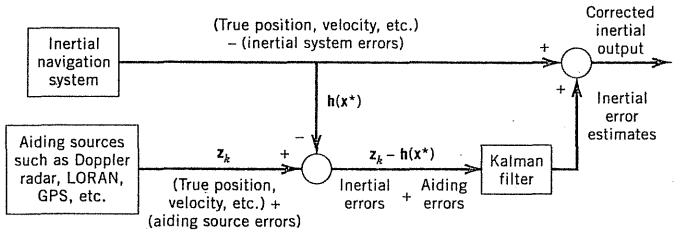


Figure 10.1 Integrated navigation system—feedforward configuration.

output and the degree of "rounding" is directly related to the amount of smoothing that is designed into the filter. This lag is undesirable in most real-time navigation applications. Normally, we want the navigation system to follow the dynamics of the aircraft faithfully, no matter how rapid the changes may be. The complementary filter philosophy accomplishes this and, at the same time, provides filtering of the measurement noise. At first glance, this may seem to be a contradiction, but this is made possible by intelligent management of the measurement redundancy. Note in Fig. 10.1 that the filter only operates on the combination of inertial system errors and the aiding source errors. The filter does not operate on the total dynamic quantities of interest; they pass through the system without any distortion or delay whatsoever. For this reason, this type of filtering is sometimes called *distortionless* filtering or *dynamically exact* system integration. Note also that the total dynamical quantities of interest (i.e., position, velocity, and attitude) do not have to be modeled as random processes. The filter only operates on the system errors, so these are the quantities that appear in the Kalman filter model.

Many real-life navigation systems have depended largely on the INS for providing the reference trajectory when using this type of integration philosophy because the INS is self-contained, continuous, and provides all the basic navigation quantities normally of interest—position, velocity, and attitude. None of the other sensors, when considered individually, can provide the same complement of information in quite the same way. Thus, the INS is the logical choice for the reference, even though by itself, its accuracy may be poorer than some of the aiding sources.

Additional Details on System Methodology

Some additional comments on Fig. 10.1 are in order before proceeding. First, it is tacitly assumed that in forming the difference $[\mathbf{z} - \mathbf{h}(\mathbf{x}^*)]$, we difference *like* quantities. For example, we do not try to compare inertially derived position directly with velocity as seen by the Doppler radar. Rather, we compare inertial *velocity* with Doppler radar *velocity;* furthermore, we compare the two in the same frame of reference. This often means that inertially derived quantities must be converted to another coordinate system before the appropriate comparison can be made. It should also be noted that each of the aiding sources does not have to measure all of the same dynamical quantities that are normally computed in the inertial system. If the "aiding" sources are really to aid, there needs to be at least one connection to the inertial quantities, but that is all. Usually, no one of the aiding sources (regardless of how accurate) provides all of these quantities and, therefore, it is convenient conceptually to lump all of the aiding sources together in the "aiding" category.

It is important to note that the INS measurements are not used purely as "measurements" but rather to form the reference trajectory against which the aiding data are compared. The reference trajectory consists of unfiltered data; position and velocity are extracted from measured inertial acceleration and attitude is formed from measured attitude rate. It is the aiding data less the reference trajectory that form the very "measurements" fed to the Kalman filter. Of the errors that the Kalman filter estimates, the inertial system errors are of prime interest, so we need a state-space model describing these errors, and it must be linear with white-noise processes as the driving functions. Various models have been used, and the model to be used in any particular application will depend on the type of inertial system at hand and the degree of complexity that the systems engineer is willing to live with in the design. This constitutes the main subject in the following sections of this chapter. In addition to modeling the inertial system errors, one must also augment the system state vector with any nonwhite aiding-source errors. Even though they may not be of primary interest in the navigation problem, they must, nevertheless, be carried along in the estimator if one is to have a properly modeled Kalman filter. The aiding-source error estimates are not, of course, used in the feedforward correction of the inertial system, but they are properly accounted for when forming the measurement residual where the entire state vector is involved.

Feedforward vs. Feedback Configuration

The block diagram shown in Fig. 10.1 is a configuration called the feedforward or open-loop configuration because the corrections to the INS output are utilized externally but are not returned to modify the INS internally. Although conceptually useful, this configuration is susceptible to divergence between the reference and actual trajectories so as to cause linearity assumptions and random process error models to gradually deteriorate. To avoid this, a feedback configuration such as that shown in Fig. 10.2 is an acceptable alternative. This configuration leads to an extended Kalman filter mode of operation in contrast to an ordinary linearized Kalman filter that is associated with the feedforward case.

One should be careful not to be too literal in the interpretation of the block diagrams of Figs. 10.1 and 10.2. They are intended to be conceptual, not literal. For example, the inertial error estimates fed back to the INS in Fig. 10.2 are "corrections" that must be made in a computer somewhere in the system. In some systems, this might be a central flight management computer that also does many other computational chores including, possibly, a separate feedforward solution. In other systems, the computational effort may be distributed among

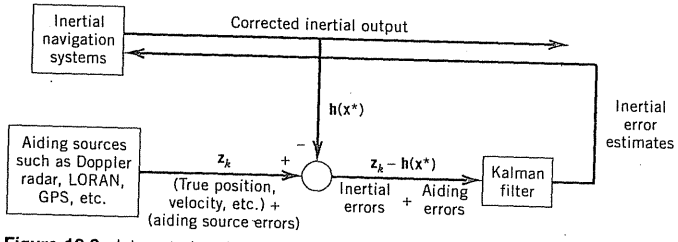


Figure 10.2 Integrated navigation system—feedback configuration.

various "boxes" in the system. Never mind which box does the computation; regardless, the key item is $\mathbf{h}(\mathbf{x}^*)$. If $\mathbf{h}(\mathbf{x}^*)$ is computed before the corrections are made to the inertial outputs, the filter is an ordinary linearized Kalman filter; if $\mathbf{h}(\mathbf{x}^*)$ is computed after the corrections are made, the filter is an extended Kalman filter. Both modes of operation have been used successfully, and each has its place. Both modes of operation have the same performance, within the linearity assumption. Clearly, though, the extended Kalman filter is to be preferred in applications where the mission time is long, as would be true of a ship at sea for many weeks or months. Otherwise, the reference trajectory would diverge from truth beyond acceptable limits. On the other hand, in a ballistic space vehicle launch, the mission time is short and, there is every reason to assume the actual trajectory will match the preprogrammed one closely. In this case, an ordinary linearized Kalman filter might well be preferred.

10.2 INS ERROR MODELS

An INS is made up of gyroscopes (gyros, for short) and accelerometers for basic sensors. A gyro senses rotational rate (angular velocity) that mathematically integrates to give overall change in attitude over time. Similarly, an accelerometer senses linear acceleration that integrates to give velocity change, or doubly integrates to give position change over time. An INS sustains attitude, position, and velocity accuracy by accurately maintaining changes in those parameters from their initial conditions. However, due to the integration process, errors in the attitude, position, and velocity data are inherently unstable but the growth characteristics of these errors depend on the type of sensors used. The level of complexity needed for the error modeling depends on the mix of sensors in the integration and the performance expected of it.

Single-Axis Inertial Error Model

We shall begin by looking at a simple model that contains the physical relationship between the gyro and the accelerometer in one axis. The following notation will be used:

Δx = position error

$\Delta \dot{x}$ = velocity error

$\Delta \ddot{x}$ = acceleration error

ϕ = platform (or attitude) error relative to level

g = gravitational acceleration

R_e = earth radius

a = accelerometer noise

ε = gyro noise in terms of angular rate

The single-axis model is instructive for the relationship it describes between the accelerometer sensor and the gyro sensor. Both sense inertial quantities, one linear acceleration and the other angular velocity. The differential equations that describe the accelerometer and the gyro errors are given as follows:

$$\Delta \ddot{x} = a - g\phi \tag{10.2.1}$$

$$\dot{\phi} = \frac{1}{R_e}\Delta \dot{x} + \varepsilon \tag{10.2.2}$$

In Eq. (10.2.1), the error in acceleration is fundamentally due to a combination of accelerometer sensor noise and a component of gravity the accelerometer senses as a result of platform error. The platform error rate, described by Eq. (10.2.2), results from gyro sensor noise and velocity error that, when projected along the earth's surface curvature, gets translated into an angular velocity error. An accelerometer error that integrates into a velocity error gives rise to a misalignment in the perceived gravity vector due to the earth's curved surface. This misalignment results in a horizontal component that fortuitously works against the effects of the initial accelerometer error. The resulting oscillation known as the Schuler oscillation provides some stability to the horizontal errors. Note that g is assumed to be the usual earth's gravitational constant. Thus, this simple model is restricted to low dynamics.

Three-Axes Inertial Error Model

In progressing from a single-axis to a level-platform three-axes INS, additional complexities arise from interaction among the three sensor pairs (2, 3). A sensor pair aligned in the north-south direction is shown in Fig. 10.3 as a transfer function block diagram denoted as the north channel. A very similar model exists for the east channel as shown in Fig. 10.4. Just as with the single-axis error model, the three-axes model is restricted to low dynamics. (See Problem 10.2 for more on this.)

The differential equations that accompany the transfer functions of Figs. 10.3 and 10.4 are given below. In the following notation, the platform rotation

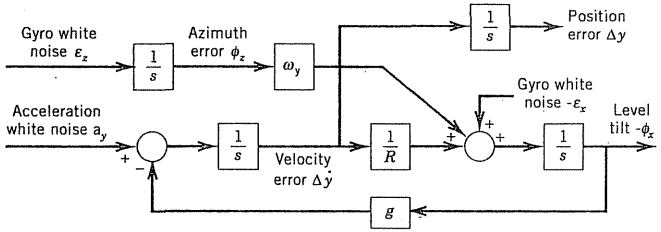


Figure 10.3 North channel error model (x is east, y is north, z is up, and ω_y = platform angular rate about y-axis).

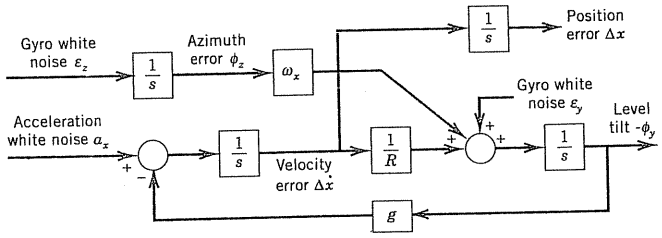


Figure 10.4 East channel error model. (x is east, y is north, z is up, and ω_x = platform angular rate about x-axis.)

rate ω is an angular velocity. Bear in mind, however, that ω is not the same as the platform tilt rate error $\dot{\phi}$, which is an angular velocity *error.*

North channel:

$$\Delta\ddot{y} = a_y - g(-\phi_x) \tag{10.2.3}$$

$$-\dot{\phi}_x = \frac{1}{R_e}\Delta\dot{y} + \omega_y\phi_z - \varepsilon_x \tag{10.2.4}$$

East channel:

$$\Delta\ddot{x} = a_x - g\phi_y \tag{10.2.5}$$

$$\dot{\phi}_y = \frac{1}{R_e}\Delta\dot{x} + \omega_x\phi_z + \varepsilon_y \tag{10.2.6}$$

Vertical channel:

$$\Delta\ddot{z} = a_z \tag{10.2.7}$$

Platform azimuth:

$$\dot{\phi}_z = \varepsilon_z \tag{10.2.8}$$

The north and east channel models take into account the previously described phenomenon that is due to the earth curvature. The vertical channel does not benefit from the Schuler phenomenon and is governed by a simpler model as shown in Fig. 10.5.*

* It can be shown that the characteristic poles for the vertical channel do not lie exactly at the origin in the s-plane. They are actually symmetrically located on the real axis, one slightly to the left of the origin and the other to the right (3). When the vertical error is observable, it is a good approximation in the Kalman filter model to assume that both poles are coincident at the origin.

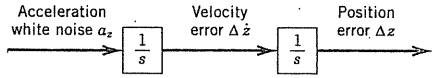


Figure 10.5 Altitude channel error model.

A basic 9-state dynamic model can be used as a foundation for an aided INS Kalman filter model. For our use here, the nine variables in the state vector will be ordered as follows:

$$\begin{aligned}
x_1 &= \text{east position error} \quad (\text{m}) \\
x_2 &= \text{east velocity error} \quad (\text{m/sec}) \\
x_3 &= \text{platform tilt about } y \text{ axis} \quad (\text{rad}) \\
x_4 &= \text{north position error} \quad (\text{m}) \\
x_5 &= \text{north velocity error} \quad (\text{m/sec}) \\
x_6 &= \text{platform tilt about } (-x) \text{ axis} \quad (\text{rad}) \\
x_7 &= \text{vertical position error} \quad (\text{m}) \\
x_8 &= \text{vertical velocity error} \quad (\text{m/sec}) \\
x_9 &= \text{platform azimuth error} \quad (\text{rad})
\end{aligned} \tag{10.2.9}$$

Based on Eqs. (10.2.3) through (10.2.8), we can write out the nine-dimensional vector first-order differential equation:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \\ \dot{x}_9 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -g & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{R_e} & 0 & 0 & 0 & 0 & 0 & 0 & \omega_x \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -g & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{1}{R_e} & 0 & 0 & 0 & \omega_y \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}}_{\mathbf{F}_{\text{INS}}} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \end{bmatrix} + \begin{bmatrix} 0 \\ u_{ax} \\ u_{gx} \\ 0 \\ u_{ay} \\ u_{gy} \\ 0 \\ u_{az} \\ u_{gz} \end{bmatrix} \tag{10.2.10}$$

From the parameters of Eq. (10.2.10), the discrete-time nine-dimensional vector first-order difference equation for the process model can be derived. Closed-form solutions for the process-noise covariance matrix $\mathbf{Q}$ or the state transition matrix $\boldsymbol{\Phi}_{\text{INS}}$ are not easily derived. These parameters are generally obtained with numerical approximations. The state transition matrix $\boldsymbol{\Phi}_{\text{INS}}$ may be approximated by $e^{F\Delta t} \approx \mathbf{I} + \mathbf{F}\Delta t$ for small Δt:

$$\boldsymbol{\Phi}_{\text{INS}} \approx \begin{bmatrix} 1 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -g\Delta t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\Delta t}{R_e} & 1 & 0 & 0 & 0 & 0 & 0 & \omega_x \Delta t \\ 0 & 0 & 0 & 1 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -g\Delta t & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\Delta t}{R_e} & 1 & 0 & 0 & \omega_y \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta t & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The INS process model has nine states in total comprising a position, a velocity, and a platform tilt state in each of three dimensions. This is a minimal system that accounts for platform misorientation, but only allows for very little complexity in errors associated with the accelerometers and gyros. In other words, the instrument errors are grossly simplified and modeled as white-noise forcing functions that drive the INS error dynamics. Also, platform angular errors are assumed to be small, and platform torquing rates are assumed to change very slowly such that they can be treated as constants. Small acceleration (relative to 1-g) is also assumed in the 9-state model given by Eq. (10.2.10). Even though the model is relatively simple, it is nonetheless workable and can be found in use in some real-life systems.

Sensor Noise Models

The acceleration and gyro error terms are simple white-noise inputs in Figs. 10.3 through 10.5. To add more fidelity to the model, additional error terms with a time-correlation structure can also be included. Generally, a first-order Gauss–Markov model is sufficient for each of these types of error. To do so, an additional six states, three for acceleration and three for gyro errors, are needed for augmenting the state vector. They are all of the form:

$$\text{Error process:} \quad x_{k+1} = \phi\, x_k + w_k \tag{10.2.11}$$

where the state transition parameter $\phi = e^{-\Delta t/\tau}$, Δt being the discrete-time interval and τ the time constant of the exponential autocorrelation function that governs this process (see Example 5.2). The variance of the process noise w, if it is time-invariant, is related to the steady-state variance of the x by the following:

$$\text{Var}(w_k) = (1 - \phi^2)\text{Var}(x_k)$$

When these additional error states are augmented to the basic 9-state model, the process model can be written in the following partitioned way:

$$\begin{bmatrix} \mathbf{x}_{1-9} \\ \hline x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \end{bmatrix}_{k+1} = \left[\begin{array}{c|cccccc} \boldsymbol{\Phi}_{\text{INS}} & \multicolumn{6}{c}{\mathbf{C}} \\ \hline & \phi_{ax} & 0 & 0 & 0 & 0 & 0 \\ & 0 & \phi_{ay} & 0 & 0 & 0 & 0 \\ \mathbf{0} & 0 & 0 & \phi_{az} & 0 & 0 & 0 \\ & 0 & 0 & 0 & \phi_{gx} & 0 & 0 \\ & 0 & 0 & 0 & 0 & \phi_{gy} & 0 \\ & 0 & 0 & 0 & 0 & 0 & \phi_{gz} \end{array}\right] \begin{bmatrix} \mathbf{x}_{1-9} \\ \hline x_{10} \\ x_{11} \\ x_{12} \\ x_{13} \\ x_{14} \\ x_{15} \end{bmatrix}_k + \begin{bmatrix} \mathbf{w}_{1-9} \\ \hline w_{ax} \\ w_{ay} \\ w_{az} \\ w_{gx} \\ w_{gy} \\ w_{gz} \end{bmatrix}_k$$

(10.2.12)

where $\mathbf{0}$ is a submatrix of zeros, and $\mathbf{C}$ is a submatrix that provides the appropriate additive connections of the augmented error states (x_{10} through x_{15}) to the INS dynamic equations:

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Table 10.1 gives a comparison of sensor error characteristics found in different inertial systems of varying qualities. Here, *high quality* refers to systems capable of stand-alone navigation and attitude sensing with excellent accuracy for extended durations of time (typically, hours). By comparison, *medium-quality* systems require external aiding to attain the performance offered by high-quality systems. Otherwise, medium-quality systems are capable of stand-alone opera-

Table 10.1 Comparison of Different Inertial Systems (4, 5)*

	INS Quality		
Sensor Parameters	**High**	**Medium**	**Low**
Gyro bias	<0.01°/h	0.1–1.0°/h	10°/h
Gyro white noise	3×10^{-5} °/sec/$\sqrt{\text{Hz}}$	0.001°/sec/$\sqrt{\text{Hz}}$	>0.001°/sec/$\sqrt{\text{Hz}}$
Accelerometer bias	10–50 μg	200–500 μg	>1000 μg
Accelerometer white noise	3–10 μg/$\sqrt{\text{Hz}}$	50 μg/$\sqrt{\text{Hz}}$	>50 μg/$\sqrt{\text{Hz}}$

*The *medium-quality* numbers given above correspond to the *low-quality* ones given by (4). The low quality given here are taken from (5). Note that the white-noise parameter values are stated in terms of the square root of the power spectral density.

tion over shorter durations. *Low-quality* systems require external aiding to provide useful performance and can only offer brief stand-alone operation.

Another level of sophistication that may be added to the accelerometer and gyro error models is that of accounting for scale factor error. The mathematical relationship that translates the physical quantity a sensor measures to the desired value representing that quantity generally involves a scale factor that may vary due to instrumentation uncertainties. This error may be estimated when the integrated system is in a highly observable state. Whether it is worthwhile to estimate this error depends on how significant the error is and its impact on the overall performance (see Problem 10.3).

10.3 DAMPING THE SCHULER OSCILLATION WITH EXTERNAL VELOCITY REFERENCE INFORMATION

It is well known that pure INS exhibit undamped oscillatory error characteristics with a period of 84 minutes. This is known as the Schuler oscillation. When undamped oscillatory systems are excited by random noise, the output grows statistically without bound (see Problem 3.12). This is, of course, undesirable for missions that last for even a few cycles of the natural oscillation. One way of mitigating the problem is to add viscous damping in much the same manner as might be done with a mechanical pendulum. To do this in an INS, an independent, noninertial source of vehicle velocity information must be available (e.g., Doppler radar). The block diagram given in Fig. 10.6 shows the traditional analog way of damping the Schuler oscillation. The key portion of the diagram is the part where the INS velocity is compared with the reference velocity, and then the difference is fed back through a scale factor to the accelerometer input. We now wish to look at how we might accomplish a similar result using the complementary filter methodology that was discussed in Section 10.1. In order to keep the modeling as simple as possible, we will only look at one horizontal channel of the INS. We will then expand the model later.

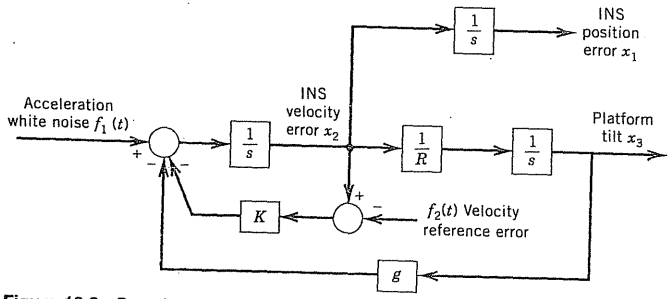


Figure 10.6 Damping of Schuler oscillation using external velocity reference.

We first note from Fig. 10.1 that the basic role of the Kalman filter is to estimate the inertial system errors. Thus, these quantities must be modeled as random processes in state-space form. The block diagram of Fig. 10.6 will serve us well in this regard, provided that we ignore the velocity reference feedback part of the diagram. (We only wish to model the unaided INS error propagation at this point.) In the interest of simplicity, we will assume the accelerometer error to be white with a known power spectral density. Suitable state equations may be obtained by choosing the three integrator outputs as state variables. They will be denoted as x_1, x_2, and x_3, and they are defined as follows:

$$x_1 = \text{INS position error} \quad \text{(m)}$$
$$x_2 = \text{INS velocity error} \quad \text{(m/sec)}$$
$$x_3 = \text{platform tilt} \quad \text{(rad)}$$

The differential equations are now obtained directly from the block diagram (if we ignore the velocity feedback part):

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = -gx_3 + f_1(t), \qquad f_1(t) \text{ is white noise with a spectral amplitude } A$$
$$\dot{x}_3 = \frac{x_2}{R_e} \tag{10.3.1}$$

Or, in the usual matrix form

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & -g \\ 0 & \frac{1}{R_e} & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} f_1(t) \tag{10.3.2}$$

We now need the discrete form of Eq. (10.3.2) for a step size of Δt, which is the interval between samples of the external velocity reference measurements. Reference velocity is assumed to be the only aiding source in this example. If Δt is small relative to the Schuler period, we can approximate the solution of Eq. (10.3.2) with just first-order terms in Δt. The result is

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_{k+1} = \underbrace{\begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & -g\Delta t \\ 0 & \dfrac{\Delta t}{R_e} & 1 \end{bmatrix}}_{\Phi_k} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_k + \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}_k \tag{10.3.3}$$

The transition matrix $\boldsymbol{\Phi}_k$ is obvious from Eq. (10.3.3). Derivation of the $\mathbf{Q}_k$ covariance matrix associated with the vector forcing function $[w_1 \quad w_2 \quad w_3]^T$ in Eq. (10.3.3) is left as an exercise at the end of the chapter (see Problem 10.5). Having determined the key process model parameters $\boldsymbol{\Phi}_k$ and $\mathbf{Q}_k$, we will look next at the measurement model.

The measurement presented to the Kalman filter is the difference between INS velocity and the aiding reference velocity. Note that the aiding source is only related directly to x_2 in this example, and not to the other two state variables. This still fits the format of Eq. (10.3.3), though, and we will account for this with the $\mathbf{H}_k$ parameter in the measurement model. First, however, it should be noted that we are dealing with the discrete Kalman filter, so we must assume that reference velocity measurements are available on a sampled basis with a sampling interval Δt. We also need to make the assumption that, after sampling, the sequence of measurement errors is a white (uncorrelated) sequence with a known variance $\mathbf{R}_k$. Putting the known measurement relationship in this case into mathematical form yields the Kalman filter measurement $\mathbf{z}$:

Kalman filter measurement
= −(true velocity − INS velocity error) + (true velocity
+ reference velocity error)
= (INS velocity error) + (reference velocity error)

Or, in terms of z_k and $\mathbf{x}_k$, we have

$$z_k = [0 \quad 1 \quad 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + v_k \tag{10.3.4}$$

The $\mathbf{H}_k$ parameter is obvious from Eq. (10.3.4) and R_k is the variance associated with v_k, the velocity reference measurement error. Presumably, R_k is known (or at least can be estimated) from the nature of the equipment being used for the noninertial velocity reference.

