CHAPTER

7

Kalman Filters

In this chapter we complete our study of linear optimum filters by developing the basic ideas of *Kalman filtering*. A distinctive feature of a Kalman filter is that its mathematical formulation is described in terms of *state-space concepts*. Another novel feature of a Kalman filter is that its solution is computed *recursively*. In particular, each updated estimate of the state is computed from the previous estimate and the new input data, so only the previous estimate requires storage. In addition to eliminating the need for storing the entire past observed data, a Kalman filter is computationally more efficient than computing the estimate directly from the entire past observed data at each step of the filtering process. The Kalman filter is thus ideally suited for implementation on a digital computer. Most importantly, it has been applied successfully to many practical problems in diverse fields, particularly in aerospace and aeronautical applications.

Our interest in the Kalman filter is motivated by the fact that it provides a unifying framework for the derivation of an important family of adaptive filters known as recursive least-squares filters, as demonstrated in subsequent chapters of the book. To pave the way for the development of the Kalman filter, we begin by solving the *recursive minimum mean-squared estimation problem* for the simple case of scalar random variables. For this solution, we use the *innovations approach* that exploits the correlation properties of a special stochastic process known as the *innovations process* (Kailath, 1968, 1970).

7.1 RECURSIVE MINIMUM MEAN-SQUARE ESTIMATION FOR SCALAR RANDOM VARIABLES

Let us assume that, based on a complete set of observed random variables $y(1), y(2), \ldots, y(n-1)$, starting with the first observation at time 1 and extending up to and including time $n-1$, we have found the minimum mean-square estimate $\hat{x}(n-1|\mathcal{Y}_{n-1})$ of a related zero-mean random variable $x(n-1)$. We are assuming that the observation at (or before) $n=0$ is zero. The space spanned by the observations $y(1), \ldots, y(n-1)$ is denoted by $\mathcal{Y}_{n-1}$. Suppose that we now have an additional observation $y(n)$ at time n, and the requirement is to compute an *updated* estimate $\hat{x}(n|\mathcal{Y}_n)$ of the related random variable $x(n)$, where $\mathcal{Y}_n$ denotes the space spanned by $y(1), \ldots, y(n)$. We may do this computation by storing the *past* observations, $y(1), y(2), \ldots, y(n-1)$, and then redoing the whole problem with the available data $y(1), y(2), \ldots, y(n-1), y(n)$, including the new observation. Computationally, however, it is much more efficient to use a *recursive estimation procedure*. In this procedure we *store* the previous estimate $\hat{x}(n-1|\mathcal{Y}_{n-1})$ and exploit it to compute the updated estimate $\hat{x}(n|\mathcal{Y}_n)$ in light of the new observation $y(n)$. There are several ways of developing the algorithm to do this recursive estimation. We will use the notion of *innovations* (Kailath, 1968, 1970), the origin of which may be traced back to Kolmogorov (1941).

Define the forward prediction error

$$f_{n-1}(n) = y(n) - \hat{y}(n|\mathcal{Y}_{n-1}), \quad n = 1, 2, \ldots \tag{7.1}$$

where $\hat{y}(n|\mathcal{Y}_{n-1})$ is the *one-step prediction* of the observed random variable $y(n)$ at time n, using *all* past observations available up to and including time $n-1$. The past observations used in this estimation are $y(1), y(2), \ldots, y(n-1)$, so the order of the prediction equals $n-1$. We may view $f_{n-1}(n)$ as the output of a forward prediction-error filter of order $n-1$, and with the filter input fed by the time series $y(1), y(2), \ldots, y(n)$. Note that the *prediction order* $n-1$ *increases linearly with* n. According to the principle of orthogonality, the prediction error $f_{n-1}(n)$ is orthogonal to all past observations $y(1), y(2), \ldots, y(n-1)$ and may therefore be regarded as a *measure* of the new information in the random variable $y(n)$ observed at time n, hence the name "innovation." The fact is that the observation $y(n)$ does not itself convey completely new information, since the predictable part, $\hat{y}(n|\mathcal{Y}_{n-1})$, is already completely determined by the past observations $y(1), y(2), \ldots, y(n-1)$. Rather, the part of the observation $y(n)$ that is really new is contained in the forward prediction error $f_{n-1}(n)$. We may therefore refer to this prediction error as the *innovation*, and for simplicity of notation write

$$\alpha(n) = f_{n-1}(n), \quad n = 1, 2, \ldots \tag{7.2}$$

The innovation $\alpha(n)$ has several important properties, as described here.