The initial conditions for the Kalman filter are chosen to correspond to the physical situation at hand when the velocity reference is "turned on." (In this simple example, the noninertial velocity reference is the only aiding source, so there is no Kalman filter operation until the velocity reference is operative.) Bear in mind that the state variables are inertial error quantities, so $\hat{\mathbf{x}}_0^-$ would normally be set equal to zero (in the absence of better information). The initial error covariance $\mathbf{P}_0^-$ is usually chosen to be diagonal with the individual terms (three in this example) being representative of the initial uncertainty in the state estimates.

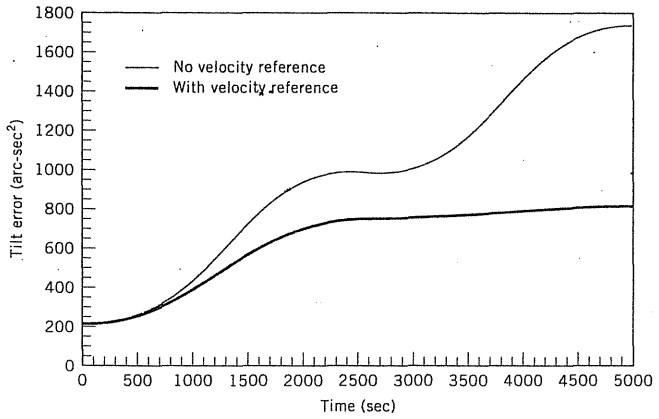


Figure 10.7 Error growth stabilization using an external velocity reference.

Figure 10.7 is intended to demonstrate the effectiveness of the Kalman filter in stabilizing the error growth in one level channel of an INS. The upper curve shows the mean-square platform error over a span of one Schuler cycle without the aiding velocity reference information. Clearly, it is ramping off in an unstable manner. The lower curve shows the same mean-square platform error with velocity reference information introduced via a Kalman filter. The filter was initialized in this case with the ideal situation of perfect alignment, and the filter parameters were chosen to be typical for damped inertial operation. It is clear from the plot that the oscillation is being damped, and that the platform error variance is approaching a finite upper bound.

EXAMPLE 10.1

INS/Doppler Radar Measurement Model Here, we will generalize the Schuler-damping problem to the full 9-state INS model. Doppler radar is a velocity-measuring system based on the Doppler shift in an electromagnetic wave that is reflected back from the ground (or water) after being transmitted by an aircraft. Although there are a variety of Doppler radar equipments, we shall simply consider the Doppler radar system to measure and output the aircraft horizontal velocity in terms of two orthogonal components: V_H in the direction of the aircraft heading, and V_D that is perpendicular to V_H (see Fig. 10.8). Thus, the Doppler radar supplies a 2-tuple velocity measurement that may be used to aid an INS. We will limit the discussion here to the development of the linearized

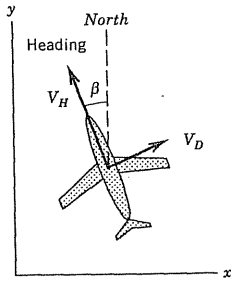


Figure 10.8 Aircraft velocity vector convention used in Example 10.1.

$\mathbf{H}_k$ matrix for the case where the measurement noise is assumed to be white, and the INS errors are modeled as a 9-state system (states described in Eq. 10.2.9).

Following the same system integration philosophy as before, we see that the INS must resolve its indicated horizontal velocity components V_x and V_y into the aircraft body frame of reference, which we will call heading and drift. The coordinate directions are shown in Fig. 10.8. Note that the azimuth angle β is the negative of the usual heading angle with respect to true north. We will assume that the INS platform azimuth angle β is used to resolve V_x and V_y into V_H and V_D. This resolution yields the INS "predicted" values of the two Doppler measurements. The idealized noiseless relationships are as follows:

$$V_H = -V_x \sin\beta + V_y \cos\beta$$
$$V_D = V_x \cos\beta + V_y \sin\beta \qquad (10.3.5)$$

Clearly, the measurement relationships are nonlinear in β, so they must be linearized. This can be done by forming the appropriate matrix of partial derivatives of V_H and V_D with respect to V_x, V_y, and β (see Section 9.1). The resulting linearized $\mathbf{H}_k$ matrix is then

$$\mathbf{H}_k = \begin{bmatrix} 0 & -\sin\beta & 0 & 0 & \cos\beta & 0 & 0 & 0 & (-V_x\cos\beta - V_y\sin\beta) \\ 0 & \cos\beta & 0 & 0 & \sin\beta & 0 & 0 & 0 & (-V_x\sin\beta + V_y\cos\beta) \end{bmatrix} \qquad (10.3.6)$$

The two Doppler measurements have been ordered with the heading component of velocity first and its drift component second. Also, it is important to note that the terms in the $\mathbf{H}_k$ matrix are time-varying in general, and they must be recomputed with each recursive step of the Kalman filter. This computation is made, of course, using the most current V_x, V_y, and β outputs of the INS.

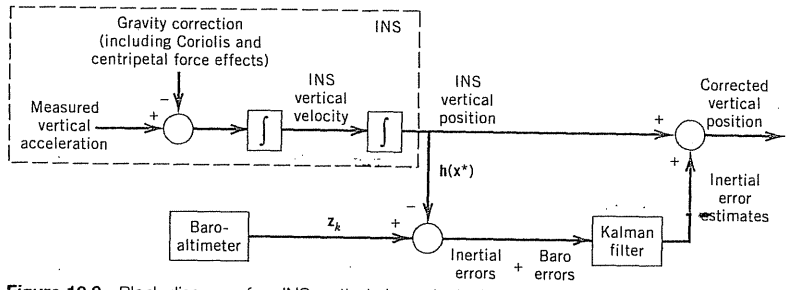
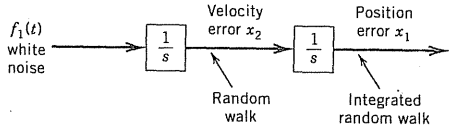


Figure 10.9 Block diagram of an INS vertical channel with baro-altimeter aiding.

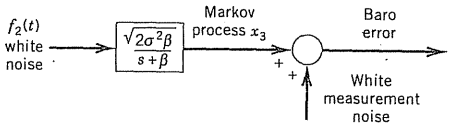
10.4 BARO-AIDED INS VERTICAL CHANNEL MODEL

Figure 10.9 shows a simplified block diagram of the vertical channel of an INS with aiding from a barometric altimeter. Note that this implementation follows the methodology discussed previously in Section 10.1 with the INS providing the reference trajectory. The INS in this case can be thought of as a strapdown system with a triad of accelerometers sensing total acceleration plus the gravity vector. In Fig. 10.9 we are looking only at the vertical component of the sensed acceleration vector. It is assumed, of course, that the dynamic reaction forces due to aircraft motion are properly accounted for in the gravity correction. The Kalman filter only operates on the system errors, so we now need to look at the way in which errors propagate in this system.

Figure 10.10 shows the error models that will be used for the vertical channel of the INS and the barometric altimeter. These models are much simplified, but they will serve our purpose here. In defense of the models, they do



(a) Inertial error



(b) Baro error

Figure 10.10 Error models used for the vertical channel shown in Fig. 10.9.

allow for nontrivial error propagation in both the INS and baro-altimeter. Clearly, the INS vertical channel has unstable error propagation. The double pole at the origin in the s-plane means that the mean-square response to white noise will grow without bound. On the other hand, the Markov model for the baro-altitude error says that this error will be bounded (stochastically, at least), and the model allows for considerable flexibility in the adjustment of the σ^2 and β parameters of the model. Adjustment for the rate of growth of the INS error is accomplished by the designer's choice of the spectral amplitude of the white-noise driving function in the INS channel. Thus, even though the 3-state model is quite simple, it does have some degree of flexibility in adapting to a variety of physical situations. We will now look at the process and measurement models for the Kalman filter in this situation.

Three state variables are necessary for the mathematical description of the models in Fig. 10.10 and they are defined as follows:

x_1 = INS vertical position error (m)

x_2 = INS vertical velocity error (m/sec)

x_3 = baro error (m)

The appropriate differential equations are obtained directly from the block diagrams, and they are

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= f_1(t) \\ \dot{x}_3 &= -\beta x_3 + \sqrt{2\sigma^2\beta}\cdot f_2(t) \end{aligned} \tag{10.4.1}$$

Equations (10.4.1) can be solved for a step size Δt, and the result in matrix form is

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_{k+1} = \underbrace{\begin{bmatrix} 1 & \Delta t & 0 \\ 0 & 1 & 0 \\ 0 & 0 & e^{-\beta\Delta t} \end{bmatrix}}_{\boldsymbol{\Phi}_k} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_k + \begin{bmatrix} w_1 \\ w_2 \\ w_3 \end{bmatrix}_k \tag{10.4.2}$$

The state transition matrix ϕ_k is obvious from Eq. (10.4.2). The $\mathbf{Q}_k$ matrix is the covariance associated with the forcing term in Eq. (10.4.2). It is not obvious, but it can be calculated with the help of random-process theory (see Chapter 3). The result is

$$\mathbf{Q}_k = \begin{bmatrix} A\dfrac{\Delta t^3}{3} & A\dfrac{\Delta t^2}{2} & 0 \\ A\dfrac{\Delta t^2}{2} & A\,\Delta t & 0 \\ 0 & 0 & \sigma^2(1 - e^{-2\beta\Delta t}) \end{bmatrix} \tag{10.4.3}$$

where A is the power spectral density of $f_1(t)$, and σ^2 and β are the Markov parameters for the baro error process.

The random-process part of the Kalman filter model is now complete. We will look at the measurement model next.

The desired measurement relationship is obtained directly from the block diagram shown in Fig. 10.9. In words, the relationship is

(Measurement presented to the Kalman filter)
= (true altitude + baro error) − (true altitude − INS altitude error)

Now we denote the discrete measurement as z_k, and note that x_1 is the INS position error and x_3 is the Markov part of the baro-altitude error. We will also say that the baro-altitude error has a white component in addition to the Markov component. The final measurement model in matrix form is then

$$z_k = \begin{bmatrix} 1 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + v_k \tag{10.4.4}$$

and the $\mathbf{H}_k$ matrix is obvious from Eq. (10.4.4). The R_k parameter is the variance associated with the white sequence v_k term in Eq. (10.4.4) and its numerical value will depend on the physical situation under consideration.

We will not attempt to assign numerical values to the various parameters in our INS baro-altimeter model (see Problem 10.1). It suffices here to say that, with reasonable values, the system effectively slaves the corrected inertial output to the baro-altitude measurement in the steady-state condition. However, when sharp transients occur, the system follows the INS-derived altitude. Said another way, the baro-altitude provides the low-frequency response, and the INS provides the high-frequency response. Thus, the Kalman filter mechanization is dynamically exact and accomplishes a result similar to the analog mechanization discussed in Problem 4.13. The Kalman filter has the advantage, though, of its digital precision and optimality (subject, of course, to the accuracy of the model). Also, with the uncorrected INS error growing without bound, it should be obvious that correcting the INS on an open-loop basis could eventually lead to numerical problems. Thus, closing the loop by using the feedback structure of Fig. 10.2 is highly desirable in this case.

In summary, the beauty of the INS/baro-altitude mechanization just presented is that it achieves the "best of both worlds"—the fast dynamic response of the INS coupled with the bounded error characteristics of the baro-altimeter.

10.5 INTEGRATING POSITIONING MEASUREMENTS

We shall consider here an integrated navigation system that is updated with positioning measurements. There are many types of positioning systems used with integrated navigation systems offering different levels of position accuracy and error characteristics. The distance-measuring equipment (DME) system mentioned in Chapter 9 is a two-dimensional (horizontal) ranging system of signals traveling to and from ground-based transmitters (6). OMEGA and LORAN are also examples of two-dimensional ranging systems (6). GPS, the premier positioning system available today (see Chapter 11), is a three-dimensional ranging system. A very-high-frequency omnidirectional ranging (VOR) system is a positioning system based on angular measurements of the direction of signals arriving from ground-based transmitters (6).

All positioning systems based on ranging principles are susceptible, in varying degrees, to ranging errors due to timing resolution and unpredictable atmospheric propagation effects. Positioning systems based on angular measurements are prone to errors in directional resolution and multipath. These errors are usually correlated between time samples, unless the sampling period is unusually long, because changes in the signal propagation path or the atmospheric characteristics affecting it evolve slowly. These errors can be modeled independently with first-order Gauss–Markov states augmented to the state vector in much the same way that accelerometer and gyro errors are accounted for in Eq. (10.2.12).

INS/DME Example

The linearization of a DME measurement was discussed in Section 9.1 (Example 9.1). There, the direct slant range from the aircraft to the DME ground station was considered to be the same as horizontal range. This approximation that can be in error by a few percent for short to medium ranges is usually absorbed into the measurement noise component of the model. Once the horizontal range is obtained, it is then compared with the predicted horizontal range based on the INS position output, and the difference becomes the measurement input to the Kalman filter. The difference quantity has a linear connection to Δx and Δy as discussed in Example 9.1. Based on the same ordering of the state vector as in Eq. (10.2.9), the rows of the $\mathbf{H}_k$ matrix corresponding to the two DME stations would then be

$$\mathbf{z}_k = \begin{bmatrix} -\sin\alpha_1 & 0 & 0 & -\cos\alpha_1 & 0 & 0 & 0 & 0 & 0 \\ -\sin\alpha_2 & 0 & 0 & -\cos\alpha_2 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{x}_k + \mathbf{v}_k \quad (10.5.1)$$

where we have written the direction cosines in terms of the bearing angle to the station α rather than θ as used in Eq. (9.1.18) (see Fig. 9.2). Note that the bearing angle is the usual clockwise angle with respect to true north, and the x-axis is assumed to be east. It is assumed, of course, that the coordinates of the DME station being interrogated are known and that the aircraft's position is known approximately from the INS position output. Thus, $\sin\alpha$ and $\cos\alpha$ are computable on-line to a first-order approximation. Range measurements from more than one DME station could be made either sequentially or simultaneously. The Kalman filter can easily accommodate to either situation by setting up appropriate rows in the $\mathbf{H}_k$ matrix corresponding to whatever measurements happen to be available.

EXAMPLE 10.2

Simulation of INS/DME For the following simulation exercise, we shall look at the performance of three different systems: (a) an integrated INS/DME system, (b) an INS-only system with initial alignment, (c) a DME-only system. The nominal aircraft motion and the DME station locations are as shown in the figure accompanying Problem 9.4 in Chapter 9. For the INS/DME system, we shall use the basic 9-state process model described in Eq. (10.2.10) and choose the following parameters for it:

Δt step size = 1 sec

Accelerometer white-noise spectral density = 0.0036 $(\text{m/sec}^2)^2/(\text{rad/sec})$

Gyro white-noise spectral density = 2.35 (10^{-11}) $(\text{rad/sec})^2/(\text{rad/sec})$

R_e = 6,380,000 m

Y-axis angular velocity ω_y = 0.0000727 rad/sec (earth rate at the equator)

X-axis angular velocity $\omega_x = -\dfrac{100 \text{ m/sec}}{R_e} = -0.0000157$ rad/sec

Initial position variance = 10 m^2

Initial velocity variance = $(0.001 \text{ m/s})^2$

Initial attitude variance = $(0.001 \text{ rad})^2$

DME white measurement error = $(15 \text{ m})^2$

The INS-only system uses the same inertial sensor parameters and the same initial alignment conditions. The main difference between the INS-only system and the integrated INS/DME system is that the DME measurements are never processed, whereas the INS errors are allowed to propagate according to the natural dynamics modeled.

For the DME-only system, the aircraft motion is modeled as a random walk (in both x- and y-positions) superimposed on constant-velocity motion in the y-direction. The filter in this case is a simple 2-state Kalman filter linearized about the nominal trajectory. The following parameters are used:

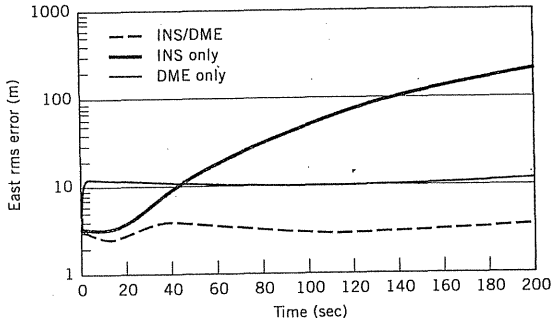
Δt step size = 1 sec

Process noise variance (in each position axis) = 400 m^2

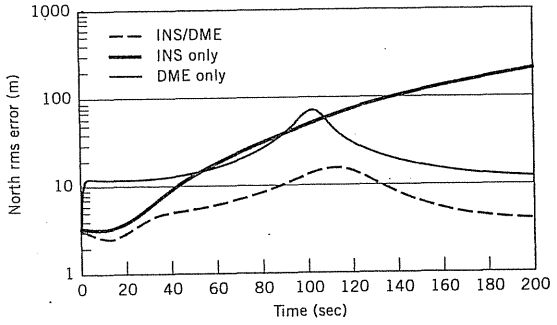
Initial position variance = 10 m^2

DME white measurement error = $(15\ m)^2$

In the 200-sec run, the aircraft starts out at the location (0, −10,000) and flies north at 100 m/sec and nominally ends up at the location (0, 10,000). Figure 10.11 shows a comparison of the standard deviations of the position error for



(a)



(b)

Figure 10.11 Comparison of various combinations between INS and DME sensors showing rms error time profiles for (*a*) the east component and (*b*) north component.

all three systems described above, for the east component (a) and the north component (b). Although the INS position error grows without bound, the position errors are stable and smaller for the integrated INS/DME system. The cresting of the north position error near the 100-sec time mark is due to the poor DME observability in the north direction when the aircraft crosses the x-axis (see Problem 9.4). The position error for the DME-only system has characteristics similar but slightly larger in magnitude by comparison to those of the integrated INS/DME position error. ▩

10.6 OTHER INTEGRATION CONSIDERATIONS

The integration philosophy discussed here has found wide use in navigation systems over the past three decades. It is clearly a philosophy that is centered around the use of an INS primarily because this type of sensor, better than any other, is capable of providing a reference trajectory representing position, velocity, and attitude with a high degree of continuity and dynamical fidelity. It is logical to then ask: If an INS is not available, can this integration philosophy still be useful? In general, any sensor that is capable of providing a suitable reference trajectory with a high level of continuity and dynamical fidelity can be used in place of the INS. In some applications, the reference trajectory need only consist of subsets of position, velocity, and attitude. An example of a reference sensor for a land vehicle is a wheel tachometer that senses speed, integrated with a compass that senses direction, to produce a velocity measurement and position through integration; attitude is not available, nor perhaps, necessary. In an aircraft, a suitable reference sensor might be derived from a combination of true air speed data with a magnetic compass for some applications.

In an integration exercise, it is always useful to keep the error stability characteristics of each sensor in mind while the details of modeling are being hashed out. In the two examples discussed in this section, INS/DME and INS/Doppler radar, it is important to note that in each case the aiding source serves the INS in a very different way. As was pointed out in Section 10.2, the position, velocity, and attitude errors produced by an INS are unstable over time. As a positioning system, the horizontal position errors produced in DME measurements are stable. Thus, an INS/DME system has stable horizontal position and velocity errors. Since the DME provides no vertical position information, the vertical position error and consequently the attitude errors are unstable over time. So if an INS/DME system is required to produce a stable three-dimensional position, velocity, and attitude data output, it will require more aiding information on vertical position, perhaps from a stable source like a baro-altimeter.

In the other integration example, the Doppler radar provides a stable source of velocity data, but when this is integrated into position, the errors become unstable over time. Thus, an INS/Doppler radar system has stable horizontal velocity errors but its horizontal position errors are unstable. The stability of its vertical position and velocity components must also be addressed in the same manner as for the case of the INS/DME.

To close, one final comment is in order. It should be apparent that the system integration philosophy presented in this chapter is not the only way of integrating inertial measurements with noninertial data. One only has to peruse navigation conference proceedings over the years to find countless integration schemes, each with a little different twist because of some special circumstance or choice. The scheme presented here is optimal though (within the constraint of dynamic exactness in the use of inertial measurements), and it represents the best way of managing the system integration theoretically. However, the systems engineer often does not have the luxury of designing a truly optimal system, and must at times settle for something less because of equipment constraints, cost, and so forth. Even so, the optimal methodology presented here is still valuable for analysis purposes, because it provides a lower bound on system errors for a given mix of sensors. That is, the optimal system serves as a "yardstick" that can be used for comparison purposes in evaluating the degree of suboptimality of various other candidate integration schemes.

PROBLEMS

10.1. A Kalman filter model for integrating vertical acceleration and baro-altitude measurements was given in Section 10.4. Consider the following set of numerical values for this Kalman filter implementation:

Δt step size = 1 sec
Accelerometer white-noise spectral density = 0.13889 $(m/sec^2)/(rad/sec)$
Markov baro error variance and inverse time constant:
$\sigma^2 = (100\ m)^2$
$\beta^{-1} = 300$ sec
White component of baro error = $(10\ m)^2$

We will assume that the system is "turned on" at $t = 0$ with perfect knowledge of altitude.

(a) Based on the above assumptions, derive the numerical values for the key Kalman filter parameters $\boldsymbol{\Phi}_k$, $\mathbf{Q}_k$, $\mathbf{H}_k$, $\mathbf{R}_k$, $\mathbf{P}_0^-$.

(b) After setting up the above Kalman filter model, cycle through 1000 steps of a covariance analysis for the given filter parameters. By the end of the run, at where the filter can be considered to have reached a steady-state condition, explain the relationship between the variances of the first and third states.

10.2. The approximate INS error equations of (10.2.3) through (10.2.8) are for a slow-moving vehicle. In this model, the observability of the azimuth error ϕ_z is poor because it can only depend on earth motion (gyrocompassing). Hence, for an INS with poor gyro stability, its steady-state azimuth error can be quite large. For a faster-moving vehicle that occasionally encounters horizontal acceleration, the improved observability of ϕ_z under such conditions actually provides a substantial reduction in the error, thus momentarily stabilizing it. The INS error equations for the east and north channels (Eqs. 10.2.3 through 10.2.6)

are rewritten here with the inclusion of the lateral acceleration components A_x and A_y tie-in to the azimuth error in the horizontal acceleration error equations (additional terms are indicated with a double underscore).

East channel:

$$\Delta\ddot{x} = a_x - g\phi_y \underline{\underline{+ A_y\phi_z}}$$
$$\dot{\phi}_y = \frac{1}{R_e}\Delta\dot{x} + \omega_x\phi_z + \varepsilon_y \qquad \text{(P10.1)}$$

North channel:

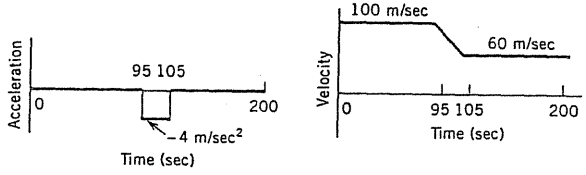
$$\Delta\ddot{y} = a_y - g(-\phi_x) \underline{\underline{- A_x\phi_z}}$$
$$-\dot{\phi}_x = \frac{1}{R_e}\Delta\dot{y} + \omega_y\phi_z + \varepsilon_x \qquad \text{(P10.2)}$$

Using the parameters for Example 10.2 in the integrated INS/DME navigation system, perform a covariance analysis to determine the time profile for the variance of the azimuth error ϕ_z for the following dynamic scenario: The nominal y-axis acceleration and velocity profiles are as shown in the accompanying figure.

The $y(t)$ profile for linearization of the $\mathbf{H}$ matrix may be approximated as constant-velocity (100 m/sec) for the first 95 sec; then a reduced constant velocity of 80 m/sec during the 10-sec deceleration period; and, finally, a constant velocity of 60 m/sec for the remaining 95 sec of the profile. (Note that we are not assuming this to be the *actual* flight path. This is simply the approximate reference trajectory to be used for the linearization.)

The parameter values given in Example 10.2 are to be used here, except that the gyro white noise power spectral density is to be increased to 2.35 (10^{-9}) $(rad/sec)^2/(rad/sec)$ and the initial azimuth error is to be set at 1 degree rms. This is intended to simulate a less-stable INS as compared with the one considered in Example 10.2.

The gravity constant g and the earth radius R_e may be assumed to be 9.8 m/sec^2 and 6.37 (10^6) m for this problem.



10.3. Instrument errors are often found to be simple quasibiases that wander over long time constants. These can simply be modeled with a single-state

Gauss–Markov process as was pointed out in Section 10.2. Some instrument errors, however, are related in a deterministic way to the magnitude of the measured variable, the most common type being known as *scale factor* errors. We shall look at the nature of a scale factor error in combination with a bias error in this problem that involves barometric-derived altitude. Suppose that the principal relationship between the internally sensed barometric reading and the reported altitude is given by the following equations:

$$H' = b'\,\gamma'$$

where

H' = reported altitude
b' = barometric reading
γ' = barometric altitude scale factor

Consider that the barometric reading b' is made up of the correct value b plus a bias error b_ϵ: $b' = b + b_\epsilon$. Consider also that the scale factor γ' is made up of the correct value γ plus an error γ_ϵ: $\gamma' = \gamma + \gamma_\epsilon$.

(a) Show that we can use the 2-state measurement model shown below to account for the bias and scale factor errors (neglect second-order effects):

$$H' - H = z_k = [1 \quad H'] \begin{bmatrix} b_\epsilon \\ \gamma_\epsilon \end{bmatrix}_k + v_k$$

(b) Suppose that the two error states are modeled as random constants:

$$\begin{bmatrix} b_\epsilon \\ \gamma_\epsilon \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} b_\epsilon \\ \gamma_\epsilon \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Let $H' = 50$. Do the variances of b_ϵ and γ_ϵ go to zero in the limit? Under what condition will the variances of b_ϵ and γ_ϵ go to zero in the limit?

(c) Suppose that the two error states are modeled individually as single-state Gauss–Markov processes:

$$\begin{bmatrix} b_\epsilon \\ \gamma_\epsilon \end{bmatrix}_{k+1} = \begin{bmatrix} 0.9999 & 0 \\ 0 & 0.9999 \end{bmatrix} \begin{bmatrix} b_\epsilon \\ \gamma_\epsilon \end{bmatrix}_k + \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}_k$$

where

$$E(\mathbf{w}\mathbf{w}^T) = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

Let $H' = 50$. Do the variances of b_ϵ and γ_ϵ go to zero in the limit? Why?

10.4. Suppose that the integrated INS/DME situation given in Example 10.2 involves a locomotive instead of an aircraft. The locomotive is constrained to railroad tracks aligned in the north–south direction. In place of an INS, the integrated navigation system uses wheel tachometer data to provide the reference trajectory in the complementary filter arrangement. A tachometer monitors wheel revolutions to determine the relative distance traveled. In words,

Relative distance = number of wheel revolutions × wheel circumference

The model for the position error in the relative distance takes on the same form as that of a scale factor error (see Problem 10.3).

The locomotive kinematics are described by the following equations:

$$x = 0$$
$$y_{k+1} = y_k + w_k + 10\,\Delta t, \qquad k = 0, 1, 2, \ldots, 2000$$

where w_k is an independent Gaussian sequence described by

$$w_k \sim N(0, 1\ \text{m}^2)$$

The sampling interval Δt is 1 sec.

(a) Formulate a process model that includes for bias and scale factor error states using random constant models for each. Also formulate a linearized measurement model using the scenario setup from Problem 9.4; use only DME station No. 2. For simplicity, a linearized Kalman filter should be used, not an extended Kalman filter. Let the initial estimation error variances for the bias and scale factor states be $(100\ \text{m})^2$ and $(0.02\ \text{per unit})^2$.

(b) Run a covariance analysis using the filter parameters worked out in (a) for $k = 0, 1, 2, \ldots, 2000$, and plot the rms estimation errors for each of the error states. Also plot the rms position error.

(c) Make another run of (b) except that, between $k = 1000$ and $k = 1800$, DME measurement updates are not available.

10.5. In the Schuler damping example of Section 10.3, the differential equation describing the dynamic process is given by Eq. (10.3.2). The state transition matrix computed for the discrete-time process difference equation in Eq. (10.3.3) is simply approximated by $\boldsymbol{\phi} \approx \mathbf{I} + \mathbf{F}\Delta t$. This same first-order approximation for $\boldsymbol{\phi}$ can also be used in the integral expression for $\mathbf{Q}_k$ given by Eq. (5.3.6). When $\mathbf{F}$ is constant and $\boldsymbol{\phi}$ is first order in the step size, it is feasible to evaluate the integral analytically and obtain an expression for $\mathbf{Q}_k$ in closed form. (Each of the terms in the resulting $\mathbf{Q}_k$ are functions of Δt.)

(a) Work out the closed-form expression for $\mathbf{Q}_k$ using a first-order approximation for $\boldsymbol{\phi}$ in Eq. (5.3.6). Call this $\mathbf{Q}1$.

(b) Next, evaluate $\mathbf{Q}_k$ with MATLAB using the numerical method described in Section 5.3 (Eqs. 5.3.23–5.3.26). Do this for $\Delta t = 5$ sec, 50 sec, and

500 sec. These will be referred to as Q2 (different, of course, for each Δt).

(c) Compare the respective diagonal terms of Q1 with those of Q2 for $\Delta t = 5$, 50, and 500 sec.

This exercise is intended to demonstrate that one should be wary of using first-order approximations in the step size when it is an appreciable fraction of the natural period or time constant of the system.

REFERENCES CITED IN CHAPTER 10

1. R. G. Brown, "Integrated Navigation Systems and Kalman Filtering: A Perspective," *Navigation, J. Inst. Navigation, 19*(4): 335–362 (Winter 1972–73).
2. G. R. Pitman (ed.), *Inertial Guidance,* New York: Wiley, 1962.
3. J. C. Pinson, "Inertial Guidance for Cruise Vehicles," in C. T. Leondes (ed.), *Guidance and Control of Aerospace Vehicles,* New York: McGraw-Hill, 1963.
4. D. Tazartes, R. Buchler, H. Tipton, and R. Grethel, "Synergistic Interferometric GPS-INS," *Proceedings of the National Technical Meeting of the Institute of Navigation,* Anaheim, CA, Jan. 18–20, 1995, pp. 657–671.
5. *GIC-100 Inertial Measurement Sensor* product description, Rockwell International, 1993.
6. M. Kayton and W. R. Fried (eds.), *Avionics Navigation Systems,* New York: Wiley, 1969.

11

The Global Positioning System: A Case Study

The Global Positioning System (GPS) is a satellite-based system that has demonstrated the provision of unprecedented levels of positioning accuracy, leading to its extensive use in both military and civil arenas (1, 2). It became fully operational in 1994 and provides worldwide coverage that benefits all nations of the world. At its conception, GPS was, by far, the most ambitious navigation project ever undertaken by the United States, or by any nation for that matter. Now, in a mature state, the applications it has spawned go beyond the usual positioning of aircraft and ships. Other applications include precise surveying, accurate land vehicle tracking, near-earth space navigation, and precise time dissemination on a worldwide basis. The central problem for the GPS receiver is the precise estimation of position, velocity, and time based on noisy observations of the satellite signals. It should come as no surprise then that this is an ideal setting for Kalman filtering. In fact, Kalman filtering has become a household word in the GPS business. Our discussion of the subject here is intended to be tutorial and must be brief. Thus, we will confine our attention to receiver applications only, and we will leave all of the other interesting facets of Kalman filtering applied to GPS as extracurricular reading.

11.1 DESCRIPTION OF GPS

GPS is a satellite-based navigation system that allows a user with the proper equipment access to useful and accurate positioning and timing information anywhere on the globe. Position and time determination is accomplished by the reception of GPS signals to obtain ranging information as well as messages transmitted by the satellites. The system of satellites that makes up the space segment of GPS consists of 24 satellites in six 12-hour orbits. This ensures a

user located anywhere on the globe a visibility of four satellites or more at any time. From an observer's viewpoint, the satellite geometry in the visible sky is always changing because the satellites are not in geosynchronous orbits. The maintenance of updated information embedded in the transmitted message is performed by ground monitoring stations collectively known as the control segment. The control segment periodically updates the information that is disseminated by all the satellites. This includes satellite ephemerides and health status, as well as a constellation almanac.

GPS signals are transmitted on two coherent carrier frequencies, L1 (1575.42 MHz) and L2 (1227.60 MHz), which are modulated by various spread spectrum signals. The major carrier, L1, is biphase-modulated by two types of pseudorandom noise codes (see Section 2.14 of Chapter 2): one at 1.023 MHz called the C/A-code, and the other at 10.23 MHz called the Y-code. The Y-code is intended only for authorized access because its one-chip wavelength of 30 m provides the most accurate positioning possible. The C/A-code, with its 300-m one-chip wavelength, is used in all cases for initial acquisition and code-signal alignment purposes. All users have access to this less accurate C/A-code for positioning. The second carrier signal L2 contains only Y-code modulation, and is intended to give authorized users the additional capability of actually measuring the ionospheric delays using the two frequencies, the delays being frequency-dependent. In official parlance, Y-code access is reserved for what is called the *Precise Positioning Service* (PPS) mode of operation, whereas everything else is classified as the *Standard Positioning Service* (SPS).

A 50 bits/sec navigation message is also combined with the pseudorandom noise codes. This navigation message, 1500 bits in length and repeated every 30 sec, carries many kinds of information with varying degrees of functional and operational importance to the user. Foremost in significance is the satellite ephemerides, a collection of data making up 60 percent of the message that, when decoded, uniquely describes the position and trajectory of the satellite that transmitted it. The remaining 40 percent of space allotted to the message is common to all satellites and carries general almanac information. This information runs the gamut from providing approximate satellite positions for visibility checks and signal acquisition purposes to satellite health status and current operational modes.

Position Determination

An observer equipped to receive and decode GPS signals must then solve the problem of position determination. In free space, there are three dimensions of position that need to be solved. Also, an autonomous user is not expected to be precisely synchronized to the satellite system time initially. In all, the standard GPS positioning problem poses four variables that can be solved from the following system of equations, representing measurements from four different satellites:

$$\begin{aligned}
\psi_1 &= \sqrt{(X_1 - x)^2 + (Y_1 - y)^2 + (Z_1 - z)^2} + c\Delta t \\
\psi_2 &= \sqrt{(X_2 - x)^2 + (Y_2 - y)^2 + (Z_2 - z)^2} + c\Delta t \\
\psi_3 &= \sqrt{(X_3 - x)^2 + (Y_3 - y)^2 + (Z_3 - z)^2} + c\Delta t \\
\psi_4 &= \sqrt{(X_4 - x)^2 + (Y_4 - y)^2 + (Z_4 - z)^2} + c\Delta t
\end{aligned} \tag{11.1.1}$$

where

$\psi_1, \psi_2, \psi_3, \psi_4$ = noiseless pseudorange
$[X_i, Y_i, Z_i]^T$ = Cartesian position coordinates of satellite i
$[x, y, z]^T$ = Cartesian position coordinates of observer
Δt = receiver offset from the satellite system time
c = speed of light

The observer position $[x, y, z]^T$ is "slaved" to the coordinate frame of reference used by the satellite system. In the case of GPS, this reference is a geodetic datum called WGS-84 (for World Geodetic System of 1984) that is earth-centered earth-fixed (3). The datum also defines the ellipsoid that crudely approximates the surface of the earth (see Fig. 11.1). Although the satellite positions are reported in WGS-84 coordinates, it is sometimes useful to deal with a locally level frame of reference, where the x'–y' plane is tangential to the surface of the earth ellipsoid. As depicted in Fig. 11.1, we shall define such a locally level reference frame by having the x'-axis pointing east, the y'-axis north, and the z'-axis normal to the plane, and equivalently, to the surface of the earth at the observer's location. It suffices here to say that the coordinate transformations to convert between the WGS-84 coordinates and any other derived

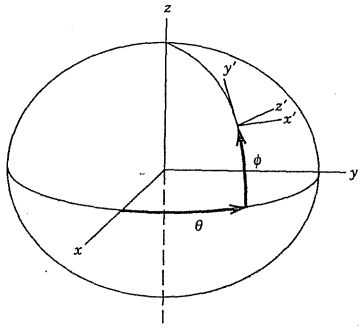


Figure 11.1 The WGS-84 coordinate reference frame (x, y, z) used by GPS and a locally level coordinate reference frame (x', y', z').

reference frame, including the locally level one given here, are usually quite straightforward.

Measurement Linearization

The measurement situation for GPS is clearly nonlinear from Eq. (11.1.1). Linearization of a measurement of this form has already been covered in Section 9.1 and will not be reiterated here. We will simply evaluate the partial derivatives necessary to obtain the linearized equations about an approximate observer location $\mathbf{x}_0 = [x_0, y_0, z_0]^T$. This nominal point of linearization $\mathbf{x}_0$ is sometimes based on an estimate of the true observer location $\mathbf{x}$ although, in general, its choice may be arbitrary.

$$\frac{\partial \psi_i}{\partial x} = -\frac{(X_i - x_0)}{\sqrt{(X_i - x_0)^2 + (Y_i - y_0)^2 + (Z_i - z_0)^2}}$$

$$\frac{\partial \psi_i}{\partial y} = -\frac{(Y_i - y_0)}{\sqrt{(X_i - x_0)^2 + (Y_i - y_0)^2 + (Z_i - z_0)^2}}$$

$$\frac{\partial \psi_i}{\partial z} = -\frac{(Z_i - z_0)}{\sqrt{(X_i - x_0)^2 + (Y_i - y_0)^2 + (Z_i - z_0)^2}} \tag{11.1.2}$$

for $i = 1, \ldots, 4$.

From a geometrical perspective, the partial derivative vector for each satellite i,

$$\left[\frac{\partial \psi_i}{\partial x} \quad \frac{\partial \psi_i}{\partial y} \quad \frac{\partial \psi_i}{\partial z}\right]^T$$

as given in Eq. (11.1.2), is actually the unit direction vector pointing from the satellite to the observer, the direction being specified by the negative sign in the equation. In classical navigation geometry, the components of this unit vector are often called *direction cosines*. The resulting measurement vector equation for pseudorange as the observable is then given by (without noise):

$$\begin{bmatrix} \psi_1 \\ \psi_2 \\ \psi_3 \\ \psi_4 \end{bmatrix} - \begin{bmatrix} \hat{\psi}_1(\mathbf{x}_0) \\ \hat{\psi}_2(\mathbf{x}_0) \\ \hat{\psi}_3(\mathbf{x}_0) \\ \hat{\psi}_4(\mathbf{x}_0) \end{bmatrix} = \begin{bmatrix} \frac{\partial \psi_1}{\partial x} & \frac{\partial \psi_1}{\partial y} & \frac{\partial \psi_1}{\partial z} & 1 \\ \frac{\partial \psi_2}{\partial x} & \frac{\partial \psi_2}{\partial y} & \frac{\partial \psi_2}{\partial z} & 1 \\ \frac{\partial \psi_3}{\partial x} & \frac{\partial \psi_3}{\partial y} & \frac{\partial \psi_3}{\partial z} & 1 \\ \frac{\partial \psi_4}{\partial x} & \frac{\partial \psi_4}{\partial y} & \frac{\partial \psi_4}{\partial z} & 1 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ c\Delta t \end{bmatrix} \tag{11.1.3}$$

where

ψ_i = noiseless pseudorange

$\mathbf{x}_0$ = nominal point of linearization based on $[x_0, y_0, z_0]^T$ and predicted receiver time

$\hat{\psi}_i(\mathbf{x}_0)$ = predicted pseudorange based on $\mathbf{x}_0$

$[\Delta x, \Delta y, \Delta z]^T$ = difference vector between true location $\mathbf{x}$ and $\mathbf{x}_0$

$c\Delta t$ = range equivalent of the receiver timing error

11.2 THE OBSERVABLES

Useful information can be derived from measurements made on the pseudorandom code and the carrier signal. The block diagram of a generic signal-tracking scheme for a GPS receiver is shown in Fig. 11.2. In it, there are separate tracking loops for the code and the carrier, with the latter cross-feeding to dynamically aid the code tracking (4). The loop filters are generally simple low-pass filters, the bandwidths of which determine the noisiness of the measurement data that are synthesized.

The observable known as *pseudorange* is a timing measurement of the propagation delay that is due to the geometric range from the transmitting satellite to the receiver and also the receiver clock offset from satellite time—hence, the term pseudorange and not just range. At the point of reception, the measurement is made in the receiver by determining the amount of shift in the pseudorandom code position since the time of transmission. By using some coherent form of signal tracking, the binary pseudorandom code can be monitored by aligning the received signal with a replica of the known code generated by the receiver. Hence, the precision of the crosscorrelation process in determining the pseudorandom code position establishes the accuracy of the pseudorange measurement, which by state-of-the-art standards is considered to be about

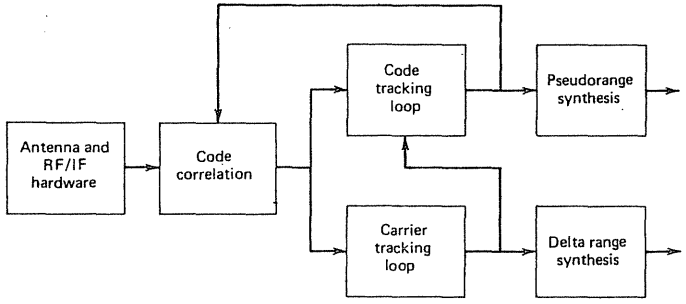


Figure 11.2 Generic signal-tracking scheme for a GPS receiver.

1 m under nominal signal reception strengths (4). In the context of Kalman filtering, these numbers represent the standard deviation of the measurement noise white sequence. The pseudorange measurement can be represented by the following equation:

$$\rho = \psi + \beta_\rho + \nu_\rho \tag{11.2.1}$$

where

ψ = noiseless pseudorange consisting of geometric range and range error due to receiver timing error

β_ρ = time-correlated errors associated with pseudorange

ν_ρ = pseudorange measurement noise

The term β_ρ represents other significant error sources, which may range from estimation errors in the reported satellite position and signal delays due to ionospheric and tropospheric refraction, to the intentional errors invoked under an accuracy-degradation scheme called selective availability. Sometimes collectively called *unmodeled biases,* these errors are generally difficult to estimate due to their poor observability characteristics and discussion of their modeling is deferred to Section 10.3. However, because they are dependent on the spatial relationship between the observer and the satellites, two observers in proximity to one another and sighting the same satellites will encounter these errors with a high degree of correlation between them. To take advantage of this, if one of these observers is at a known reference location, these so-called unmodeled biases can be measured as a lumped error and the information, when shared, can be used by the other observer to correct for the error. This mode of operation is known as *differential positioning* and such corrections are very effective in enhancing position accuracy for all participants of such a positioning network (see Section 11.7). Today, differential GPS plays an important role in the development of high-accuracy positioning systems (1, 5, 6).

In addition to code tracking, most GPS navigation receivers possess the capability of tracking the carrier signal as well. The wavelength of the L1 carrier is little more than 19 cm, thus allowing very precise measurements of the phase of the carrier to be made. This type of measurement is made after the code modulation has been stripped off. The amount of noise in the carrier phase data depends largely on the parameters of the tracking loops used by the receiver. It may be less than 1 percent of the wavelength for a stationary receiver, but to maintain carrier tracking in high dynamics, the noise in the data for most navigation-type receivers may be as high as 2 percent (4, 7).

In most conventional GPS receivers, the rate of change in the carrier phase over a brief interval is used to represent the measured Doppler frequency. As an approximation to the true Doppler frequency, this observable, called *delta range,* is generally used to provide accurate velocity information. Although the Doppler frequency due to satellite motion as seen by a stationary observer is constantly changing, it is nevertheless accurately predictable. Hence, when the measured Doppler is compared against the predicted value, the difference reflects the velocity of the observer as well as the frequency error of the receiver clock.

The use of the delta range measurement to approximate the Doppler frequency, of course, makes the assumption that the velocity is constant throughout the integration interval used to form the delta range. This assumption holds for the most part even if the velocity is not constant, provided that the integration interval remains short. Most receiver designs utilize integration intervals that are no more than a fraction of a second long. The accuracy of the delta range measurement in terms of velocity under nominal signal reception strengths is about 0.008 m/sec (4).

More recently, applications have surfaced whereby the Doppler data are exploited for information related to relative position rather than just velocity (8, 9). The difference between the integrated Doppler (sometimes also called *continuous carrier phase*) and the delta range is graphically compared in Fig. 11.3. The applications that utilize integrated Doppler measurements (discussed later in Sections 11.6 and 11.7) involve long integration intervals that demand uninterrupted and flawless tracking of the carrier. Under real operating conditions, such demands are seldom easily satisfied. But owing to the great benefits of accuracy that can be reaped, the efforts to overcome these limitations continue with measured success. The carrier-phase measurement can be represented by the equation:

$$\phi = \psi + N_\phi + \beta_\phi + \nu_\phi \tag{11.2.2}$$

where

ψ = noiseless pseudorange consisting of geometric range and range error due to receiver timing error

N_ϕ = range uncertainty sometimes called integer cycle ambiguity (see Section 11.9)

β_ϕ = time-correlated errors associated with carrier phase

ν_ϕ = carrier-phase measurement noise

When pseudorange and delta range (or integrated Doppler) data are used in a combined setting, it may be presumed that the measurement noises for both types of data can be regarded as independent of each other. In the receiver, different processes are involved in obtaining the two types of measurements (see

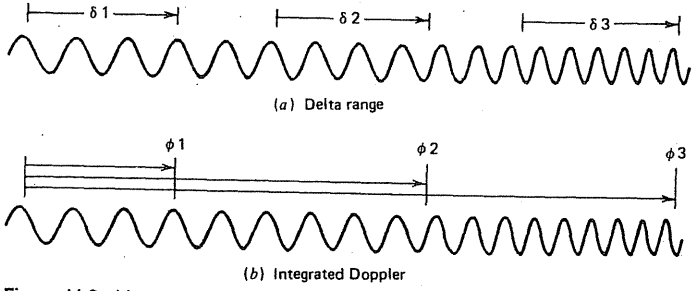


Figure 11.3 Measurements formed from carrier-phase data.

Fig. 11.2). The cross-feed from the carrier to the code-tracking loops can largely be ignored because the carrier data are virtually noiseless when compared to the pseudorange measurements made in the code loop.

11.3 GPS ERROR MODELS

The accuracy of a GPS position solution is dictated by errors in the observables described. The error components are listed in Table 11.1 with their approximate statistical characteristics.

Selective Availability

The intentional timing distortion applied to the GPS signal for civil users to reduce its ranging accuracy is known as *selective availability* or SA for short. This distortion appears as a random process to SPS receivers that do not have full access to removing it. Many studies have been made on the statistical time-correlation structure of this random process that can be adequately approximated by a second-order Gauss–Markov process (13). Figure 11.4 shows SA variations over a 100-min interval for four satellites. The process for the different satellites appears to be uncorrelated.

Examples on the modeling of SA were given earlier in Chapters 5 and 6 (see Problem 5.4 and Example 6.1). If the second-order Gauss–Markov model requires two states for each satellite, then for minimal navigation function, at least eight states are needed just to account for SA. More than 20 states are needed if all visible satellites are to be included!

Table 11.1 Correlated Errors Affecting Pseudorange: Approximate Statistical Parameters for a Stationary Observer (10, 11, 12)

Error Component	Standard Deviation*	Time Constant	Factors Causing Unpredictability
Satellite broadcast parameters	3, 30 m	>1 h	GPS Control Segment
Selective availability	0, 30 m	~2 min	GPS Control Segment
Iono refraction with correction	1, 5 m	>1 h	Sunspot cycle; scintillation activity
Tropo refraction with correction	2, 2 m	>1 h	Local atmospheric conditions; altitude
Code multipath	1, 5 m	$\frac{1}{2}$ to 10 min	Local scattering conditions

*The two numbers given are for the PPS (authorized) and SPS (unauthorized) operational modes, respectively.