Property 1. *The innovation $\alpha(n)$, associated with the observed random variable $y(n)$, is orthogonal to the past observations $y(1), y(2), \ldots, y(n-1)$, as shown by*

$$E[\alpha(n)y^*(k)] = 0, \quad 1 \le k \le n-1 \tag{7.3}$$

This is simply a restatement of the principle of orthogonality.

Property 2. *The innovations* $\alpha(1), \alpha(2), \ldots, \alpha(n)$ *are orthogonal to each other, as shown by*

$$E[\alpha(n)\alpha^*(k)] = 0, \quad 1 \leq k \leq n-1 \tag{7.4}$$

This is a restatement of the fact that [see part (e) of Problem 20, Chapter 6]:

$$E[f_{n-1}(n)f^*_{k-1}(k)] = 0, \quad 1 \leq k \leq n-1$$

Equation (7.4), in effect, states that the innovation process $\alpha(n)$, described by Eqs. (7.1) and (7.2), is *white*.

Property 3. *There is a one-to-one correspondence between the observed data* $\{y(1), y(2), \ldots, y(n)\}$ *and the innovations* $\{\alpha(1), \alpha(2), \ldots, \alpha(n)\}$, *in that the one sequence may be obtained from the other by means of a causal and causally invertible filter without any loss of information. We may thus write*

$$\{y(1), y(2), \ldots, y(n)\} \rightleftharpoons \{\alpha(1), \alpha(2), \ldots, \alpha(n)\} \tag{7.5}$$

To prove this property, we use a form of the Gram–Schmidt orthogonalization procedure (described in Chapter 6). The procedure assumes that the observations $y(1), y(2), \ldots, y(n)$ are linearly independent in an algebraic sense. We first put

$$\alpha(1) = y(1) \tag{7.6}$$

where it is assumed that $\hat{y}(1|\mathcal{Y}_0)$ is zero. Next we put

$$\alpha(2) = y(2) + a_{1,1}y(1) \tag{7.7}$$

The coefficient $a_{1,1}$ is chosen such that the innovations $\alpha(1)$ and $\alpha(2)$ are orthogonal, as shown by

$$E[\alpha(2)\alpha^*(1)] = 0 \tag{7.8}$$

This requirement is satisfied by choosing

$$a_{1,1} = -\frac{E[y(2)y^*(1)]}{E[y(1)y^*(1)]} \tag{7.9}$$

Except for the minus sign, $a_{1,1}$ is a partial correlation coefficient in that it equals the cross-correlation between the observations $y(2)$ and $y(1)$, normalized with respect to the mean-square value of $y(1)$.

Next, we put

$$\alpha(3) = y(3) + a_{2,1}y(2) + a_{2,2}y(1) \tag{7.10}$$

where the coefficients $a_{2,1}$ and $a_{2,2}$ are chosen such that $\alpha(3)$ is orthogonal to both $\alpha(1)$ and $\alpha(2)$, and so on. Thus, in general, we may express the transformation of the observed data $y(1), y(2), \ldots, y(n)$ into the innovations $\alpha(1), \alpha(2), \ldots, \alpha(n)$ by writing

$$\begin{bmatrix} \alpha(1) \\ \alpha(2) \\ \cdot \\ \cdot \\ \cdot \\ \alpha(n) \end{bmatrix} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ a_{1,1} & 1 & \cdots & 0 \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ a_{n-1,n-1} & a_{n-1,n-2} & \cdots & 1 \end{bmatrix} \begin{bmatrix} y(1) \\ y(2) \\ \cdot \\ \cdot \\ \cdot \\ y(n) \end{bmatrix} \tag{7.11}$$

The nonzero elements of row k of the *lower triangular transformation matrix* on the right-hand side of Eq. (7.11) are deliberately denoted as $a_{k-1,k-1}, a_{k-1,k-2} \ldots, 1$, where $k = 1, 2, \ldots, n$. These elements represent the coefficients of a *forward prediction-error filter* of order $k - 1$. Note that $a_{k,0} = 1$ for all k. Accordingly, given the observed data $y(1), y(2), \ldots, y(n)$, we may compute the innovations $\alpha(1), \alpha(2), \ldots, \alpha(n)$. There is no loss of information in the course of this transformation, since we may recover the original observed data $y(1), y(2), \ldots, y(n)$ from the innovations $\alpha(1), \alpha(2), \ldots, \alpha(n)$. This we do by premultiplying both sides of Eq. (7.11) by the inverse of the lower triangular transformation matrix. This matrix is nonsingular, since its determinant equals 1 for all n. The transformation is therefore reversible.