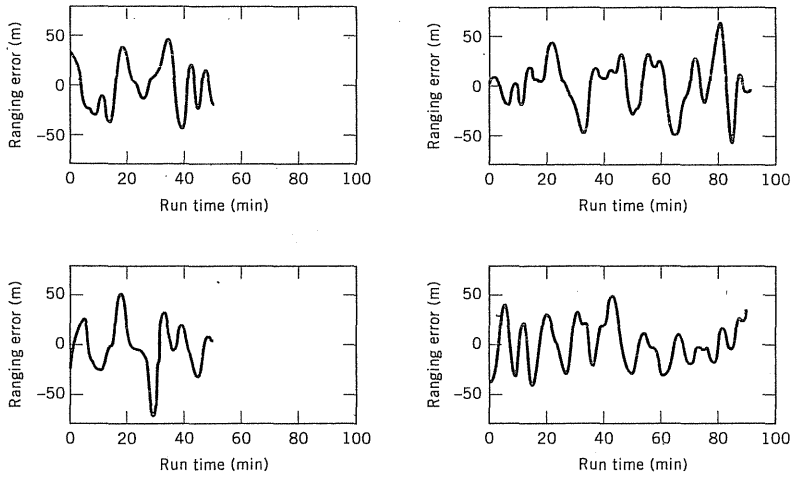


Figure 11.4 Examples of SA dither for four different satellites over a 100-min interval (25).

Satellite and Atmospheric Errors

During receiver operation, several error components are minimized using parameters that are broadcast in the 50 bits per sec (bps) navigation message. These error components include satellite position, satellite clock, and ionospheric refraction errors. The compensating parameters, being established from estimates made by the ground-tracking network of the Control Segment, are imperfect and contain errors at levels that are approximately represented in Table 11.1. Tropospheric refraction error, being dependent on local atmospheric conditions, is not compensated by any satellite broadcast parameter. Rather, the tropospheric errors are compensated by user-defined models that typically depend on inputs of altitude and satellite elevation angle, and for more complex models, temperature, and humidity.

The time-correlation characteristics of all of the aforementioned components are difficult to determine with precision but, in general, they change rather slowly. To choose an appropriate random process to model these errors, the first key characteristic to note is that these errors are all bounded over time. Hence, we shall consider the Gauss–Markov process to be a suitable candidate. Certainly, the second-order Gauss–Markov process used for the SA dither described previously can also be used here, after modification to reduce the steady-state variance and lengthening the correlation time constant. However, because of the smaller steady-state variance and the longer correlation time, the second-order rate-type state is practically negligible. Therefore, a first-order Gauss–Markov process for each component should suffice.

Multipath

Multipath phenomena refer to the distortion of a directly received GPS signal by its spurious replica that took an indirect path by way of reflecting off one or more objects. Clearly, the indirect path taken by the replica or multipath signal will be longer than that taken by the direct signal. This signifies ranging error that can be quite sizable. As long as the receiver is locked onto the direct signal by virtue of its stronger power, the multipath signal with its erroneous ranging information will only appear to distort the direct signal and perturb its phase to introduce a small ranging error. If the multipath signal is stronger than the direct signal, particularly when the latter is completely obstructed, the multipath error that results can well be far more significant.

Without getting into the intricate details of the mechanism involved in the interference of the direct and multipath signals, it suffices to say that the resulting error is generally oscillatory in nature as shown in Fig. 11.5. The reader is referred to (14) for more details on this issue. The period and amplitude of this oscillation generally vary from one situation to another, depending on the interaction of the satellite signal with the reflective objects surrounding the receiving antenna. In a situation involving moderate incidence of multipath, Fig. 11.5 shows the variable nature of the code-tracking error over a 10-min period (the offset shown in the plot is arbitrary).

Receiver Clock Modeling

The GPS receiver clock introduces a timing error that translates into ranging error that affects measurements made to all satellites (Eq. 11.1.1). This error is generally time-varying. If the satellite measurements are made simultaneously, this receiver clock error is the same on all measurements. Due to this commonality, the clock error has no effect on positioning accuracy if there are enough satellites to solve for it and so is not included as a source of positioning error.

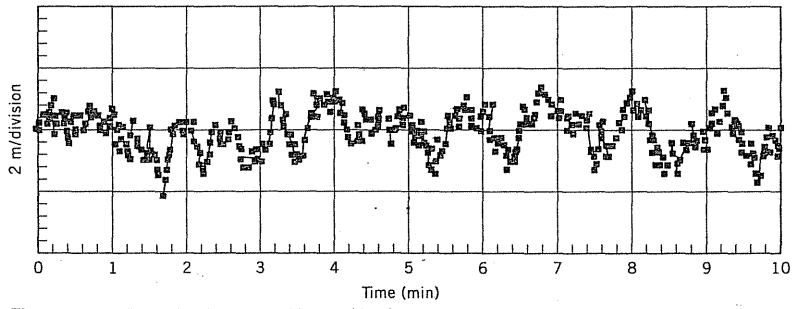


Figure 11.5 Example of code-tracking multipath.

Even so, there are advantages to properly modeling the receiver clock. This is illustrated in an example given at the end of this section.

A suitable clock model that makes good sense intuitively is a 2-state random-process model. It simply says that we expect both the oscillator frequency and phase to random walk over reasonable spans of time. We now wish to look at how the **Q** parameters of the state model may be determined from the more conventional Allan variance parameters that are often used to describe clock drift. We begin by writing the expressions for the variances and covariances for the general 2-state model shown in Fig. 11.6.

The clock states x_p and x_f represent the clock phase and frequency error, respectively. Let the elapsed time since initiating the white-noise inputs be Δt. Then, using the methods given in Chapter 3, we have

$$E[x_p^2(\Delta t)] = \int_0^{\Delta t}\int_0^{\Delta t} 1 \cdot 1 \cdot S_f \cdot \delta(u - v)\, du\, dv + \int_0^{\Delta t}\int_0^{\Delta t} u \cdot v \cdot S_g \cdot \delta(u - v)\, du\, dv = S_f \Delta t + \frac{S_g \Delta t^3}{3} \tag{11.3.1}$$

$$E[x_f^2(\Delta t)] = \int_0^{\Delta t}\int_0^{\Delta t} 1 \cdot 1 \cdot S_g \cdot \delta(u - v)\, du\, dv = S_g \Delta t \tag{11.3.2}$$

$$E[x_p(\Delta t)x_f(\Delta t)] = \int_0^{\Delta t}\int_0^{\Delta t} 1 \cdot v \cdot S_g \cdot \delta(u - v)\, du\, dv = \frac{S_g \Delta t^2}{2} \tag{11.3.3}$$

Now let us concentrate on the variance of state x_p. In particular, let us form the rms value of x_p time-averaged over the elapsed time Δt.

$$\text{Avg. rms } x_p = \left\{\sqrt{S_f \Delta t + \frac{S_g \Delta t^3}{3}}\right\} (\Delta t)^{-1} = \sqrt{\frac{S_f}{\Delta t} + \frac{S_g \Delta t}{3}} \tag{11.3.4}$$

Drift characteristics of real clocks have been studied extensively over the past few decades (15). Figure 11.7 shows a timing stability plot for a typical

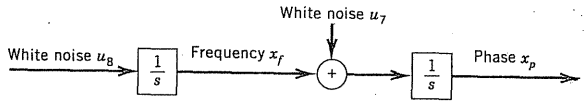


Figure 11.6 General 2-state model describing clock errors. The independent white-noise inputs y_7 and u_8 have spectral amplitudes of S_f and S_g.

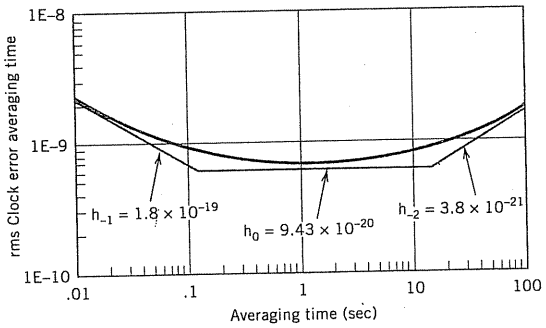


Figure 11.7 Allan variance plot with asymptotes for a typical crystal oscillator.

crystal clock (16). Such plots are known as *Allan variance* plots that depict the amount of rms drift that occurs over a specified period Δt, normalized by Δt. Note that over the time interval range shown in Fig. 11.7, there are three distinct asymptotic segments, the middle one of which is flat and associated with what is known as flicker noise. This segment, however, is missing from the response of the 2-state model described by Eq. (11.3.4).

There was good reason for this omission in the 2-state model. Flicker noise gives rise to a term in the variance expression that is of the order of Δt^2, and it is impossible to model this term exactly with a finite-order state model (17). To resolve this modeling dilemma, an approximate solution is to simply elevate the theoretical V of the 2-state model so as to obtain a better match in the flicker floor region.* This leads to a compromise model that exhibits somewhat higher drift than the true experimental values for both small and large averaging times. The amount of elevation of the V depends on the width of the particular flicker floor in question, so this calls for a certain amount of engineering judgment.

By completely ignoring the flicker term, we then compare terms of similar order in Δt in Eq. (11.3.1) and the q_{11} term in Eq. (60) of ref. 16 that is repeated in the footnote for convenience. This leads to the correspondence

* The approximate solution given here is simpler and less rigorous than the one given in (16). A word of caution is in order, though, about using the model given in (16). There are mistakes in the **Q** matrix given by Eq. (60) in the reference. The correct expressions for the q_{12} and q_{22} terms are

$$q_{12} = q_{21} = h_{-1}\Delta t + \pi^2 h_{-2}\Delta t^2$$
$$q_{22} = \frac{h_0}{2\Delta t} + 4h_{-1} + \frac{8}{3}\pi^2 h_{-2}\Delta t$$

The q_{11} term is correct in the reference and is repeated here for convenience:

$$q_{11} = \frac{h_0}{2}\Delta t + 2h_{-1}\Delta t^2 + \frac{2}{3}\pi^2 h_{-2}\Delta t^3$$

$$S_f \sim h_0/2$$
$$S_g \sim 2\pi^2 h_{-2} \qquad (11.3.5)$$

If an elevation of the V asymptotes is desired (see Problem 11.1 for one such method), the values of S_f and S_g will consequently be larger than those indicated in Eq. (11.3.5).

Precision clocks can have widely diverse Allan variance characteristics. Thus, one must treat each case individually and work out an approximate model that fits the application at hand. Table 11.2 gives typical values of h_0, h_{-1}, and h_{-2} for various types of timing standards widely used in GPS receivers. Note that the numbers given in Table 11.2 correspond to clock error in units of seconds. When used with clock error in units of meters, the values in Table 11.2 must be multiplied by the square of the speed of light $(3 \times 10^8)^2$.

EXAMPLE 11.1

Time-Transfer Error Model One of the many applications of GPS in routine use today is time transfer. This is the process of using a GPS receiver to derive accurate time relative to GPS system time, which is by itself a very precise standard. Recall from Eq. (11.1.1) that the receiver clock error is one of the variables, apart from the position components, that can be computed from the GPS ranging measurements. This clock error establishes the relationship between the receiver's time and the GPS system time. In the ideal time-transfer problem, it is assumed that the receiver's position is known from prior surveys. Thus, the measurement equation is strictly a direct connection between the pseudorange observable and the range equivalent of the clock error plus other errors.

$$\rho_1 - \hat{\rho}_1(\mathbf{x}) = \underbrace{c\Delta t}_{\chi} + \lambda_{\rho 1} + \sigma_{\rho 1} + \nu_{\rho 1} \qquad (11.3.6)$$

where

$\mathbf{x}$ = true location of observer
χ = range equivalent of receiver clock error Δt
$\lambda_{\rho 1}$ = long-term correlated errors
$\sigma_{\rho 1}$ = selective availability dither error
$\nu_{\rho 1}$ = pseudorange tracking noise

Table 11.2 Typical Power Spectral Density Coefficients for Various Timing Standards

Timing Standard	h_0	h_{-1}	h_{-2}
Compensated* crystal	$2(10^{-19})$	$7(10^{-21})$	$2(10^{-20})$
Ovenized crystal	$8(10^{-20})$	$2(10^{-21})$	$4(10^{-23})$
Rubidium	$2(10^{-20})$	$7(10^{-24})$	$4(10^{-29})$

*Compensation is for temperature variations.

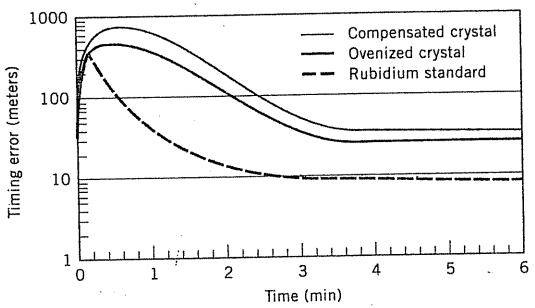


Figure 11.8 Time error profiles showing convergence and steady-state characteristics of time-transfer models for crystal and rubidium standards.

The main problem here is to estimate the desired parameter χ. Clearly, a single measurement is insufficient to separate the value of χ from the contribution of the other three error components. Instead, we can use a Kalman filter to estimate the various components separately using observations made over time. The degree of success to which this can be done depends on how different the components are spectrally. Of course, the long-term correlated error component (λ_ρ) with a time constant on the order of tens of minutes is sufficiently different spectrally than the selective availability dither (σ_ρ) time constant that is about several minutes. Also, the pseudorange tracking noise (ν_ρ) is uncorrelated between samples in time. What remains to determine the Kalman filter's ability to obtain a good estimate of χ is dependent on the stability of χ itself. In other words, the steady-state error variance of the χ estimate is influenced by how much uncertainty there is about the true value of χ as it randomly varies over a given period of time.

This can be illustrated with a covariance analysis performed using the very error model described above. Figure 11.8 shows the error standard deviation of the χ estimate for different clocks. What this means is that time transfer is more accurate with increasing clock stability. (See MATLAB M-file ex11_1.m for the details of this simulation.) ■

11.4 GPS DYNAMIC ERROR MODELS USING INERTIALLY DERIVED REFERENCE TRAJECTORY

The reference trajectory approach to integrating information derived from different sensors has been successfully used for many types of navigation systems (see Chapter 10). Positioning information from a GPS sensor can similarly be handled with this approach using the complementary filter methodology that was

described in Chapter 10 in the context of error modeling. We shall investigate, in this section, the derivation of the reference trajectory from an external inertial source.

The Inertial Navigation System (INS) is a traditional source for the reference trajectory in many integrated navigation systems. We can build upon the core 9-state process model introduced in Section 10.2—the model is described by Eqs. (10.2.9) and (10.2.10). In addition to these INS error states, there are GPS error states that may have to be accounted for, depending on the measurement model used. At the very least, two additional states representing the receiver clock error associated with the GPS pseudorange measurement are needed.*

EXAMPLE 11.2

The reference trajectory is nine-dimensional, consisting of three states each of position, velocity, and attitude obtained from the inertial measurements. In the complementary arrangement, the Kalman filter's role is to estimate the deviation of the reference from the truth trajectory. This is then the established state space. Hence, it is important to note that the state model for the Kalman filter here consists of inertial *error* quantities, rather than "total" dynamical quantities as in the stand-alone GPS case. Correspondingly, the random-process model should reflect the random character of the errors in the inertial sensor.

For the inertial error model, the chosen navigation reference frame will follow the locally level convention established in Section 10.1, where x points east, y points north, and z is the altitude above the reference WGS-84 ellipsoid. The error models are represented by block diagrams in Figs. 10.3, 10.4, and 10.5 showing separate channels for the east, north, and altitude. These can be regarded, in part, as generic models that are representative of a typical north-oriented locally level platform INS subject to modest dynamics. For the GPS error model, the receiver clock model is simply the same one as described in Section 11.3. Together, the new augmented process model in its differential equation form is given by

$$\begin{bmatrix} \dot{\mathbf{x}}_{1\text{-}9} \\ -- \\ \dot{x}_{10} \\ \dot{x}_{11} \end{bmatrix} = \left[\begin{array}{c|cc} \mathbf{F}_{\text{INS}} & \mathbf{0} & \\ -- & -- & -- \\ & 0 & 1 \\ \mathbf{0} & 0 & 0 \end{array}\right] \begin{bmatrix} \mathbf{x}_{1\text{-}9} \\ -- \\ x_{10} \\ x_{11} \end{bmatrix} + \begin{bmatrix} \mathbf{u}_{1\text{-}9} \\ -- \\ u_{10} \\ u_{11} \end{bmatrix}_k \tag{11.4.1}$$

where $\mathbf{F}_{\text{INS}}$ is obtained from Eq. (10.2.10).

To review, the state variables are ordered as follows:

* The clock model may be dispensed with if the pseudorange measurements from different satellites made simultaneously are processed in pairs. When two such pseudoranges are differenced, the clock error component is totally eliminated. Such a modification does not affect positioning accuracy, but it also denies any observability of the clock error (see Example 11.4 in Section 11.7).

x_1 = east position error

x_2 = east velocity error

x_3 = platform tilt about north (y) axis

x_4 = north position error

x_5 = north velocity error

x_6 = platform tilt about east (x) axis (11.4.2)

x_7 = vertical position error

x_8 = vertical velocity error

x_9 = platform azimuth error

x_{10} = range bias error due to GPS receiver clock

x_{11} = range rate error due to GPS receiver clock

The full state transition matrix corresponding to the differential equation of Eq. (11.4.1) is given by

$$\Phi = \begin{bmatrix} 1 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -g\Delta t & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{\Delta t}{R} & 1 & 0 & 0 & 0 & 0 & 0 & \omega_x \Delta t & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -g\Delta t & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{\Delta t}{R} & 1 & 0 & 0 & \omega_y \Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta t & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{11.4.3}$$

The measurement vector associated with the Kalman filter will consist of GPS pseudorange measurements. At least four pseudorange measurements are needed to sustain a stable solution, but there may certainly be more. A common supplementary measurement used in almost all standard GPS receivers is the delta range that is an approximate measure of velocity obtained from the Doppler frequency. Under normal tracking conditions, a delta range measurement is available for each satellite being tracked. Therefore, the measurement vector is usually augmented by a number of delta range measurements equal to the pseudorange measurements. The measurement matrix $\mathbf{H}$ comprises the usual

components of the unit direction vectors for the corresponding satellites. The complete measurement model is given as follows*:

$$\underbrace{\begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_n \\ \delta_1 \\ \delta_2 \\ \vdots \\ \delta_n \end{bmatrix}_k - \begin{bmatrix} \hat{\rho}_1 \\ \hat{\rho}_2 \\ \vdots \\ \hat{\rho}_n \\ \hat{\delta}_1 \\ \hat{\delta}_2 \\ \vdots \\ \hat{\delta}_n \end{bmatrix}_k}_{\mathbf{z}_k} = \underbrace{\begin{bmatrix} h_x^{(1)} & 0 & 0 & h_y^{(1)} & 0 & 0 & h_z^{(1)} & 0 & 0 & 1 & 0 \\ h_x^{(2)} & 0 & 0 & h_y^{(2)} & 0 & 0 & h_z^{(2)} & 0 & 0 & 1 & 0 \\ \vdots & & & \vdots & & & \vdots & & & \vdots & \\ h_x^{(n)} & 0 & 0 & h_y^{(n)} & 0 & 0 & h_z^{(n)} & 0 & 0 & 1 & 0 \\ 0 & h_x^{(1)} & 0 & 0 & h_y^{(1)} & 0 & 0 & h_z^{(1)} & 0 & 0 & 1 \\ 0 & h_x^{(2)} & 0 & 0 & h_y^{(2)} & 0 & 0 & h_z^{(2)} & 0 & 0 & 1 \\ \vdots & & & \vdots & & & \vdots & & & \vdots & \\ 0 & h_x^{(n)} & 0 & 0 & h_y^{(n)} & 0 & 0 & h_z^{(n)} & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{H}_k} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \\ x_9 \\ x_{10} \\ x_{11} \end{bmatrix} + \begin{bmatrix} v_{\rho 1} \\ v_{\rho 2} \\ \vdots \\ v_{\rho n} \\ v_{\delta 1} \\ v_{\delta 2} \\ \vdots \\ v_{\delta n} \end{bmatrix} \tag{11.4.4}$$

where

ρ_i = measured pseudorange

$\hat{\rho}(\mathbf{x}_0)$ = predicted pseudorange based on the nominal point of linearization $\mathbf{x}_0$

δ_i = measured delta range (in units of velocity)

$\hat{\delta}(\mathbf{x}_0)$ = predicted delta range based on the nominal point of linearization $\mathbf{x}_0$

$[h_x^{(i)}, h_y^{(i)}, h_z^{(i)},] = \left[\frac{\partial \psi_i}{\partial x} \quad \frac{\partial \psi_i}{\partial y} \quad \frac{\partial \psi_i}{\partial z}\right]^T$ for $i = 1, \ldots, n$; $n \geq 4$

$[x_1, x_2, \ldots, x_{11}]^T$ = state vector defined in Eq. (11.4.2)

$v_{\rho i}$ = pseudorange measurement noise sequence

$v_{\delta i}$ = delta range measurement noise sequence ▩

The model given in the example above is a simple scheme to provide aiding for a GPS navigation system with inertial data. For tutorial purposes, the example is quite adequate in illustrating the fundamentals of the system integration. Clearly, the complexity of the model, especially for the inertial part, can be raised significantly to improve the estimation accuracy. It should be pointed out that the vertical (altitude) channel used for the example model here is unstable when required to operate in the absence of GPS. However, without any other source of data available, the double integration of vertical acceleration is about the best we can do to obtain altitude. Unfortunately, this process leads to an

* This model assumes that the vehicle velocity is approximately constant over the delta range time interval. If the constant-velocity approximation is questionable (i.e., high dynamics), a more accurate model can be obtained with the delayed-state measurement model. (See Section 9.2 for details.)

altitude error that builds up exponentially. This arrangement is probably quite acceptable if the loss of GPS is brief, that is, on the order of a few minutes. In cases where the inertial portion has to operate by itself for extended durations of time, the addition of another sensor such as the barometric altimeter is necessary to bound the error growth. The barometric altitude might then be combined together with the inertial data in another complementary coupling such as the one suggested in Problem 4.13.

Since the contribution of measurement noise from the inertial unit is comparatively minuscule, the $2n \times 2n$ measurement noise covariance matrix $\mathbf{R}$ should account mostly for the GPS pseudorange and delta range noise terms.

$$\mathbf{R} = \begin{bmatrix} r_\rho & & & & & & \\ & r_\rho & & & \mathbf{0} & & \\ & & \ddots & & & & \\ & & & r_\rho & & & \\ & & & & r_\delta & & \\ & & \mathbf{0} & & & r_\delta & \\ & & & & & & \ddots \\ & & & & & & & r_\delta \end{bmatrix} \tag{11.4.5}$$

Equation (11.4.5) assumes uncorrelated measurement errors among satellites. Based on this assumption, the measurements may be processed one component at a time as scalar measurement data (see Section 6.3). Although it was noted in Section 11.2 that the variance of the pseudorange measurement white noise is rather small, it is often the case in practice that a slightly larger variance is chosen for r_ρ. A conservative choice of 15 m^2, for example, in the absence of selective availability, will do well to help offset the effects of other correlated measurement noise unaccounted for in the measurement models given here.

Cascaded Filters

A problem often encountered problem in GPS/INS integration should be brought to the reader's attention. It is not always that the designer or system integrator commands the luxury of having to solve problems without design constraints. When dealing with avionics, for instance, the modularity of the equipment might necessitate the treatment of a GPS receiver as an instrument sensor that outputs position and velocity data. In order to use this data to aid an INS, a system integrator is faced with using the output of some Kalman filter in the GPS receiver to feed into another integrated Kalman filter that blends the data from all sensors to calibrate the INS. This situation is commonly known as the *cascaded Kalman filters* problem where the output of the first filter, corrupted by a time-correlated estimation error, is fed into a second filter that "thinks" the noise corrupting the data is white. This problem has been studied somewhat extensively, sometimes under the heading of decentralized Kalman filters, chiefly because of its potential usefulness in large modular systems. A limited treatment

of this subject is given in Section 9.6. We will not pursue this further here, though.

11.5 STAND-ALONE GPS MODELS

In the absence of an inertial reference, the complementary filter that accommodates the use of the Kalman filter degenerates into a special nonoptimal case. Its use is quite common in many applications found today where GPS is the only sensor available. Such stand-alone applications must depend on an operating environment where the loss of GPS signals is rare. Examples of this include civil aviation, maritime, and farming applications.

In this situation, the complementary filter block diagram of Fig. 10.2 is redrawn to remove the replaced inertial sensor with a null block that serves as a virtual reference source (see Fig. 11.9). Clearly, this reference source depends entirely on the data provided by the GPS sensor so it presents no *new* dynamical information. The use of the reference trajectory in this case is still beneficial for linearization of the GPS measurement situation. This mode of operation will be recognized as an extended Kalman filter (see Section 9.1).

Dynamic State Process

Without an inertial sensor to provide a reference trajectory, the process dynamics of the position states do not represent random sensor errors but rather "total" observer dynamics. The accurate description of such dynamics in the Kalman filter process model may not be altogether straightforward, depending on the type of observer dynamics encountered in a given application.

In its most basic form, the state vector for a stand-alone GPS model should consist of three position states and two clock states. This 5-state filter is ideal for a stationary observer (i.e., constant position) where the random bias model might be appropriate for the position states, even though the random walk model is usually preferred. Recall from Chapter 5 that with $\mathbf{Q} = \mathbf{0}$ for a random bias,

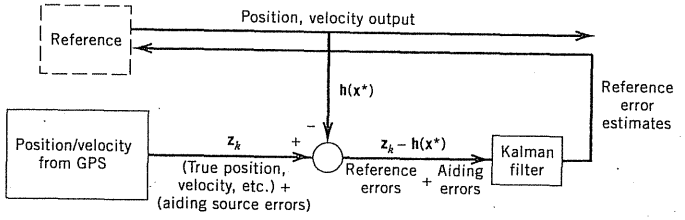


Figure 11.9 Stand-alone GPS as a special case of the feedback complementary filter of Fig. 10.2.

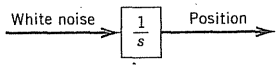


Figure 11.10 Random walk model for a stationary observer.

no process noise is added with each iteration of the filter, thereby causing the error covariance matrix $\mathbf{P}$ to eventually converge to zero. We have seen, in Section 6.6, how such a situation invites the prospect of divergence as a result of numerical problems when dealing with infinitesimally small numbers, especially after a prolonged duration of processing. With the random walk process, as shown in Fig. 11.10, $\mathbf{Q} \neq \mathbf{0}$. Note here that this dynamical model is also suitable for a near-stationary observer, such as a buoy drifting at sea. This will be known as the *Position* model. Note further that this can be a linearized or an extended Kalman filter model, with the linearization taking place about some nominal point.

Where the observer is not stationary but moving with nearly constant velocity, we have a case resembling that of the oscillator with a stable frequency error. Here, the random walk model is not ideally suitable because it makes the assumption that the driving white-noise input to the integrator is unbiased when, in fact, the velocity-induced bias would be nontrivial. A better model, in this instance, would be one corresponding to the double integrator transfer function diagram of Fig. 11.11 that we shall refer to as the *Position-Velocity* (PV) model. Here, the velocity is not white noise but a random walk process.

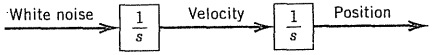


Figure 11.11 Integrated random walk model for a dynamic observer.

EXAMPLE 11.3

In the PV model, each spatial dimension will have two degrees of freedom, one of position and the other of velocity. Therefore, for the GPS problem where there are three spatial dimensions and one time dimension, the state vector now becomes an 8-tuple. The PV dynamic process can be described by the following vector differential equation:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \\ \dot{x}_5 \\ \dot{x}_6 \\ \dot{x}_7 \\ \dot{x}_8 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} + \begin{bmatrix} 0 \\ u_2 \\ 0 \\ u_4 \\ 0 \\ u_6 \\ u_7 \\ u_8 \end{bmatrix} \tag{11.5.1}$$

where

x_1 = east position
x_2 = east velocity
x_3 = north position
x_4 = north velocity
x_5 = altitude
x_6 = altitude rate
x_7 = range (clock) bias error
x_8 = range (clock) drift error

and the spectral amplitudes associated with the white-noise driving functions are S_p for u_2, u_4, and u_6, S_f for u_7, and S_g for u_8 (see Fig. 11.6).

From Eq. (11.5.1), the state transition matrix can be derived as

$$\Phi = \begin{bmatrix} 1 & \Delta t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & \Delta t & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & \Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{11.5.2}$$

To obtain the process noise covariance matrix $\mathbf{Q}$, we resort to the methods given in Chapter 3. Note from Eq. (11.5.1) that we can treat the three position-velocity state variable pairs independently.

$$E[x_i^2(\Delta t)] = \int_0^{\Delta t} \int_0^{\Delta t} u \cdot v \cdot S_p \cdot \delta(u - v)\, du\, dv = \frac{S_p \Delta t^3}{3} \tag{11.5.3}$$

$$E[x_{i+1}^2(\Delta t)] = \int_0^{\Delta t} \int_0^{\Delta t} 1 \cdot 1 \cdot S_p \cdot \delta(u - v)\, du\, dv = S_p \Delta t \tag{11.5.4}$$

$$E[x_i(\Delta t) x_{i+1}(\Delta t)] = \int_0^{\Delta t} \int_0^{\Delta t} 1 \cdot v \cdot S_p \cdot \delta(u - v)\, du\, dv = \frac{S_p \Delta t^2}{2} \tag{11.5.5}$$

for $i = 1$, 3, and 5.

The equations involving the clock states x_7 and x_8 were derived in Eqs. (11.3.1) through (11.3.3). With these, the process noise covariance matrix is as follows:

$$\mathbf{Q} = \begin{bmatrix} S_p\frac{\Delta t^3}{3} & S_p\frac{\Delta t^2}{2} & 0 & 0 & 0 & 0 & 0 & 0 \\ S_p\frac{\Delta t^2}{2} & S_p\Delta t & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & S_p\frac{\Delta t^3}{3} & S_p\frac{\Delta t^2}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & S_p\frac{\Delta t^2}{2} & S_p\Delta t & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & S_p\frac{\Delta t^3}{3} & S_p\frac{\Delta t^2}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & S_p\frac{\Delta t^2}{2} & S_p\Delta t & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & S_f\Delta t + S_g\frac{\Delta t^3}{2} & S_g\frac{\Delta t^2}{2} \\ 0 & 0 & 0 & 0 & 0 & 0 & S_g\frac{\Delta t^2}{2} & S_g\Delta t \end{bmatrix} \tag{11.5.6}$$

The corresponding measurement model takes on a form very similar to the one used in the INS/GPS example in Section 11.4. With slight modification, Eq. (11.4.4) becomes

$$\underbrace{\begin{bmatrix} \rho_1 \\ \rho_2 \\ \vdots \\ \rho_n \\ \delta_1 \\ \delta_2 \\ \vdots \\ \delta_n \end{bmatrix}_k - \begin{bmatrix} \hat{\rho}_1 \\ \hat{\rho}_2 \\ \vdots \\ \hat{\rho}_n \\ \hat{\delta}_1 \\ \hat{\delta}_2 \\ \vdots \\ \hat{\delta}_n \end{bmatrix}_k}_{\mathbf{z}_k} = \underbrace{\begin{bmatrix} h_x^{(1)} & 0 & 0 & h_y^{(1)} & 0 & 0 & h_z^{(1)} & 0 & 0 & 1 & 0 \\ h_x^{(2)} & 0 & 0 & h_y^{(2)} & 0 & 0 & h_z^{(2)} & 0 & 0 & 1 & 0 \\ \vdots & & & \vdots & & & \vdots & & & \vdots & \\ h_x^{(3)} & 0 & 0 & h_y^{(3)} & 0 & 0 & h_z^{(3)} & 0 & 0 & 1 & 0 \\ 0 & h_x^{(1)} & 0 & 0 & h_y^{(1)} & 0 & 0 & h_z^{(1)} & 0 & 0 & 1 \\ 0 & h_x^{(2)} & 0 & 0 & h_y^{(2)} & 0 & 0 & h_z^{(2)} & 0 & 0 & 1 \\ \vdots & & & \vdots & & & \vdots & & & \vdots & \\ 0 & h_x^{(3)} & 0 & 0 & h_y^{(3)} & 0 & 0 & h_z^{(3)} & 0 & 0 & 1 \end{bmatrix}_k}_{\mathbf{H}_k} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} + \begin{bmatrix} v_{\rho 1} \\ v_{\rho 2} \\ \vdots \\ v_{\rho n} \\ v_{\delta 1} \\ v_{\delta 2} \\ \vdots \\ v_{\delta n} \end{bmatrix} \tag{11.5.7}$$

The determination of the spectral amplitude S_p for the position random process is at best a "guesstimate" based roughly on expected vehicle dynamics. The PV model also becomes inadequate for cases where the near-constant velocity assumption does not hold, that is, in the presence of severe accelerations. To accommodate acceleration in the process model, it is appropriate to add another degree of freedom for each position state. Although we can easily add one more integrator to obtain a *Position-Velocity-Acceleration* (PVA) model, a stationary process such as the Gauss–Markov process is perhaps more appropriate than the nonstationary random walk for acceleration (Fig. 11.12). This goes in accordance with real-life physical situations where vehicular acceleration is usually brief and seldom sustained. The state vector for this PVA model then becomes 11-dimensional with the addition of three more acceleration states.

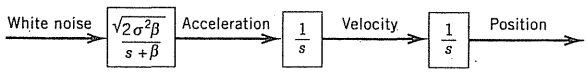


Figure 11.12 Position-velocity-acceleration model for high (accelerative) dynamics observer.

Derivation of the corresponding state transition and the process noise covariance matrices will be left as part of an exercise in Problem 11.3. It should be pointed out here that although acceleration is being accounted for in some stochastic form, the exercise is an approximation at best. There is no exact way to fit deterministic accelerations into any of the random-process dynamics suitable for a Kalman filter model. With the PVA model, the Kalman filter will do a better job of estimation than the PV model, but it may still be inadequate by other measures of optimality.

Carrier (Aided) Smoothing

The stand-alone models described previously depend on pseudorange measurements to obtain positioning information. The individual pseudorange measurement is relatively noisy, so improved position accuracy must be achieved by filtering. Without an external inertial reference, the ability of these models to filter out the pseudorange measurement noise while estimating the position state variables invariably depends on the level of dynamics encountered by the observer. The degree of dynamical uncertainty must be properly reflected in the size of the white-noise spectral amplitude S_p, which in turn dictates the size of the parameters in the $\mathbf{Q}$ matrix. The larger the $\mathbf{Q}$ parameters are, the lower the overall estimation accuracy becomes. If observer dynamics are known *a priori,* then this information may be exploited. For example, a stationary observer in effect possesses dynamics that are known perfectly—there is no uncertainty in its dynamics! An aircraft in straight and level flight has some uncertainty though it is small. Freely maneuvering crafts in any medium generally have high levels of dynamic uncertainty.

The carrier-phase measurement described in Section 11.2 contains high-fidelity information about relative changes in the satellite-to-observer range, which implicitly also accounts for the dynamics of the observer. Since this type of measurement cannot readily yield positioning information because of an initial range uncertainty that is associated with it (see Eq. 11.2.2), but paints a very accurate picture of position changes over time, its characteristics closely conform to those of inertially derived position data. To properly utilize it, we can treat it just like inertial data and resort once again to the complementary filter philosophy of data integration.

In the complementary filter scheme shown in Fig. 11.13, the carrier-phase measurement representing the reference information is subtracted from the pseudorange measurement and the residual is fed to a Kalman filter. With the observer's dynamics totally extracted, the residual data then appear to the Kalman filter as if the modified "measurement" situation were a stationary one. The

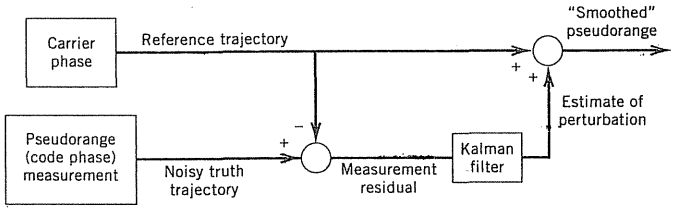


Figure 11.13 Complementary Kalman filter combining continuous carrier-phase and pseudorange data from GPS signal measurements.

Kalman filter can then afford to lengthen its averaging time constant to significantly reduce the large pseudorange measurement noise. The filter output when recombined with the reference trajectory data provides a final position solution that is highly responsive to any kind of dynamical motion and also has a small estimation error. This is the complementary facet of this form of data integration.

$$\rho - \phi = -N_\phi + (\beta_\rho - \beta_\phi) + (\nu_\rho - \nu_\phi) \tag{11.5.8}$$

If $\beta_\rho = \delta + \iota + \tau$ and $\beta_\phi = \delta - \iota + \tau$,

where

δ = satellite broadcast and timing error (including SA)
ι = ionospheric refraction
τ = tropospheric refraction

then $\beta_\rho - \beta_\phi = 2\iota$.

Equation (11.5.8) then becomes $\rho - \phi \approx \underbrace{-N_\phi + 2\iota}_{\eta} + \nu_\rho$ (11.5.9)

On the right-hand side of Eq. (11.5.9), the principal parameter of interest is $\eta = -N_\phi + 2\iota$. The ionospheric parameter ι could not be eliminated in Eqs. (11.5.8) and (11.5.9) because it shows up in the code-phase and carrier-phase equations with opposite signs. The code-tracking error is at least an order of magnitude larger than its carrier-tracking counterpart, so the latter is not considered to be significant for Eq. (11.5.9). The term ν_ρ represents the code-tracking error that, in general, may contain a time-correlated multipath. Filtering is needed to reduce the code-tracking noise parameter ν_ρ down to an estimation error ε. The appropriate model for the term η is a random walk process to reflect time variation in the ionospheric parameter ι. The Kalman filter outputs an estimate of η, which is then recombined with the reference phase ϕ to give what amounts to a "smoothed" pseudorange:

$$\underbrace{-N_\phi + 2\iota + \varepsilon}_{\text{Kalman filter estimate}} + \underbrace{\psi + N_\phi + \beta_\phi + \nu_\phi}_{\text{Reference carrier phase}} = \psi + \beta_\rho + \varepsilon + \nu_\phi = \hat{\rho}$$

since

$$\beta_\rho = \beta_\phi + 2\iota.$$

Therefore, the process of carrier smoothing reduces the noise level corrupting the pseudorange measurement (Eq. 11.2.1) from ν_ρ to $(\nu_\phi + \varepsilon)$, where ε is the estimation error from the carrier smoothing. It is worth mentioning that complementary filtering does not mitigate the SA error. It is still present in the β_ρ term after the filtering.

An efficient way to implement the carrier-smoothing complementary filter is as a parallel bank of individual filters, each one associated with a different satellite (see Figure 11.14 below) that is considered to have an independent set of error processes. The filter outputs can be combined by a least-squares filter or a simple Kalman filter that behaves nearly like one.

Before moving on, it should be pointed out that one of the main difficulties usually associated with the use of the continuous carrier phase has been the issue of cycle slips. Recall that the continuous carrier phase is a piece of data that is referenced to a known initial point, from which incremental changes in phase are integrated so that the exact number of whole cycles accumulated, although unobservable at any instant in time, is properly accounted for. If the signal is weak, or worse altogether lost, the integration process becomes prone to error, which affects most critically the integer portion of the carrier-phase count. This pathological situation is called cycle slip. There are various methods to detect cycle slips and we refer the interested reader to the suggested references for detailed information (8, 18).

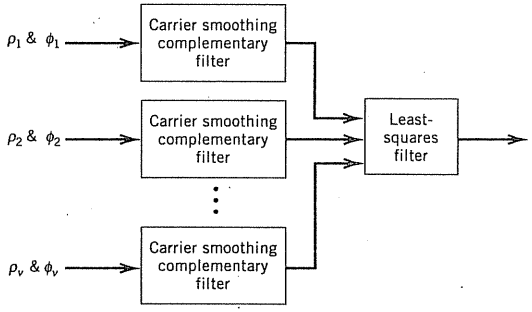


Figure 11.14 A parallel bank implementation for carrier smoothing.

11.6 EFFECTS OF SATELLITE GEOMETRY

One of the more prominent characteristics of GPS is the nongeostationary nature of its satellite orbits. This has the effect of making the matrix $\mathbf{H}$ time-varying.

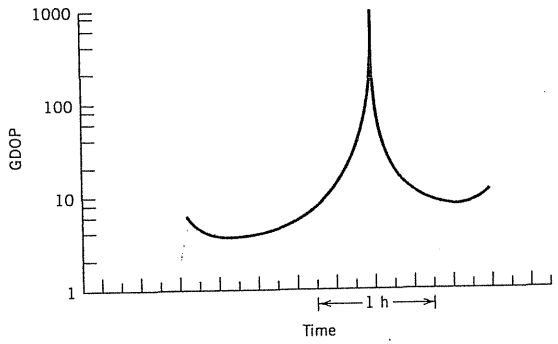


Figure 11.15 A typical incidence of a "GDOP chimney."

As a consequence, the observability of the system also varies with time. The instantaneous measure of the degree of observability commonly used, called the geometric dilution of precision* (GDOP), is literally a "snapshot" rms error for the least-squares position and time solution caused by 1-m rms of error in the measurements.

For a given set of satellites, there may be times when the GDOP condition becomes singular. A typical plot for such an instance can be seen in Fig. 11.15, a condition sometimes figuratively called the "GDOP chimney." GDOP chimneys are generally avoidable by changing satellite combinations if enough satellites are available to do so. However, we now concern ourselves with the case where a GDOP chimney is unavoidable, and look at its effect on a Kalman filter solution.

Within the vicinity of the GDOP chimney, the high GDOP condition causes a deterioration in position solution accuracy. As an example, the error for an instantaneous (unfiltered) solution when the GDOP is 50, assuming a measurement accuracy of 10 m, would then be an intolerable 500 m (19). Fortunately, some relief can be obtained with filtering by relying partially on data from outside the chimney window that are at lower GDOP levels. In this regard, the Kalman filter can be considered an *information manager* that weights each piece of data it uses according to the observability conditions at the time of acquisition. In other words, the information collected during the GDOP chimney is given proportionately less weight by the filter than the information collected before the onset of the poor geometric condition.

At the other end of the balance, if the GDOP were constant throughout, the accuracy of the information collected, with regard to the random process it serves to estimate, also degrades with time. When the two opposing effects of GDOP and the aging of data are put together, the compromise results in situations depicted by the plots of solution error shown in Fig. 11.16. In this comparison, the three cases shown differ in the relative spectral amplitudes of their process noise. This reflects the dependence of the maximal peak of position error (near the point of GDOP singularity) on the anticipated state dynamics that are modeled in the Kalman filter. The more stable (i.e., more predictable) the observer's vehicle and clock dynamics are, the lesser the effect of the GDOP chimney on position accuracy will be felt. On the other hand, under high dynamical stress, the random process with a larger process noise amplitude will cause the filter to "forget" past information faster, and it has to depend on more recent data collected, thereby reacting with a response that follows closer to the actual GDOP chimney. Note that the latter is really a limiting condition when the process noise gets infinitely large, and the filter becomes "memory-less" from one iteration to the next.

This property of the Kalman filter has been somewhat forgotten in GPS applications. It is easily overlooked when the observability conditions relating to satellite geometry are favorable. However, when we must deal with situations that use the satellite resources to the fullest, particularly problems that require redundant data from more than four satellites, favorable satellite geometries are not always assured. This is where optimal estimation with Kalman filtering becomes indispensable.

* For the deterministic measurement equation $\mathbf{z} = \mathbf{H}\mathbf{x}$,

$$\text{GDOP} = \sqrt{\text{trace}\,(\mathbf{H}^T\mathbf{H})^{-1}} \quad \text{(see ref. 17)}$$

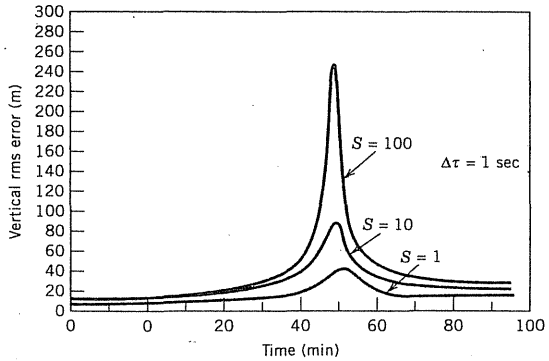


Figure 11.16 Comparison of the choice of process noise spectral amplitudes on the peak error during a GDOP chimney encounter.

11.7 DIFFERENTIAL AND KINEMATIC POSITIONING

Thus far, we have only been concerned with the use of a single GPS receiver for position determination. The use of a second receiver as a reference can greatly enhance the positioning accuracy of the first receiver when it can take advantage of the error information provided by the former. The accuracy improvement hinges on what is commonly known as the *differential* principle,

whereby many of the system errors (represented by β in Eqs. 11.2.1, 11.2.2, and 11.5.8) can be eliminated because they are common in the measurements made by the two receivers.

Differential Corrections

In real-time differential systems, it is both inefficient and unnecessary to pass on raw measurements from one receiver to the other. Instead, the system error associated with each satellite is made and a correction parameter issued accordingly. This saves information bandwidth in the data link between the two receivers and also allows for asynchronous processing of measurement data from the two receivers. Perhaps the most widely used differential correction standard today is one defined by the Radio Technical Commission on Maritime (RTCM) Special Committee 104 (RTCM-104 for short).

A Kalman filter can be effectively used to estimate the error in a reference station by treating the problem in much the same way as in our timing receiver example in Section 11.3. We now look back at Eq. (11.3.6) in a slightly different way:

$$\rho_1 - \hat{\rho}_1(\mathbf{x}) = \underbrace{c\Delta t}_{\chi} + \underbrace{\lambda_{\rho 1} + \sigma_{\rho 1}}_{\varepsilon} + \nu_{\rho 1} \tag{11.3.6}$$

Here, we shall consider the estimation of two states, χ and ε, the latter being essentially dominated in character by the selective availability dither component σ. In a real-time situation, the estimate of the correction is applied to the measurement of the second receiver usually at a slightly later time because of latencies encountered in the processing of the correction and the delays in the data link. Such a time delay in applying the correction to the measurements made in the second receiver amounts to a prediction problem when seen from the perspective of the correction estimation process itself. To handle this properly, the correction data received must be adjusted for the time delay by prediction involving the relevant random processes that govern the correction data for one particular satellite (see Eq. 6.1.1).

If the stability of the reference station receiver clock is high, such as for an atomic standard, the contribution of $\hat{\chi}$ may be insignificant over a reasonable length of time such that the prediction error will be dominated by $\hat{\varepsilon}$, which in turn mostly accounts for the selective availability dither σ. In that case, the prediction error will closely resemble that shown in Fig. 6.2 for the SA dither process. However, if the clock stability is of the quality found in most affordable GPS receivers, the contribution of $\hat{\chi}$ to the prediction error becomes important, particularly for long prediction intervals. This is because χ, as a clock error process (see Section 11.3), has unbounded variance. The presence of SA generally makes the handling of situations with long prediction intervals impractical.

The reader is referred to (13) for further details on the combined effects of a stationary SA dither process and a nonstationary clock process. We will not dwell further on this here.

Kinematic Positioning

Some innovative work in the field of terrestrial surveying over the past decade has led to the blossoming of a very exciting and productive application area over the past decade called *kinematic positioning* (20). Interferometric techniques developed in the past for radiating sources such as quasars were adapted for GPS using the pure tone of the carrier signal only and, in turn, reaping the benefits of its accuracy. The method is also based on differential principles where the solution obtained is only for the position of one observer with respect to another reference observer. The errors that corrupt the observations of two observers that are relatively close together are strongly correlated and can be removed in the differential GPS solution. Today, kinematic positioning has ventured beyond its terrestrial surveying roots into dynamic positioning applications, such as the precision landing of aircraft.