Using Eq. (7.5), we may thus write

$$\hat{x}(n|\mathcal{Y}_n) = \text{minimum mean-square estimate of } x(n) \text{ given the observed data } y(1), y(2), \ldots, y(n)$$

or, equivalently,

$$\hat{x}(n|\mathcal{Y}_n) = \text{minimum mean-square estimate of } x(n) \text{ given the innovations } \alpha(1), \alpha(2), \ldots, \alpha(n)$$

Define the estimate $\hat{x}(n|\mathcal{Y}_n)$ as a linear combination of the innovations $\alpha(1), \alpha(2), \ldots, \alpha(n)$:

$$\hat{x}(n|\mathcal{Y}_n) = \sum_{k=1}^{n} b_k \alpha(k) \tag{7.12}$$

where the b_k are to be determined. With the innovations $\alpha(1), \alpha(2), \ldots, \alpha(n)$ orthogonal to each other, and the b_k chosen to minimize the mean-square value of the estimation error $x(n) - \hat{x}(n|\mathcal{Y}_n)$, we find that

$$b_k = \frac{E[x(n)\alpha^*(k)]}{E[\alpha(k)\alpha^*(k)]}, \quad 1 \le k \le n \tag{7.13}$$

We rewrite Eq. (7.12) in the form

$$\hat{x}(n|\mathcal{Y}_n) = \sum_{k=0}^{n-1} b_k \alpha(k) + b_n \alpha(n) \tag{7.14}$$

where

$$b_n = \frac{E[x(n)\alpha^*(n)]}{E[\alpha(n)\alpha^*(n)]} \tag{7.15}$$

However, by definition, the summation term on the right-hand side of Eq. (7.14) equals the previous estimate $\hat{x}(n-1|\mathcal{Y}_{n-1})$. We may thus express the recursive estimation algorithm that we are seeking as

$$\hat{x}(n|\mathcal{Y}_n) = \hat{x}(n-1|\mathcal{Y}_{n-1}) + b_n\alpha(n) \tag{7.16}$$

where b_n is defined by Eq. (7.15). Thus, by adding a *correction term* $b_n\alpha(n)$ to the previous estimate $\hat{x}(n-1|\mathcal{Y}_{n-1})$, with the correction being proportional to the innovation $\alpha(n)$, we get the updated estimate $\hat{x}(n|\mathcal{Y}_n)$.

The simple formulas of Eq. (7.15) and (7.16) are the basis of all recursive linear estimation schemes. Equipped with these simple and yet powerful ideas, we are now ready to study the more general Kalman filtering problem.

7.2 STATEMENT OF THE KALMAN FILTERING PROBLEM

Consider a *linear, discrete-time dynamical system* described by the signal-flow graph shown in Fig. 7.1. The time-domain description of the system presented here offers the following advantages (Gelb, 1974):

- Mathematical and notational convenience
- Close relationship to physical reality
- Useful basis for accounting for statistical behavior of the system

The notation of *state* plays a key role in this formulation. The *state vector*, denoted by $\mathbf{x}(n)$ in Fig. 7.1, is defined as any set of quantities that would be sufficient to uniquely describe the unforced dynamical behavior of the system. Typically, the state vector $\mathbf{x}(n)$, assumed to be of dimension M, is unknown. To estimate it, we use a set of observed data, denoted by the vector $\mathbf{y}(n)$ in Fig. 7.1. The *observation vector* $\mathbf{y}(n)$ is assumed to be of dimension N.

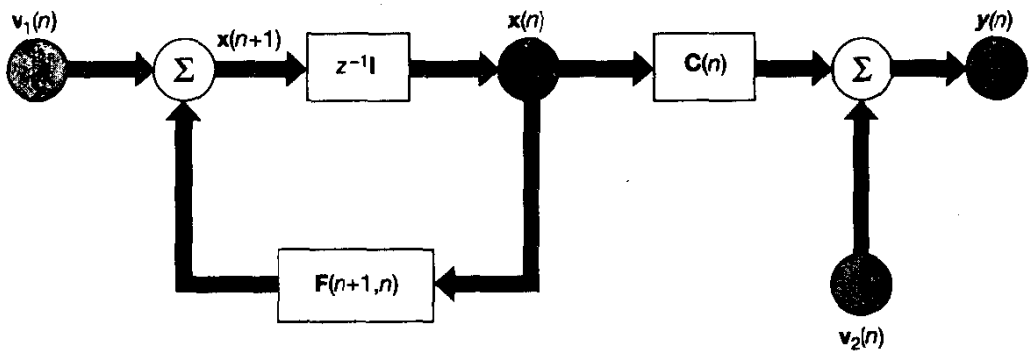

Figure 7.1 Signal-flow graph representation of a linear, discrete-time dynamical system.

In mathematical terms, the signal-flow graph of Fig. 7.1 embodies the following pair of equations:

1. A *process equation*

$$\mathbf{x}(n+1) = \mathbf{F}(n+1, n)\mathbf{x}(n) + \mathbf{v}_1(n) \tag{7.17}$$

where $\mathbf{F}(n+1, n)$ is a known M-by-M *state transition matrix* relating the state of the system at times $n+1$ and n. The M-by-1 vector $\mathbf{v}_1(n)$ represents *process noise*. The vector $\mathbf{v}_1(n)$ is modeled as a zero-mean, white-noise process whose correlation matrix is defined by

$$E[\mathbf{v}_1(n)\mathbf{v}_1^H(k)] = \begin{cases} \mathbf{Q}_1(n), & n = k \\ \mathbf{0}, & n \neq k \end{cases} \tag{7.18}$$

2. A *measurement equation*, describing the observation vector as

$$\mathbf{y}(n) = \mathbf{C}(n)\mathbf{x}(n) + \mathbf{v}_2(n) \tag{7.19}$$

where $\mathbf{C}(n)$ is a known N-by-M *measurement matrix*. The N-by-1 vector $\mathbf{v}_2(n)$ is called *measurement noise*. It is modeled as a zero-mean, white-noise process whose correlation matrix is

$$E[\mathbf{v}_2(n)\mathbf{v}_2^H(k)] = \begin{cases} \mathbf{Q}_2(n), & n = k \\ \mathbf{0}, & n \neq k \end{cases} \tag{7.20}$$

It is assumed that $\mathbf{x}(0)$, the initial value of the state, is uncorrelated with both $\mathbf{v}_1(n)$ and $\mathbf{v}_2(n)$ for $n \geq 0$. The noise vectors $\mathbf{v}_1(n)$ and $\mathbf{v}_2(n)$ are statistically independent, so we may write

$$E[\mathbf{v}_1(n)\,\mathbf{v}_2^H(k)] = \mathbf{0} \qquad \text{for all } n \text{ and } k \tag{7.21}$$

The Kalman filtering problem may now be formally stated as follows: Use the entire observed data, consisting of the vectors $\mathbf{y}(1), \mathbf{y}(2), \ldots, \mathbf{y}(n)$, to find for each $n \geq 1$ the minimum mean-square estimates of the components of the state $\mathbf{x}(i)$. The problem is called the *filtering* problem if $i = n$, the *prediction* problem if $i > n$, and the *smoothing* problem if $1 \leq i < n$. In this chapter we will only be concerned with the filtering and prediction problems, which are closely related. As remarked earlier in the introduction, we will solve the Kalman filtering problem by using the innovations approach (Kailath, 1968, 1970, 1981; Tretter, 1976).

7.3 THE INNOVATIONS PROCESS

Let the vector $\hat{\mathbf{y}}(n|\mathcal{Y}_{n-1})$ denote the minimum mean-square estimate of the observed data $\mathbf{y}(n)$ at time n, given all the past values of the observed data starting at time $n = 1$ and extending up to and including time $n - 1$. These past values are represented by the vec-

tors $\mathbf{y}(1), \mathbf{y}(2), \ldots, \mathbf{y}(n-1)$, which span the vector space $\mathcal{Y}_{n-1}$. We define the *innovations process* associated with $\mathbf{y}(n)$ as

$$\boldsymbol{\alpha}(n) = \mathbf{y}(n) - \hat{\mathbf{y}}(n|\mathcal{Y}_{n-1}), \quad n = 1, 2, \ldots \tag{7.22}$$

The M-by-1 vector $\boldsymbol{\alpha}(n)$ represents the new information in the observed data $\mathbf{y}(n)$.