EXAMPLE 11.4

In static surveying where the object is to determine the baseline vector between two points, the observable consists of the difference between carrier-phase measurements made simultaneously at the two ends of the baseline. The one-dimensional situation depicted in Fig. 11.17 shows the relationship between the total phase delay and the baseline separation through the known geometry.

When the receivers make their initial measurements, the total phase delay is, of course, unobservable. What is measurable as the phase delay, however, is known to be in error by an integral number of cycles, a quantity called the *initial integer ambiguity*. Thus, this one-dimensional measurement equation contains two variables to be solved for (1) the baseline separation and (2) the initial integer ambiguity. The Kalman filter is certainly suited for handling estimation problems of such a nature.

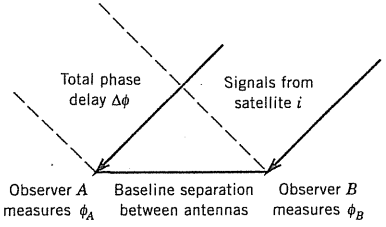


Figure 11.17 One-dimensional phase difference measurement model. Baseline separation must be small enough to maintain linearity assumptions.

In the GPS situation, where only the position-related quantities are of interest, we can eliminate the range- (or clock-) related biases by carrying out a second difference across satellites of the single difference measurements that are originally formed across the two stations (see Eqs. 11.7.1 and 11.7.2). It is assumed that the set of phase measurements used to form the double differences are made simultaneously in order to do this (20).

Single difference:

$$\Delta\phi^{(i)} = \phi_A{}^{(i)} - \phi_B{}^{(i)} \tag{11.7.1}$$

Double difference:

$$\nabla\Delta\phi^{(i)} = \Delta\phi^{(i)} - \Delta\phi^{(i+1)} \tag{11.7.2}$$

for $i = 1, 2, \ldots$, where ϕ_A is the phase measured by observer A and ϕ_B is the phase measured by observer B from satellite i.

The example used here is based on a model by Brown and Hwang (21) where the initial integer ambiguities, although integer in nature, are estimated as continuous random variables.

$$\begin{bmatrix} \nabla\Delta\phi^{(1)} \\ \nabla\Delta\phi^{(2)} \\ \nabla\Delta\phi^{(3)} \end{bmatrix}_k = \begin{bmatrix} \nabla h_x^{(1)} & \nabla h_y^{(1)} & \nabla h_z^{(1)} & 1 & 0 & 0 \\ \nabla h_x^{(2)} & \nabla h_y^{(2)} & \nabla h_z^{(2)} & 0 & 1 & 0 \\ \nabla h_x^{(3)} & \nabla h_y^{(3)} & \nabla h_z^{(3)} & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ \nabla N^{(1)} \\ \nabla N^{(2)} \\ \nabla N^{(3)} \end{bmatrix}_k + \begin{bmatrix} \nabla v^{(1)} \\ \nabla v^{(2)} \\ \nabla v^{(3)} \end{bmatrix}_k \tag{11.7.3}$$

where

$\nabla\Delta\phi^{(1)}, \nabla\Delta\phi^{(2)}, \nabla\Delta\phi^{(3)}$ = double difference measurement set
$\nabla h_x^{(i)}, \nabla h_y^{(i)}, \nabla h_z^{(i)}$ = ith unit direction vector differenced across satellites
$\Delta x, \Delta y, \Delta z$ = position vector
$\nabla N^{(1)}, \nabla N^{(2)}, \nabla N^{(3)}$ = initial integer ambiguity vector
$\nabla v^{(1)}, \nabla v^{(2)}, \nabla v^{(3)}$ = measurement noise vector

The process model is trivial since all elements of the state vector are random biases. Therefore, the $\mathbf{Q}$ matrix is zero. Here, we have an example of where it is safe to add no process noise only because this problem is confined to establishing the fixed values of the state vector and does not run indefinitely.

The measurement noise covariance matrix depends on the combination used for the double difference. For the combinations shown in Eq. (11.7.2), the covariance matrix for the differenced measurement noise vector $[\nabla v^{(1)}, \nabla v^{(2)}, \nabla v^{(3)}]^T$ is given by

$$\mathbf{R}_{\nabla\Delta\phi} = \begin{bmatrix} 2r_\phi & -r_\phi & 0 \\ -r_\phi & 2r_\phi & -r_\phi \\ 0 & -r_\phi & 2r_\phi \end{bmatrix}$$

■

In practice, r_ϕ is in the range of 1 to 5 mm for a stationary receiver, as is the case here with static surveying. Once again, as with r_ρ for a code-ranging measurement model, the choice of a value for r_ϕ may also depend greatly on time-correlated errors that are unaccounted for in this model. These time-correlated errors, which may originate from all the usual GPS error sources (see Table 11.1), arise in this situation only when the differential assumption, made when posing this problem, starts to break down.

It should be noted here that the simple formulation of Example 11.4 can be carried further and the accuracy refined by taking advantage of the certainty that the initial integer ambiguity states are, in fact, integers. In (21, 22), the Magill adaptive scheme (see Section 9.3) was used where a parallel bank of Kalman filters, each assuming an integer value for a hypothesis, processes the same double difference phase measurement sequence. Although cumbersome at first glance, the parallel filter viewpoint serves as an excellent conceptual starting point, whereupon many simplifications are available to make the problem a tractable one indeed, especially in a computational sense! This statistical scheme provides a convenient use of Kalman filtering to solve the multiple hypothesis testing problem formulated in an optimal way.

11.8 OTHER APPLICATIONS

The applications discussed in this chapter by no means cover the entire myriad of GPS problems that are or can be solved by Kalman filtering techniques. Attention here has mostly been devoted to the primary user issues of navigation, positioning, and timing. Even so, the coverage has not been comprehensive in trying to keep the chapter to a tutorial pace. For readers that are interested in delving further into other applications of Kalman filtering in the areas of GPS, several references are hereby suggested (35, 36).

The effects and modeling of ionospheric refraction error (23) and SA (12, 24, 25) are often among the foremost concerns for the nonauthorized user of the system. Kinematic positioning takes the carrier-phase methods of static surveying one step further by introducing motion to the problem (26). The result appears to be, in addition to more efficiently run survey operations, a promising extension of the high accuracies associated with carrier-phase surveying to real-time differential navigation (27). GPS has also found itself forming synergistic collaborations with other types of instrumentation (28, 29, 30). In the most natural of combinations, GPS has found an inertial navigation field in the midst of a technological surge in miniaturization activity in gyro and accelerometer sensors (31). The GPS/inertial combination is especially attractive because the

inertial system makes it possible to "coast" through short periods of poor satellite geometry or brief satellite obscurations.

Wide-area differential GPS has become a hotbed of activity over the past few years where its main interest lies in the estimation of errors over an extensive spatial domain. In addition to SA modeling, the estimation problem here includes orbital and satellite clock errors as well as ionospheric error. The processing of data collected by a network of ground-monitoring stations must provide near-real-time corrections to users of the system. Another form of ground monitoring under development concerns GPS signal integrity. This is an area where Kalman filtering could play a central role in the area of failure detection (33).

It would not be an overstatement to say that the usefulness of GPS has, by this point in time, far surpassed that originally envisioned by its early designers. Despite more than ten years of extensive development, it does not appear that the well of ideas for GPS applications is about to dry up quite yet. This optimism ensures a continued role for applied Kalman filtering to play in the wealth of GPS applications already in development as well as those that are yet untapped.

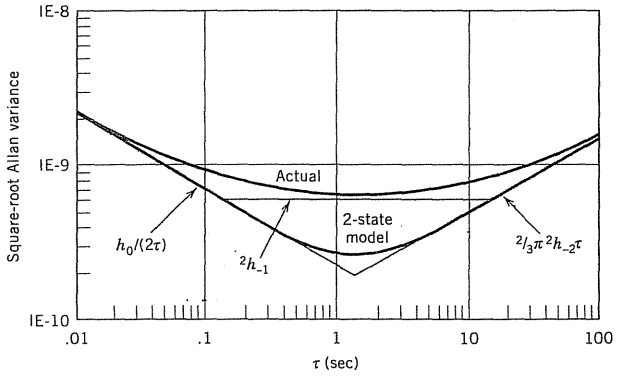
PROBLEMS

11.1 In Section 11.3 on receiver clock modeling, it was pointed out that a finite-order state model cannot adequately account for the flicker noise component. The relationships given by Eq. (11.3.5) were derived by simply ignoring the flicker noise term:

$$\text{Allan variance of true clock} = \frac{h_0}{2\tau} + 2h_{-1} + \frac{2}{3}\pi^2 h_{-2}\tau$$

$$\text{Allan variance of 2-state model} = \frac{h_0}{2\tau} + \frac{2}{3}\pi^2 h_{-2}\tau = \frac{S_f}{\tau} + S_g\tau$$

This approximation can be further improved by adjusting the parameters of the 2-state model, S_f and S_g, so that its resulting Allan variance is better matched to the true Allan variance of the clock for certain regions of consequence.



Problem 11.1

In the filtering of GPS measurements, the Kalman filter rate is nominally 1 Hz, or equivalently, the time interval is 1 sec. When measurements are not available every second, the time interval between measurements generally varies, depending on the application. Suppose that the time interval region we are interested in is the range between 1 to 100 sec. One approach is to match the Allan variance profiles for the 2-state model and the true version exactly at the 1- and 100-sec points, the extremes of the range of interest.

(a) Determine expressions for S_f and S_g in terms of the asymptotic Allan variance parameters h_0, h_{-1}, and h_{-2} needed to do so.

(b) Calculate the largest deviation in the Allan variance between the two models within the 1- to 100-sec range for the specific values of $h_0 = 9.4\ (10^{-20})$, $h_{-1} = 1.8\ (10^{-19})$, and $h_{-2} = 3.8\ (10^{-21})$.

(c) Plot the Allan variance for the two models.

11.2 In an integrated INS/GPS arrangement such as the one given in Section 11.4, the attitude states are stabilized even though the information derived from GPS consists only of translational parameters such as position and velocity but nothing with respect to orientation. This is possible because of the loose coupling of the translational parameters to the angular ones involving attitude (including heading).

(a) Set up a covariance analysis of a Kalman filter model for the 11-state INS/GPS model described by Eqs. (11.4.1) through (11.4.5) using the following direction cosines for the $\mathbf{H}$ matrix:

$$[h_{xi} \quad h_{yi} \quad h_{zi}] = \begin{cases} [.3523, .0495, -.9346], & i = 1 \\ [-.6199, -.7406, -.2593], & i = 2 \\ [.9506, .2553, -.1764], & i = 3 \\ [-.9613, -.2129, -.1747], & i = 4 \end{cases}$$

For the $\mathbf{R}$ matrix in Eq. (11.4.5), use $r_\rho = (25\text{ m})^2$ and $r_\delta = (0.01\text{ m/sec})^2$. Use the following values for the initial error variances (zero out the covariance terms):

$$\mathbf{P}_{1,1} = (1000\text{ m})^2$$
$$\mathbf{P}_{2,2} = (10\text{ m/sec})^2$$
$$\mathbf{P}_{3,3} = (0.08\text{ rad})^2$$
$$\mathbf{P}_{4,4} = (1000\text{ m})^2$$
$$\mathbf{P}_{5,5} = (10\text{ m/sec})^2$$
$$\mathbf{P}_{6,6} = (0.08\text{ rad})^2$$
$$\mathbf{P}_{7,7} = (1000\text{ m})^2$$
$$\mathbf{P}_{8,8} = (10\text{ m/sec})^2$$
$$\mathbf{P}_{9,9} = (0.08\text{ rad})^2$$
$$\mathbf{P}_{10,10} = (1000\text{ m})^2$$
$$\mathbf{P}_{11,11} = (100\text{ m/sec})^2$$

For the inertial sensor, the parameters given in Example 10.2 described in Section 10.5 should be used.

With the above model assembled, process the filter's error covar-

iance over 1000 steps and plot the time profiles of the east attitude (platform tilt above the x-axis) error and the azimuth error (platform tilt about the z-axis).

(b) Multiantenna systems are capable of deriving attitude information from GPS signals. They are known as *attitude determination systems* (34). Suppose, without further simplification, that a GPS-derived three-axes attitude solution is available in synchronism with the GPS pseudorange and delta range measurements once per second. Assume further that when the GPS attitude solution is resolved into the three attitude components (about the x-, y-, and z-axis), their measurement errors are uncorrelated between components and also uncorrelated in time. Reformulate the measurement model from (a) to incorporate the three additional attitude "measurements." For the measurement noise variance associated with these measurements, use $(0.01 \text{ rad})^2$.

With this model, process the filter's error covariance over 1000 steps and plot the time profiles of the east attitude (platform tilt about the x-axis) error and the azimuth error (platform tilt about the z-axis). Compare the rates of convergence and steady-state error levels between the models used in (a) and (b).

11.3 Derive the process noise matrix $\mathbf{Q}$ for the position-velocity-acceleration (PVA) dynamical model whose transfer function block diagram is shown in Fig. 11.12 in terms of the integration interval Δt and spectral amplitude S of the input white noise. Choose the three states required for a complete description to be position, velocity, and acceleration.

11.4 Unlike most real-life positioning problems, the positioning of a train is unique in being essentially a one-dimensional measurement situation—its one degree of freedom is along the track to which it is constrained. To take advantage of this reduction in the model dimensionality, the exact trajectory of the railroad track must be known and linearized for the approximate vicinity of the train's location.

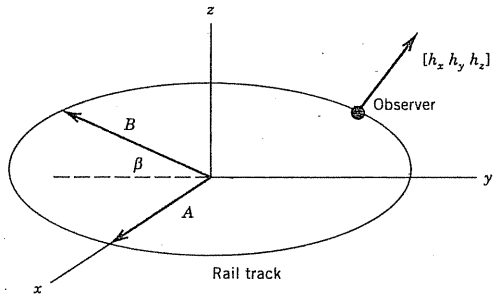
(a) Begin by assuming the standard four-variable GPS measurement model:

$$\tilde{z} = \rho - \hat{\rho} = [h_x(t) \quad h_y(t) \quad h_z(t) \quad 1] \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ c\Delta t \end{bmatrix}$$

Let the rail track (see the figure) be described by the set of parametric equations:

$$x = A \cos \psi$$
$$y = B \sin \psi \cos \beta$$
$$z = B \sin \psi \sin \beta$$

where ψ is the parametric variable and β is a fixed angle of inclination of the elliptical track whose semimajor and semiminor axes are A and B. Using a linearized approximation of the above set of parametric equations, rewrite the measurement model such that the state variables



Problem 11.4

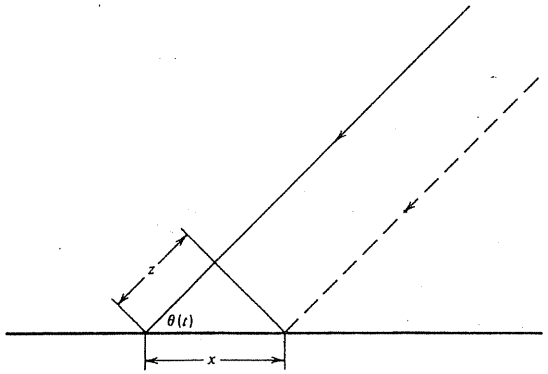
comprise just ψ and $c\Delta t$. How is the new linear measurement connection vector $\mathbf{h}$ written in terms of $h_x(t)$, $h_y(t)$, $h_z(t)$?

(b) What is the minimum number of satellites required to solve the positioning problem of the train with given track trajectory information, as is the case here?

(c) Formulate a similar measurement model that allows for dynamical motion (start out with the position-velocity model of Eqs. 11.5.1 and 11.5.7) by also assuming the same track described above.

(d) What is the random-process model (specify $\boldsymbol{\Phi}$ and $\mathbf{Q}$ matrices) for the modified state vector that corresponds to the position-velocity model.

11.5 Sometimes, a simple *moving-average* process is used to filter GPS data. A direct solution of the position is obtained from the linearized measurement equations (Eqs. 11.1.3) for each set of measurements made. This unfiltered solution of each set of satellite measurements made at an instance in time constitutes one data point. When a fixed number N of consecutive data points are collected, an average is computed. Then when a new data point is available, it



Problem 11.5

then displaces the oldest data point in the collection. With this, the averaging time window slides over to absorb new information and "forgets" old information after N samples, thereby providing the averaging needed to obtain solution accuracy while concurrently adapting to changes in the solution.

The main philosophical difference between this crude estimator and the Kalman filter lies in the weighting of the data samples. The simple average draws equally weighted information from among all the samples in the window (see Section 5.1). To compare the moving average and the Kalman filter, a one-dimensional model can be simulated.

Process:

$$x_{k+1} = x_k + w_k; \qquad Ew_k^2 = 100 \text{ m}^2$$

Measurement:

$$z_k = [\cos\ \theta(t_k)] \cdot x_k + v_k; \qquad Ev_k^2 = 225 \text{ m}^2$$

Let

$$\theta(t_k) = 75° + \frac{t_k}{43{,}600} \cdot 360°; \qquad t_k = 10\,k, \quad k = 0, 1, \ldots 300$$

(a) Generate the state trajectory and the corresponding measurement data for the process given above. Then use the measurement data in the Kalman filter to obtain a time profile for the position estimates. By taking the difference between the position estimates and the truth state trajectory generated, we arrive at the error profile.

(b) Use the same measurement data generated in (a) to assess the moving-average process as well. To obtain the unfiltered solution, the Kalman filter can be used by setting the value of Q to be very high, say 10^6. The unfiltered solution is then filtered with the moving-average process:

$$\hat{x}_k = \sum_{i=k-N}^{k} X_k; \qquad k = N, N+1, N+2, \ldots$$

Here again as in (a), generate the error profile by taking the difference between the moving-average solution and the truth state trajectory.

(c) Compare the results obtained for the error profiles using the Kalman filter and the moving-average filter. Explain the differences.

11.6 An elementary 5-state Kalman filter model for a GPS receiver is discussed in Section 11.5. Even though this model is used only rarely because of its limitations, it is still useful as a means of assessing the effect of varying key filter parameters. We will now pursue this further.

We begin by defining the basic 5-state model as follows:

$x_1 = \Delta x$
$x_2 = \Delta y$
$x_3 = \Delta z$
} The position perturbations from the nominal (x, y, z)

$x_4 = c\Delta t$ Range equivalent of clock bias

$x_5 = c\Delta \dot{t}$ Range equivalent of clock drift

First, write out the general expressions for the state transition matrix ϕ_k and the process noise matrix $\mathbf{Q}_k$. Let the step size be denoted as Δt. Further assume that the measurements consist of only four pseudoranges at each update (no delta range) and that the model has been linearized. Therefore, the measurement model will be of the form

$$\mathbf{z}_k = \mathbf{H}_k\,\mathbf{x}_k + \mathbf{v}_k$$

where $\mathbf{z}_k$ is 4×1, $\mathbf{H}_k$ is 4×5, $\mathbf{x}_k$ is 5×1, and $\mathbf{v}_k$ is 4×1. We will make the usual assumption that the measurement errors are white and have zero cross-correlation. Therefore, $\mathbf{R}_k$ is diagonal. The motion of the GPS satellites will be assumed to be slow relative to the time span of interest here, so we will regard $\mathbf{H}_k$ to be a constant matrix as follows:

$$\mathbf{H}_k = \begin{bmatrix} .3523 & .0495 & -.9346 & 1.0 & 0.0 \\ -.6199 & -.7406 & -.2593 & 1.0 & 0.0 \\ .9506 & .2553 & -.1764 & 1.0 & 0.0 \\ -.9613 & -.2129 & -.1747 & 1.0 & 0.0 \end{bmatrix}$$

(The specified $\mathbf{H}_k$ matrix would provide a snapshot solution GDOP of about 3.0, which is typical for GPS.) The model is now complete except for numerical values.

(a) Let each of the white-noise spectral amplitudes that drive the random walk position states be

$$S_p = 1.0\,\frac{(\text{m/sec})^2}{\text{rad/sec}}$$

(This allows for a modest amount of receiver motion.) Also, let the clock model spectral amplitudes be

$$S_f = 0.4\ (10^{-18}) \text{ sec}$$
$$S_g = 1.58\ (10^{-18}) \text{ sec}^{-1}$$

Equations (11.3.1), (11.3.2), and (11.3.3) can then be used to find the clock $\mathbf{Q}$ parameters.

[*Note:* The results of these calculations must be rescaled by the square of the speed of light (c^2) to yield the correct units.] Also let

$$\text{Step size } \Delta t = 1.0 \text{ sec}$$
$$\text{Measurement error variance} = 25 \text{ m}^2$$

Finally, the filter error covariance matrix is to be initialized as a diagonal matrix as follows:

$$\mathbf{P}_0^- = \begin{bmatrix} (100 \text{ m})^2 & 0 & 0 & 0 & 0 \\ 0 & (100 \text{ m})^2 & 0 & 0 & 0 \\ 0 & 0 & (100 \text{ m})^2 & 0 & 0 \\ 0 & 0 & 0 & (300 \text{ m})^2 & 0 \\ 0 & 0 & 0 & 0 & (30 \text{ m/sec})^2 \end{bmatrix}$$

Specification of the numerical values is now complete.

It is desirable to have the receiver filter settle to steady-state con-

dition (or nearly so) in about 1 min. To test this, cycle the filter error covariance equations through 60 steps and note the rate of convergence. The filter parameters have been assumed to be constant, so the filter's characteristic roots in the z-plane can also be calculated (see Section 7.6). Calculate the characteristic roots and see if they are consistent with the results of the covariance analysis.

(b) Now increase the elements of the $\mathbf{Q}_k$ matrix by a factor of 10. Repeat the experiment of part (a) and note how this change affects both the error covariance convergence and the characteristic roots. A comparative plot of key elements of the $\mathbf{P}$ matrices will be helpful here.

(c) Use the same inflated values of $\mathbf{Q}_k$ as in part (b) and now increase the terms of the $\mathbf{R}_k$ matrix by a factor of 10. Perform an analysis similar to that of part (b) and note the effect of increasing the terms of the $\mathbf{R}_k$ matrix.

11.7 One way to reduce the number of states in a GPS measurement model is to form a new set of measurements from the original set by means of differencing (see Example 11.4). By forming proper linear combinations of the original measurements, the common clock error term found in them can be eliminated.

(a) Starting out with the 5-state model given in Problem 11.6, modify the process and state models to utilize the following differencing scheme; the resulting model should have only three states:

$$\begin{bmatrix} z_1' \\ z_2' \\ z_3' \end{bmatrix} = \begin{bmatrix} z_1 - z_2 \\ z_2 - z_3 \\ z_3 - z_4 \end{bmatrix}$$

(b) Compute the steady-state variance associated with the position error states Δx, Δy, and Δz. Similarly, compute the steady-state variance for the same three terms using the 5-state model given in Problem 11.6. Which model yields smaller values for the position error variances? Why are they different?

(c) Modify the 5-state model to yield the same results as the 3-state model. Due to numerical roundoff errors, it is sufficient to deem that two values are the same within the first four decimal places.

11.8 In addition to relative positioning, the continuous-carrier-phase measurement can also be used for absolute *point* positioning as well. A similar type of measurement principle was originally used with the Navy's transit satellite system (39). Point positioning refers to the determination of a location that is fixed or stationary.

Starting with the continuous-carrier-phase measurement described in Eq. (11.2.2), we then proceed to formulate a measurement model that accounts for its change over time (for a single satellite):

$$\begin{aligned} \phi_k - \phi_0 &= (\psi_k + N_\phi + \beta_{\phi k} + \nu_{\phi k}) - (\psi_0 + N_\phi + \beta_{\phi 0} + \nu_{\phi 0}) \\ &= (\psi_k - \psi_0) + \beta'_{\phi k} + \nu'_{\phi k} \end{aligned}$$

By linearizing, we get

$$[\phi_k - \psi_k(\mathbf{x}')] - [\phi_0 - \psi_0(\mathbf{x}')] = [h_{xk} \quad h_{yk} \quad h_{zk} \quad 1 \quad 0] \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ c\Delta t_k \\ c\Delta \dot{t}_k \end{bmatrix}$$

$$- [h_{x0} \quad h_{y0} \quad h_{z0} \quad 1 \quad 0] \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ c\Delta t_0 \\ c\Delta \dot{t}_0 \end{bmatrix} + \beta'_{\phi k} + \nu'_{\phi k}$$

On recombining terms, we get

$$[\phi_k - \psi_k(\mathbf{x}')] - [\phi_0 - \psi_0(\mathbf{x}')]$$

$$= [(h_{xk} - h_{x0}) \quad (h_{yk} - h_{y0}) \quad (h_{zk} - h_{z0}) \quad 1 \quad 0]_k \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \\ c(\Delta t_k - \Delta t_0) \\ c(\Delta \dot{t}_k - \Delta \dot{t}_0) \end{bmatrix}_k + \beta'_{\phi k} + \nu'_{\phi k}$$

The measurement model above can be treated as a 5-state model by absorbing $\beta'_{\phi k}$, into the position states Δx, Δy, Δz, which are otherwise random constants. In absorbing $\beta'_{\phi k}$, the position states should then be modeled as random walk processes. The states $c(\Delta t_k - \Delta t_0)$ and $c(\Delta \dot{t}_k - \Delta \dot{t}_0)$ take on the same character as the usual range (clock) states (see Problem 11.6), except that the initial value of the $c(\Delta t_k - \Delta t_0)$ state at t_0 is known to be zero with a probability of 1 (error variance of zero).

(a) Set up and run a Kalman filter simulation for the model formulated above. For the random walk states Δx, Δy, and Δz use a spectral amplitude of $S_p = (0.1 \text{ m})^2$ for its process noise. For the timing states, use the model given in Problem 11.6 and a spectral amplitude of $S_f = 0.4$ (10^{-18}) sec and $S_g = 1.58$ (10^{-18}) sec^{-1}. Let the measurement noise $\mathbf{R} = (0.005)^2\,\mathbf{I}$ ($\mathbf{I}$ is a 4×4 identity matrix). Use the idealized geometry given for a typical GPS example in Appendix B (choose satellites 7, 10, 17, and 24) to obtain the unit direction vectors for the $\mathbf{H}$ matrix, and plot the variances of the Δx, and Δy, and Δz states over an interval of 1800 sec.

(b) Make a second run using the same parameters except use satellites 7, 10, and 17 for this run. Graphically compare in general terms the convergence of the errors in the Δx, Δy, and Δz position states for both cases.

11.9 The geometric dilution of precision (commonly called GDOP) introduced in Section 11.6 is a measure of the position solution accuracy as a function of satellite geometry. It is the "snapshot" rms error of the combined position and time solution caused by 1-m rms of error in the measurements. The term *snapshot* implies that the solution is entirely dependent on the satellite measurements made at one specific instance of time. In other words, the solution is unfiltered.

A Kalman filter model can be set up to derive the GDOP. Use a 4-state model [three position error states and a range (clock) error state]. Set up the $\mathbf{H}$ matrix for the ideal satellite constellation described in Appendix B using satellites 4, 7, 11, and 23. Choose the proper values for the Kalman filter parameters to obtain such a snapshot solution. The square root of the trace of the updated $\mathbf{P}$ matrix is defined as the GDOP. What is the GDOP value for this particular set of satellites at time $t = 1200$ sec?

REFERENCES CITED IN CHAPTER 11

1. *Global Positioning System,* Vols. I, II, III, and IV, The Institute of Navigation, Washington, DC, 1980–86.
2. B. Hofmann-Wellenhof, H. Lichtenegger, and J. Collins, *Global Positioning System, Theory and Practice,* 2nd ed., New York: Springer-Verlag, 1993.
3. *Department of Defense World Geodetic System 1984—Its Definition and Relationships with Local Geodetic Systems,* Defense Mapping Agency tech. rept. no. 8350.2, Sept. 1987.
4. J. C. Rambo, "Receiver Processing Software Design of the Rockwell International DoD Standard GPS Receivers," *Proceedings of the 2nd International Technical Meeting of the Satellite Division of the Institute of Navigation,* Colorado Springs, CO, Sept. 27–29, 1989, pp. 217–225.
5. R. M. Kalafus, J. Vilcans, and N. Knable, "Differential Operation of NAVSTAR GPS," *Navigation, J. Inst. Navigation,* 30(3):187–204 (Fall 1983).
6. E. G. Blackwell, "Overview of Differential GPS Methods," *Navigation, J. Inst. Navigation,* 32(2):114–125 (Summer 1985).
7. P. W. Ward, "GPS Receiver RF Interference, Monitoring, Mitigation, and Analysis Techniques," *Navigation, J. Inst. Navigation,* 41(4):367–391 (Winter 1994–95).
8. P. Y. C. Hwang and R. G. Brown, "GPS Navigation: Combining Pseudorange with Continuous Carrier Phase Using a Kalman Filter," *Navigation, J. Inst. Navigation,* 37(2)(Summer 1990) pp. 181–196.
9. K. W. Ulmer, P. Y. Hwang, B. A. Disselkoen, and M. R. Wagner, "Accurate Azimuth from a Single PLGR + GLS DoD GPS Receiver Using Time Relative Positioning," *Proceedings of the 8th International Technical Meeting of the Satellite Division of the Institute of Navigation,* Palm Springs, CA, Sept. 1995, pp. 1733–1741.
10. R. I. Abbot, Y. Bock, C. C. Counselman, R. W. King, S. A. Gourevitch, and B. J. Rosen, "Interferometric Determination of GPS Satellite Orbits," *Proceedings of the 1st International Symposium on Precise Positioning with the Global Positioning,* Rockville, MD, April 15–19, 1985, Vol. I, pp. 63–72.
11. W. A. Feess and S. G. Stephens, "Evaluation of GPS Ionospheric Time Delay Algorithm for Single-Frequency Users," *Proceedings of the IEEE Position Location and Navigation Symposium (PLANS '86),* Las Vegas, NV, Nov. 4–7, 1986, pp. 206–213.
12. G. T. Kremer, R. M. Kalafus, P. V. W. Loomis, and J. C. Reynolds, "The Effect of Selective Availability on Differential GPS Corrections," *Proceedings of the 2nd International Technical Meeting of the Satellite Division of the Institute of Navigation,* Colorado Springs, CO, Sept. 27–29, 1989, pp. 307–312.
13. P. Y. Hwang, "Recommendations for Enhancement of RTCM-104 Differential Standard and Its Derivatives," *Proceedings of the 6th International Technical Meeting of the Satellite Division of the Institute of Navigation,* Salt Lake City, UT, Sept. 22–24, 1993, pp. 1501–1508.
14. M. S. Braasch, "Isolation of GPS Multipath and Receiver Tracking Errors," *Navigation, J. Inst. Navigation,* 41(4):415–434 (Winter 1994–95).

15. J. A. Barnes, "Models for the Interpretation of Frequency Stability Measurements," NBS tech. note 683, Boulder, CO, Aug. 1976.
16. A. J. van Dierendonck, J. B. McGraw, and R. G. Brown, "Relationship Between Allan Variances and Kalman Filter Parameters," *Proceedings of the 16th Annual Precise Time and Time Interval (PTTI) Applications and Planning Meeting,* NASA Goddard Space Flight Center, Nov. 27–29, 1984, pp. 273–293.
17. R. G. Brown, "Kalman Filter Modeling," *Proceedings of the 16th Annual Precise Time and Time Interval (PTTI) Applications and Planning Meeting,* NASA Goddard Space Flight Center, Nov. 27–29, 1984, pp. 261–272.
18. J. Westrop, M. E. Napier, and V. Ashkenazi, "Cycle Slips on the Move: Detection and Elimination," *Proceedings of the 2nd International Technical Meeting of the Satellite Division of the Institute of Navigation,* Colorado Springs, CO, Sept. 27–29, 1989, pp. 31–34.
19. R. J. Milliken and C. J. Zoller, "Principle of Operation of NAVSTAR and System Characteristics," *Global Positioning System,* Vol. I, Institute of Navigation, Washington, DC, 1980, pp. 3–14.
20. B. W. Remondi, "Modeling the GPS Carrier Phase for Geodetic Applications," *Proceedings of the 1st International Symposium on Precise Positioning with the Global Positioning System,* Rockville, MD, April 15–19, 1985, Vol. I, pp. 325–336.
21. R. G. Brown and P. Y. C. Hwang, "A Kalman Filter Approach to Precision GPS Geodesy," *Navigation, J. Inst. Navigation,* 30(4):338–349 (Winter 1983–84).
22. P. Y. C. Hwang and R. G. Brown, "GPS Geodesy: Experimental Results Using the Kalman Filter Approach," *Proceedings of the IEEE Electronics and Aerospace Systems Conference (EASCON '85),* Washington, DC, Oct. 28–30, 1985, pp. 9–18.
23. P. S. Jorgenson, "An Assessment of Ionospheric Effects on the GPS User," *Navigation, J. Inst. Navigation,* 36(2):195–204.
24. P. W. McBurney, "A Robust Approach to Reliable Real-Time Kalman Filtering," *Proceedings of the IEEE Position Location and Navigation Symposium (PLANS '90),* Las Vegas, NV, March 21–23, 1990, pp. 549–556.
25. M. S. Braasch, N. M. Fink, and K. Duffus, "Improved Modeling of GPS Selective Availability," *Proceedings of the 1993 National Technical Meeting of the Institute of Navigation,* San Francisco, CA, Jan. 22–22, 1993, pp. 121–130.
26. B. W. Remondi, "Performing Centimeter-Level Surveys in Seconds with GPS Carrier Phase: Initial Results," *Navigation, J. Inst. Navigation,* 32(4):386–400 (Winter 1985–86).
27. P. Y. C. Hwang, "Kinematic GPS: Resolving Integer Ambiguities on the Fly," *Proceedings of the IEEE Position Location and Navigation Symposium (PLANS '90),* Las Vegas, NV, March 21–23, 1990, pp. 579–586.
28. L. Chin, "Feasibility Study of Using GPS to Calibrate an Instrumentation Radar," *Navigation, J. Inst. Navigation,* 32(1):57–67 (Spring 1985).
29. P. Braisted, R. Eschenbach, and A. Tiwari, "Combining LORAN and GPS—The Best of Both Worlds," *Global Positioning System,* Vol. III, Institute of Navigation, Washington, DC, 1986, pp. 235–240.
30. P. Axelrad and B. W. Parkinson, "Closed Loop Navigation and Guidance for Gravity Probe B Orbit Insertion," *Navigation, J. Inst. Navigation,* 36(1):45–61 (Spring 1989).
31. A. Mathews, "Utilization of Fiber-Optic Gyros in Inertial Measurement Unit," *Proceedings of the IEEE Position Location and Navigation Symposium (PLANS '90),* Las Vegas, NV, March 21–23, 1990, pp. 147–160.
32. C. Kee, B. W. Parkinson, and P. Axelrad, "Wide Area Differential GPS," *Navigation, J. Inst. Navigation,* 38(2):123–145 (Summer 1991).
33. R. G. Brown, "A Baseline GPS RAIM Scheme and a Note on the Equivalence of Three RAIM Methods," *Navigation, J. Inst. Navigation,* 39(3):301–316 (Fall 1992).

34. R. Brown and P. W. Ward, "A GPS Receiver with Built-In Precision Pointing Capability," *Proceedings of the IEEE Position Location and Navigation Symposium (PLANS '90),* Las Vegas, NV, March 21–23, 1990, pp. 83–93.

Additional General References on GPS

35. B. W. Parkinson and J. J. Spilker, Jr. (eds.), *Global Positioning System: Theory and Applications,* Vols. 1 and 2, American Institute of Aeronautics and Astronautics, Inc., Washington, DC, 1996.
36. E. D. Kaplan (ed.), *Understanding GPS Principles and Applications,* Artech House, Boston, 1996.

APPENDIX A

Laplace and Fourier Transforms

Elementary treatments of Laplace and Fourier transforms usually gloss over matters of convergence and formal inversion of these transforms by the inversion integral. There are problems in signal analysis where these matters are important, though (e.g., Wiener filter theory), and thus we will embellish on these ideas here. It is not the intent here to teach linear transform theory from the beginning. We assume that the reader has the usual manipulative skills in Laplace and Fourier transforms that one would normally get in an undergraduate electrical engineering program. [For example, see Mayhan (1) or Kamen (2) for Fourier transforms and Dorf (3) or D'Azzo and Houpis (4) for Laplace transforms.] Here the emphasis will be to place Laplace and Fourier transforms in perspective relative to each other, and to discuss, in particular, formal inversion of these transforms by the inversion integral. We begin with the one-sided Laplace transform.

A.1 THE ONE-SIDED LAPLACE TRANSFORM

Electrical engineers usually first encounter Laplace transforms in circuit analysis, and then again in linear control theory. In both cases the central problem is one of finding the system response to an input initiated at $t = 0$. Since the time history of the system prior to $t = 0$ is summarized in the form of the initial conditions, the ordinary one-sided Laplace transform serves us quite well. Recall that it is defined as

$$F(s) = \int_{0+}^{\infty} f(t)e^{-st}\,dt \tag{A.1}$$

Table A.1. Common One-Sided Laplace Transform Pairs[a]

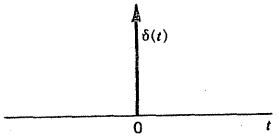
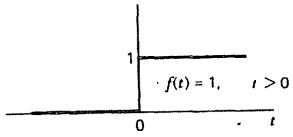
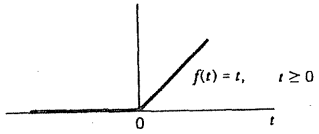
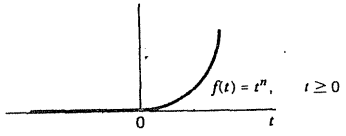
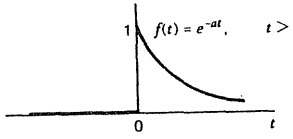
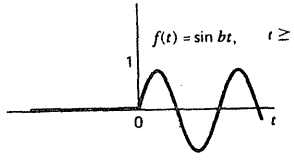
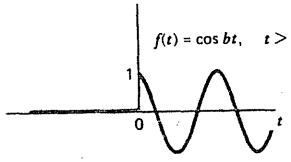
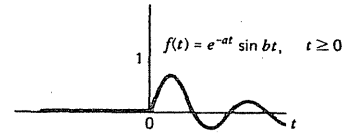
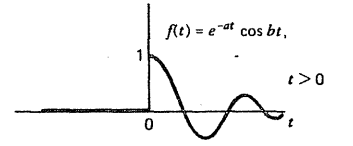
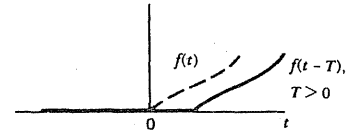
Name	Pictorial Description	Laplace Transform
Unit impulse (Area is to right of origin)		1
Unit step		$\frac{1}{s}$
Unit ramp		$\frac{1}{s^2}$
[b]General power of t		$\frac{n!}{s^{n+1}}$
Damped exponential		$\frac{1}{s+a}$
Sine wave		$\frac{b}{s^2+b^2}$
Cosine wave		$\frac{s}{s^2+b^2}$

Table A.1. *(Continued)*

Name	Pictorial Description	Laplace Transform
Damped sine wave		$\frac{b}{(s+a)^2+b^2}$
Damped cosine wave		$\frac{s+a}{(s+a)^2+b^2}$
Delayed positive-time function		$F(s)e^{-Ts}$

[a]Time functions having a discontinuity at $t = 0$ are intentionally left undefined at the origin in this table. The missing value does not affect the direct transform. See Exercise A.1 at the end of this appendix for a discussion of the appropriate choice of $f(0)$ to assure compatibility with the inversion integral.

[b]When n is not an integer, $n!$ must be interpreted as the gamma function $\Gamma(n+1)$.

The defining integral is, of course, insensitive to $f(t)$ for negative t; but, for reasons that will become apparent shortly, we *arbitrarily* set $f(t) = 0$ for $t < 0$ in one-sided transform theory. The integral of Eq. (A.1) has powerful convergence properties because of the e^{-st} term. We know it will always converge somewhere in the right-half s-plane, provided that we consider only inputs (and responses) that increase no faster than at some fixed exponential rate. This is usually the case in circuits and control problems, and hence the actual region of convergence is of little concern. A common region of convergence is tacitly assumed to exist somewhere, and we simply adopt a table look-up viewpoint for getting back and forth between the time and complex s domains. For reference purposes a brief list of common transform pairs is given in Table A.1. Note again that we have intentionally defined all time functions in the table to be zero for $t < 0$. We will have occasion later to refer to such functions as *positive-time* type functions. It is also worth mentioning that the impulse function of one-sided transform theory is considered to have all its area to the right of the origin in the limiting process. Thus it is a positve-time function. (The word *function* is abused a bit in describing an impulse, but this is common usage, so it will be continued here.)

A.2 THE FOURIER TRANSFORM

The Fourier transform is used widely in communications theory where we often wish to consider signals that are nontrivial for both positive and negative time. Thus, a two-sided transform is appropriate. Recall that the Fourier transform of $f(t)$ is defined as

$$F(j\omega) = \int_{-\infty}^{\infty} f(t)e^{-j\omega t}\,dt \tag{A.2}$$

We know, through the evolution of the Fourier transform from the Fourier series, that $F(j\omega)$ has the physical significance of signal spectrum. The parameter ω in Eq. (A.2) is $(2\pi) \times$ (frequency in hertz), and in elementary signal analysis we usually consider ω to be real. This leads to obvious convergence problems with the defining integral, Eq. (A.2), and is usually circumvented simply by restricting the class of time functions being considered to those for which convergence exists for real ω. The two exceptions to this are constant (d-c) and harmonic (sinusoidal) signals. These are usually admitted by going through a limiting process that leads to Dirac delta functions in the ω domain. Even though the class of time functions allowed is somewhat restrictive, the Fourier transform is still very useful because many physical signals just happen to fit into this class (e.g., pulses and finite-energy signals). If we take convergence for granted, we can form a table of transform pairs, just as we did with Laplace transforms, and Table A.2 gives a brief list of common Fourier transform pairs.

For those who are more accustomed to one-sided Laplace transforms than Fourier transforms, there are formulas for getting from one to the other. These are especially useful when the time functions have either even or odd symmetry. Let $f(t)$ be a time function for which the Fourier transform exists, and let

$$\mathfrak{F}[f(t)] = \text{Fourier transform of } f(t)$$
$$F(s) = \text{one-sided Laplace transform of } f(t)$$

Then, if $f(t)$ is even,

$$\mathfrak{F}[f(t)] = F(s)|_{s=j\omega} + F(s)|_{s=-j\omega} \tag{A.3}$$

or if $f(t)$ is odd,

$$\mathfrak{F}[f(t)] = F(s)|_{s=j\omega} - F(s)|_{s=-j\omega} \tag{A.4}$$

These formulas follow directly from the defining integrals of the two transforms.

Table A.2. Common Fourier Transform Pairs

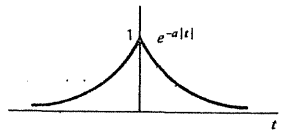
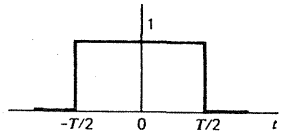
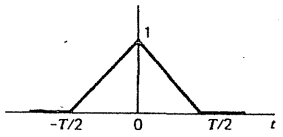
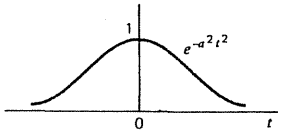
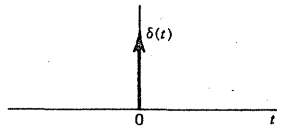
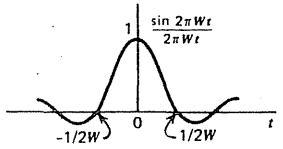
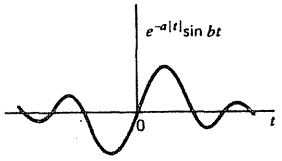
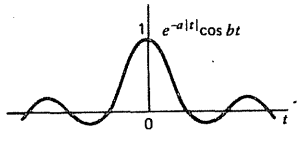
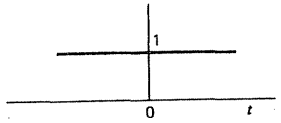
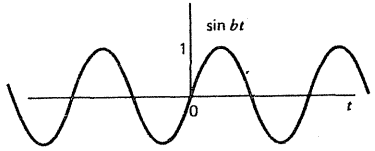
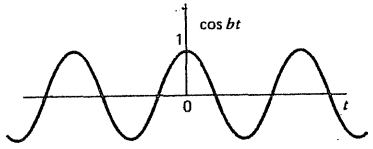
Name	Pictorial Description	Fourier Transform
Damped exponential		$\frac{2a}{\omega^2 + a^2}$
Rectangular pulse		$T\frac{\sin(\omega T/2)}{(\omega T/2)}$
Triangular pulse		$\frac{T}{2}\left[\frac{\sin(\omega T/4)}{(\omega T/4)}\right]^2$
Gaussian pulse		$\frac{\sqrt{\pi}}{a}e^{-(\omega^2/4a^2)}$
Symmetric impulse		1
Sinc function (sinc $2Wt$)		$F(j\omega) = \begin{cases} \frac{1}{2W}, & \|\omega\| < 2\pi W \\ 0, & \|\omega\| > 2\pi W \end{cases}$

Table A.2. *(Continued)*

Name	Pictorial Description	Fourier Transform
Damped sine wave		$\frac{ja}{a^2 + (\omega + b)^2} - \frac{ja}{a^2 + (\omega - b)^2}$
Damped cosine wave		$\frac{a}{a^2 + (\omega + b)^2} + \frac{a}{a^2 + (\omega - b)^2}$
Constant		$2\pi\,\delta(\omega)$
Sine wave		$j\pi\,\delta(\omega + b) - j\pi\,\delta(\omega - b)$
Cosine wave		$\pi\delta(\omega + b) + \pi\delta(\omega - b)$

A.3 TWO-SIDED LAPLACE TRANSFORM

The usual Fourier transform with ω real serves us well for much of signal analysis, but there are occasions where we wish to consider functions for which this transform does not exist. In such cases the so-called two-sided Laplace transform is sometimes useful. It is defined as

$$F(s) = \int_{-\infty}^{\infty} f(t)e^{-st}\,dt \qquad \text{(A.5)}$$

It can be seen to be identical in form to the Fourier transform except for notation, that is, $j\omega$ is replaced with s. Innocent as this change may appear at first glance, it is important, because we now wish to let s be complex. Furthermore, we do not wish to place any restrictions on the type of time functions being considered, other than to say the integral of Eq. (A.5) must converge *somewhere* in the s-plane. Now the region of convergence becomes important, as an example will illustrate.

EXAMPLE A.1

Consider the two time functions shown in Fig. A.1. Their respective two-sided Laplace transforms may be found from Eq. (A.5).

For signal $f_1(t)$ of Fig. A.1a,

$$F_1(s) = \int_{-\infty}^{0} e^{at}e^{-st}\,dt = \frac{-1}{s-a}, \qquad \text{for Re}[s] < a$$

For signal f_2(,t) of Fig. A.1b,

$$F_2(s) = \int_{0}^{\infty} e^{at}e^{-st}\,dt = \frac{1}{s-a}, \qquad \text{for Re}[s] > a$$

Here we have the uncomfortable situation of two different time functions having the same transform except for sign! There is a saving feature though—their regions of convergence are nonoverlapping. Hence, if we add the qualification of the convergence region to the functional form in the s-domain, we then restore the one-to-one transform-pair relationship that we must have in any transform algebra. In this case the entry in our table of transform pairs might appear as follows:

Time Function	Transform
$f(t) = \begin{cases} e^{at}, & t < 0 \\ 0, & t > 0 \end{cases}$	$\frac{-1}{s-a}$ *and* region of convergence is Re$[s] < a$
$f(t) = \begin{cases} 0, & t < 0 \\ e^{at}, & t > 0 \end{cases}$	$\frac{1}{s-a}$ *and* region of convergence is Re$[s] > a$

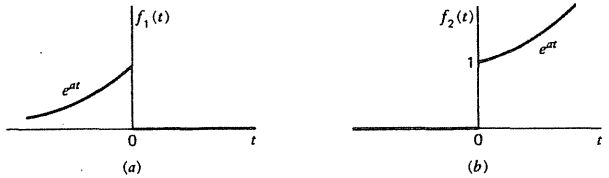


Figure A.1 Time functions for Example A.1. (*a*) Negative-time function. (*b*) Positive-time function.

Tacking on the region of convergence in our table of transform pairs is a bit cumbersome, but it must be done, or at least tacitly implied somehow.

Table A.3 shows a few common time functions, their two-sided Laplace transforms, and their respective regions of convergence. The list is by no means complete. It is simply intended to show the variety of situations that can occur. Note that a number of the entries have regions of convergence that do not include the $j\omega$-axis, and therefore usual Fourier transforms will not exist for these time functions. Also, two of the time functions shown do not have two-sided Laplace transforms, because the defining integral does not converge anywhere in the s-plane. ■

The Inversion Integral

To compile and work with two-sided transform tables such as Table A.2 would be awkward, to say the least. The addition of the specification of convergence region to the functional form is like adding an extra dimension to the transform. Thus, rather than use cumbersome two-sided transform tables for inversion, it is more convenient to use one-sided transform tables with the appropriate interpretation as to positive- and negative-time parts of the time function. This is where the formal rules of inversion are helpful. Beginning with the Fourier transform, we have the inversion formula

$$f(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} F(j\omega)e^{j\omega t}\,d\omega \tag{A.6}$$

Recall that this came from a limiting situation of the complex Fourier series where the expansion interval was allowed to approach infinity. The corresponding inversion integral for the Laplace transform is

$$f(t) = \frac{1}{2\pi j}\int_{c-j\infty}^{c+j\infty} F(s)e^{st}\,ds \tag{A.7}$$

This formula applies to both the one- and two-sided transforms. Note that the integration is along a vertical line of the s-plane, and the constant c is chosen such that *the path of integration lies within the strip of convergence of* $F(s)$. Knowledge of the convergence region is the key to getting the correct inverse transform (i.e., time function) in the inversion process. An example will illustrate this.

EXAMPLE A.2

Consider the transform pair shown as the seventh entry in Table A.3. The transform and the strip of convergence are

$$F(s) = \frac{1}{s-a} + \frac{1}{-s+b}; \qquad a < \mathrm{Re}[s] < b$$

Table A.3. Two-Sided Laplace Transforms

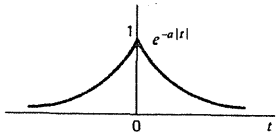
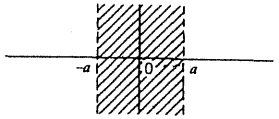
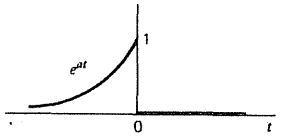
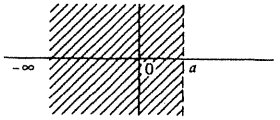
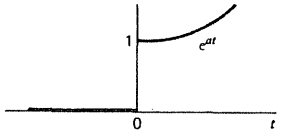
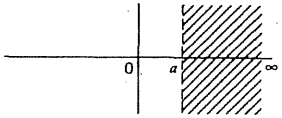
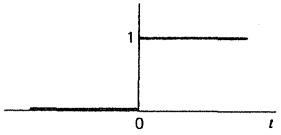
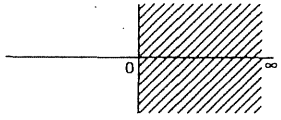
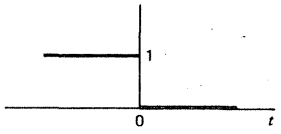
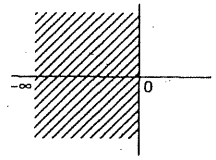
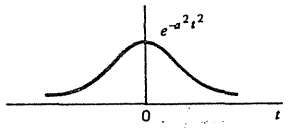
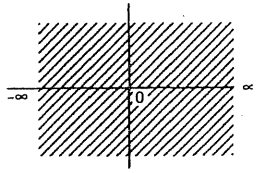
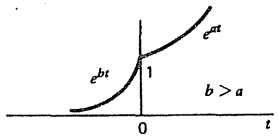
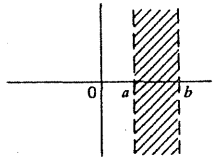
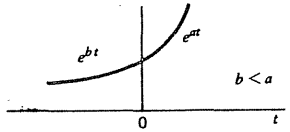
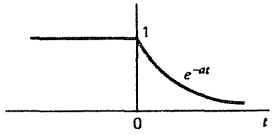
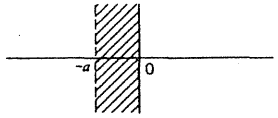
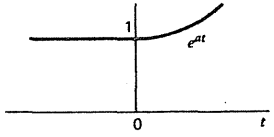
Time Function	Two-sided Laplace Transform	Region of Convergence
	$\frac{2a}{-s^2+a^2}$	
	$\frac{1}{-s+a}$	
	$\frac{1}{s-a}$	
	$\frac{1}{s}$	
	$\frac{1}{-s}$	

Table A.3. *(Continued)*

Time Function	Two-sided Laplace Transform	Region of Convergence
	$\frac{\sqrt{\pi}}{a} e^{(s^2/4a^2)}$	
	$\frac{1}{s-a} + \frac{1}{-s+b}$	
	---	Converges nowhere in s plane
	$\frac{1}{s+a} + \frac{1}{-s}$	
	---	Converges nowhere in s plane

The theory says to choose the path of integration between the poles at a and b as shown in Fig. A.2. One could evaluate the integral by treating the complex function as a sum of real and imaginary parts, and then use real variable calculus for each part. This, however, is working the problem the hard way (see Exercise A.1). An easier way is by means of residue theory (5). It can be shown (6) that the addition of an infinite semicircular path to the left as shown in Fig. A.3 contributes nothing to the integral of Eq. (A.7) *for positive t.* (This can be seen intuitively by noting that the real part of s has a large negative value along this path.) Thus, the positive-time part may be evaluated by the method of residues as follows:

$$f(t) = \frac{1}{2\pi j}\int_{c-j\infty}^{c+j\infty} F(s)e^{st}\,dt$$

$$= \frac{1}{2\pi j} \cdot 2\pi j \cdot [\text{sum of residues of } F(s)e^{st} \text{ within contour } \Gamma]$$

$$= \left[\left(\frac{1}{s-a} + \frac{1}{-s+b}\right)(e^{st})(s-a)\right]_{s=a}$$

$$= e^{at} \quad (\text{for } t > 0 \text{ only})$$

Similarly, it should be apparent that if we close the contour to the right, we get the negative-time portion of $f(t)$. In this example it is

$$f(t) = -\frac{1}{2\pi j} \cdot 2\pi j \cdot \left[\begin{array}{l}\text{sum of residues of } F(s)e^{st} \text{ within} \\ \text{clockwise contour closed to the right}\end{array}\right]$$

$$= \left[-\left(\frac{1}{s-a} + \frac{1}{-s+b}\right)(e^{st})(s-b)\right]_{s=b}$$

$$= e^{bt} \quad (\text{for } t < 0 \text{ only})$$

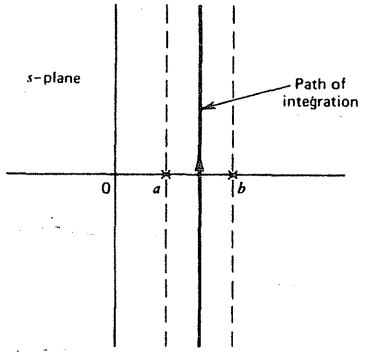


Figure A.2 Convergence strip and path of integration for Example A.2.

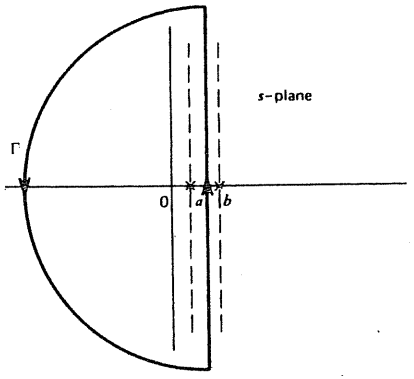


Figure A.3 Closing the path to left to get the positive-time part of $f(t)$.

The procedure illustrated in Example A.2 can be simplified even further by simply noting that poles of $F(s)$ to the left of the vertical path of integration contribute to the positive-time part of $f(t)$, and those to the right yield the negative-time part. Thus, if $F(s)$ is rational, a simpler inversion algorithm may be stated:

1. First think of writing $F(s)$ in terms of a partial fraction expansion and grouping terms together in the form

$$F(s) = \begin{bmatrix}\text{terms with poles to left}\\ \text{of path of integration}\end{bmatrix} + \begin{bmatrix}\text{terms with poles to right}\\ \text{of path of integration}\end{bmatrix} \tag{A.8}$$

Note that $F(s)$ is to be resolved into an *additive* combination, and thus this is not the same as spectral factorization.

2. Then the inverse transform of $F(s)$ is obtained by evaluating the positive- and negative-time parts separately. The first term of Eq. (A.8) yields the positive-time part, and the second gives the negative-time part. Each may be evaluated using one-sided transform methods. [See Exercise A.1 for a discussion of the evaluation of $f(t)$ at $t = 0$.]

Note that if there are no poles to the right of the path of integration (e.g., one-sided Laplace transforms), the inversion integral automatically yields zero for the negative-time part. This is why we must consider $f(t)$ to be zero for negative time in one-sided transform theory, even though the defining integral is insensitive to the negative-time portion of the function. Also note that if the strip of convergence includes the imaginary axis, the two-sided Laplace transform degenerates to the Fourier transform with a simple change in notation (i.e., $j\omega$ is replaced with s). Thus resolving the transform into positive- and negative-time parts is also useful in the evaluation of inverse Fourier transforms.

Our presentation of Laplace and Fourier transforms has been largely an overview, and it was intended to place these transforms in proper perspective relative to each other. Many details have thus been omitted. Also, there are other linear transforms that are of interest in signal analysis, and these have not even been mentioned. Thus, much of linear transform theory has been left unsaid here. For further reading, see Bracewell (7). This is an especially readable book on the subject and it includes many applications.

Exercise A.1.

It was stated earlier that the inversion integral automatically yields zero for $t < 0$ in one-sided transform theory. This statement becomes more convincing after actually performing the integration for a sample problem. Consider the one-sided transform pair:

$$f(t) = \begin{cases} e^{-2t}, & t > 0 \\ 0, & t < 0 \end{cases} \tag{A.9}$$

$$F(s) = \frac{1}{s+2}, \qquad \text{Re}[s] > -2 \tag{A.10}$$

Obviously, the "strip" of convergence is the entire semiinfinite s-plane to the right of $\text{Re}[s] = -2$. Thus, it includes the $j\omega$-axis. A legitimate inversion integral would then be

$$\text{Inverse of } F(s) = \frac{1}{2\pi j}\int_{-j\infty}^{j\infty} \frac{1}{j\omega + 2} e^{j\omega t}\, d(j\omega) \tag{A.11}$$

(a) Rewrite Eq. (A.11) as the sum of real and imaginary parts and then integrate each part separately using ordinary real calculus methods. The end result should be just $f(t)$ as stated in the transform-pair statement. [*Hint:* Take advantage of even and odd symmetry wherever possible. Also, you may find it necessary to use tables of integrals in evaluating the resulting real integrals. The tables given in Dwight (8) are adequate for this problem.]

(b) In Eq. (A.9) the value of $f(t)$ was intentionally left undefined at the point of discontinuity, that is, at $t = 0$. Find the value of $f(t)$ at $t = 0$ as dictated by the inversion integral, Eq. (A.11). Does this seem reasonable in terms of Fourier *series* theory?

(*Hint:* Evaluate the Fourier series for a square wave at the point of discontinuity and compare the result with that obtained from the inversion integral.)

REFERENCES CITED IN APPENDIX A

1. R. J. Mayhan, *Discrete-Time and Continuous-Time Linear Systems,* Reading, MA: Addison-Wesley, 1984.
2. E. Kamen, *Introduction to Signals and Systems,* New York: Macmillan, 1987.
3. R. C. Dorf and R. T. Bishop, *Modern Control Systems,* 7th ed., Reading, MA: Addison-Wesley, 1995.
4. J. J. D'Azzo and C. H. Houpis, *Linear Control System Analysis and Design,* 4th ed., New York: McGraw-Hill, 1995.
5. R. V. Churchill, J. W. Brown, and R. F. Verhey, *Complex Variables and Applications,* 3rd ed., New York: McGraw-Hill, 1976.
6. S. Goldman, *Transformation Calculus and Electrical Transients,* Englewood Cliffs, NJ: Prentice-Hall, 1949.
7. R. N. Bracewell, *The Fourier Transform and Its Applications,* 2nd ed., New York: McGraw-Hill, 1978.
8. H. B. Dwight, *Tables of Integrals and Other Mathematical Data,* 4th ed., New York: Macmillan, 1961.

APPENDIX B

Typical Navigation Satellite Geometry

The exact description of a satellite orbital trajectory can be very complex. A perfectly elliptical orbit can be fully described by six parameters: size and flatness of the ellipse, orientation of the elliptical plane, orientation of the ellipse in its plane, and position of the satellite in the elliptical orbit. These are known as *Keplerian parameters.* To account for external forces other than the ideal satellite–earth interaction, many perturbation correction parameters are included with the standard set of six, listed above, for the purpose of describing the GPS satellite orbits to meet its accuracy requirements. For computer simulation purposes, though, it is seldom important to place the satellites and move them with the kind of detailed precision that must be accounted for in the real physical situation. Although these finer perturbations from the ideal elliptical trajectories may vary up to several hundreds of meters, in relationship to the large satellite–observer distances, they hardly affect the angular relationships that dictate the satellite geometries. It is the latter and its rate of change in time that determine the time-varying observability of the measurement situation, an aspect of GPS that is often the foremost issue addressed by computer simulations.

To aid the reader in running computer simulations of GPS scenarios, we shall derive here equations that analytically describe typical satellite orbits. For simplicity, these orbits are assumed to be perfectly circular with a radius R (26,559,800 m) and inclined 55° to the equatorial plane. As seen in Fig. B.1, the parameter Ω, called the *right ascension,* is the longitude where the orbital plane intersects the equatorial plane as the orbit crosses over from the southern to the northern hemisphere. Although Ω is very nearly constant in inertial space, it changes with respect to an earth-fixed frame of reference because of the rotation of the earth. A second parameter θ represents the location of the satellite as the angular phase in the circular orbit using the right ascension as the reference point. For the assumption of a circular orbit, θ changes at a constant rate, which gives rise to a period of approximately 43,082 sec (slightly less than one half a day). Hence, the earth-centered earth-fixed coordinates for the satellite position in the orbital model of Fig. B.1 are given by

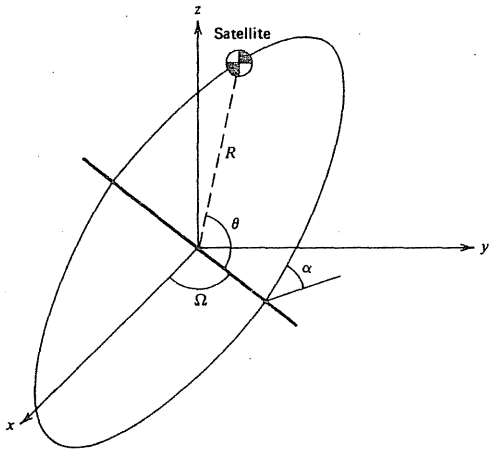


Figure B.1 Circular orbital trajectory ($\alpha = 55°$).

$$x = R[\cos\theta\cos\Omega - \sin\theta\sin\Omega\cos 55°]$$
$$y = R[\cos\theta\sin\Omega + \sin\theta\cos\Omega\cos 55°]$$
$$z = R\sin\theta\sin 55°$$

where

$$\theta = \theta_0 + (t - t_0)\cdot\frac{360}{43{,}082}\ \text{deg}$$

$$\Omega = \Omega_0 - (t - t_0)\cdot\frac{360}{86{,}164}\ \text{deg}$$

$$R = 26{,}559{,}800\ \text{m}$$

By way of concatenating a series of translational and rotational transformations, a direct linear transformation from the earth-centered earth-fixed coordinate frame to the locally level coordinate frame shown in Fig. 10.1 can be obtained as

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} -\sin\theta_u & \cos\theta_u & 0 & x_u\sin\theta_u - y_u\cos\theta_u \\ -\sin\phi_u\cos\theta_u & -\sin\phi_u\sin\theta_u & \cos\phi_u & x_u\sin\phi_u\cos\theta_u + y_u\sin\phi_u\sin\theta_u - z_u\cos\phi_u \\ \cos\phi_u\cos\theta_u & \cos\phi_u\sin\theta_u & \sin\phi_u & -x_u\cos\phi_u\cos\theta_u - y_u\cos\phi_u\sin\theta_u - z_u\sin\phi_u \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

where $[x_u, y_u, z_u]$ are the observer's coordinates, and ϕ_u and θ_u are the local reference latitude and longitude, respectively. One way to use the locally level

coordinate frame and yet avoid the complications of the transformation is to adopt as the observer's local reference, the "zero-zero" location ($\phi_u = 0°$; $\theta_u = 0°$), where we simply have $x' = y$, $y' = z$, and $z' = x - r$ (r is the earth radius − 6380 km).

The following table consists of parameters for the Optimized 24 GPS satellite constellation that serves as a standard model for airborne operational performance analyses (1).

Satellite ID	Ω_0(°)	θ_0(°)[a]
1	272.847	268.126
2	272.847	161.786
3	272.847	11.676
4	272.847	41.806
5	332.847	80.956
6	332.847	173.336
7	332.847	309.976
8	332.847	204.376
9	32.847	111.876
10	32.847	11.796
11	32.847	339.666
12	32.847	241.556
13	92.847	135.226
14	92.847	265.446
15	92.847	35.156
16	92.847	167.356
17	152.847	197.046
18	152.847	302.596
19	152.847	333.686
20	152.847	66.066
21	212.847	238.886
22	212.847	345.226
23	212.847	105.206
24	212.847	135.346

[a] θ_0 is the value of θ at the reference time t_0. Ω_0 is the value of Ω at the reference time t_0. The reference time is midnight July 1, 1993.

The satellite positions in the locally level coordinate frame chosen above can be written as functions of time as follows:

$$x'(t) = R[\cos\theta(t)\sin\Omega(t) + \sin\theta(t)\cos\Omega(t)\cos 55°]$$

$$y'(t) = R\sin\theta(t)\sin 55°$$

$$z'(t) = R[\cos\theta(t)\cos\Omega(t) - \sin\theta(t)\sin\Omega(t)\cos 55°] - r$$

The components of the unit direction vector can be formulated as functions of time t in the following manner:

$$h_x = \frac{-x'(t)}{\sqrt{x'^2(t) + y'^2(t) + z'^2(t)}}$$

$$h_y = \frac{-y'(t)}{\sqrt{x'^2(t) + y'^2(t) + z'^2(t)}}$$

$$h_z = \frac{-z'(t)}{\sqrt{x'^2(t) + y'^2(t) + z'^2(t)}}$$

In a simulation scenario, $t = 0$ corresponds to the reference time t_0 (midnight July 1, 1993). The subset of this 24-satellite constellation that is visible to the observer naturally depends on the location of the observer. For the location that is chosen, there is a simple test of whether a particular satellite is indeed visible. If h_z is negative, then the vertical component of the unit direction vector from the observer to that satellite points upwards in the locally level coordinate frame, implying that the satellite is above the horizon.

REFERENCE CITED IN APPENDIX B

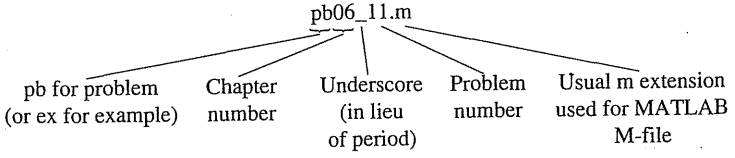
1. *Minimum Operational Performance Standards for Global Positioning System/Wide Area Augmentation System Airborne Equipment,* Document RTCA/DO-229, RTCA Inc., Washington, DC, January 16, 1996.

APPENDIX C
Kalman Filter Software

Software relating to Kalman filtering is contained in a diskette located in a pocket on the inside back cover of this book. The software is intended for tutorial purposes, and no attempt was made to make the programs efficient in terms of speed or lines of code. The programs have been debugged, though, and they are correct to the best of the authors' knowledge. The diskette contains two directories, and the DOS files in each can be easily copied onto the user's PC hard drive. Briefly, the contents of the directories are as follows.

C.1 DIRECTORY MFILES

This directory contains a group of MATLAB M-files that provide solutions of most of the computer-oriented examples and problems in this book. The computer problems are flagged with a small icon, and their file names are derived from the problem numbers as illustrated by the following example:



The directory MFILES also contains a few general script M-files that can be used either alone or imbedded in other M-files as "subroutines." For example, filcovar.m is a fairly general program for performing Kalman filter error covariance analysis. The inputs are the filter parameters, number of steps, etc.; the output is the desired sequence of error covariance matrices stacked side by side in a larger matrix called COVAR. Instructions for entering the input data into filcovar.m are given in the comment statements in the program. Therefore, it is

important that the user browse through the program before using it. This is easily done, because all the M-files are simple ASCII files. The same advice about browsing before using also applies to the problem M-files. Most of these include conclusions at the end of the files, and these will be missed if you do not read the code.

In addition to the general utility and problem M-files, the diskette also includes a few special function M-files that are needed for specific problems. Their use is obvious from inspection of the problem solutions.

In working the problems, the authors suggest that you write your own programs first (either with MATLAB or some other suitable mathematics software); then, simply use the problem solutions given in the diskette as a check on your solutions. There is much to be learned about the Kalman filter operation by writing your own code line by line. This is easy to do with MATLAB, and it is not especially time-consuming.

C.2 DIRECTORY KALM

This directory contains the DOS program Kalm 'N Smooth that was included with the previous edition of this text. The version is 1.4, which is the final version of this software package. Kalm 'N Smooth is a menu-oriented program for use in Kalman filtering error-covariance analysis and Monte Carlo simulation. Some utility programs supporting the main programs are also included. Its use is quite different from the MATLAB approach, though, because it is menu-oriented. The user is simply asked to follow instructions at a sequence of prompts, rather than to write executable lines of code.

The user's guide for Kalm 'N Smooth is in the READ.ME file in the KALM directory. All the necessary information for using Kalm 'N Smooth is given in this file. The READ.ME file is fairly lengthy, so it is suggested that the user print out a hard copy of it for easy reference. (The READ.ME file is an ASCII file.)

Kalm 'N Smooth is a very specialized software package, so it is not as versatile as MATLAB. However, it can also be used to solve many of the Kalman filter problems in this book. One way that Kalm 'N Smooth can be used to advantage is to use it as a means of verifying the MATLAB solutions for problems where this is possible. Working a problem by two independent means and getting the same answer certainly adds to one's confidence in the answer.

C.3 FINAL COMMENT ABOUT THE SOFTWARE

There is no guarantee that the software included in the diskette will be compatible with all machines, present and future. In particular, the MATLAB M-files were developed to run on both Versions 3.5 and 4 as nearly as possible.

Where there were conflicts, they were resolved in favor of Version 4. As a result, there are a few problem solutions where the matrix size limitation is exceeded for those using the earlier Student Version 3.5 (circa 1992). Comment statements are provided in these problem M-files which suggest how the code should be modified to mitigate the matrix-size difficulty (and, in some cases, this results in a slightly different version of the original problem). The moral to all this is that there is no substitute for reading the solution M-file and understanding the step-by-step tasks that the code is intended to implement. Once this is understood, there should be no difficulty in making whatever modifications that may be necessary to make the programs play on your machine.

Index