Generalizing the results of Eqs. (7.3), (7.4) and (7.5), we find that the innovations process $\boldsymbol{\alpha}(n)$ has the following properties:

1. The innovations process $\boldsymbol{\alpha}(n)$, associated with the observed data $\mathbf{y}(n)$ at time n, is orthogonal to all past observations $\mathbf{y}(1), \mathbf{y}(2), \ldots, \mathbf{y}(n-1)$ as shown by

$$E[\boldsymbol{\alpha}(n)\mathbf{y}^H(k)] = \mathbf{0}, \quad 1 \leq k \leq n-1 \tag{7.23}$$

2. The innovations process consists of a sequence of vector random variables that are orthogonal to each other, as shown by

$$E[\boldsymbol{\alpha}(n)\boldsymbol{\alpha}^H(k)] = \mathbf{0}, \quad 1 \leq k \leq n-1 \tag{7.24}$$

3. There is a one-to-one correspondence between the sequence of vector random variables $\{\mathbf{y}(1), \mathbf{y}(2), \ldots, \mathbf{y}(n)\}$ representing the observed data and the sequence of vector random variables $\{\boldsymbol{\alpha}(1), \boldsymbol{\alpha}(2), \ldots, \boldsymbol{\alpha}(n)\}$ representing the innovations process, in that the one sequence may be obtained from the other by means of linear stable operators without loss of information. Thus, we may state that

$$\{\mathbf{y}(1), \mathbf{y}(2), \ldots, \mathbf{y}(n)\} \rightleftharpoons \{\boldsymbol{\alpha}(1), \boldsymbol{\alpha}(2), \ldots, \boldsymbol{\alpha}(n)\} \tag{7.25}$$

To form the sequence of vector random variables defining the innovations process, we may use a Gram–Schmidt orthogonalization procedure similar to that described in Section 7.1, except that the procedure is now formulated in terms of vectors and matrices (see Problem 1).

Correlation Matrix of the Innovations Process

To determine the correlation matrix of the innovations process $\boldsymbol{\alpha}(n)$, we first solve the state equation (7.17) recursively to obtain

$$\mathbf{x}(k) = \mathbf{F}(k, 0)\mathbf{x}(0) + \sum_{i=1}^{k-1} \mathbf{F}(k, i+1)\mathbf{v}_1(i) \tag{7.26}$$

where we have made use of the following assumptions and properties:

1. The initial value of the state vector $\mathbf{x}(0)$.
2. As previously assumed, the observed data [and therefore the noise vector $\mathbf{v}_1(n)$] are zero for $n \leq 0$.

3. The state transition matrix has the properties

$$\mathbf{F}(k, k-1)\mathbf{F}(k-1, k-2)\ldots\mathbf{F}(i+1, i) = \mathbf{F}(k, i)$$

and

$$\mathbf{F}(k, k) = \mathbf{I}$$

where $\mathbf{I}$ is the identity matrix. Note that for a time-invariant system we have

$$\mathbf{F}(n+1, n) = \mathbf{F}(n+1-n) = \mathbf{F}(1) = \text{constant}.$$

Equation (7.26) shows that $\mathbf{x}(k)$ is a linear combination of $\mathbf{x}(0)$ and $\mathbf{v}_1(1), \mathbf{v}_1(2), \ldots, \mathbf{v}_1(k-1)$.

By hypothesis, the measurement noise vector $\mathbf{v}_2(n)$ is uncorrelated with both the initial state vector $\mathbf{x}(0)$ and the process noise vector $\mathbf{v}_1(n)$. Accordingly, premultiplying both sides of Eq. (7.26) by $\mathbf{v}_2^H(n)$, and taking expectations, we deduce that

$$E[\mathbf{x}(k)\mathbf{v}_2^H(n)] = \mathbf{O}, \qquad k, n \leq 0 \tag{7.27}$$

Correspondingly, we deduce from the measurement equation (7.19) that

$$E[\mathbf{y}(k)\mathbf{v}_2^H(n)] = \mathbf{O}, \qquad 0 \leq k \leq n-1 \tag{7.28}$$

Moreover, we may write

$$E[\mathbf{y}(k)\mathbf{v}_1^H(n)] = \mathbf{O}, \qquad 0 \leq k \leq n \tag{7.29}$$

Given the past observations $\mathbf{y}(1), \ldots, \mathbf{y}(n-1)$ that span the space $\mathcal{Y}_{n-1}$, we also find from the measurement equation (7.19) that the minimum mean-square estimate of the present value $\mathbf{y}(n)$ of the observation vector equals

$$\hat{\mathbf{y}}(n|\mathcal{Y}_{n-1}) = \mathbf{C}(n)\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1}) + \hat{\mathbf{v}}_2(n|\mathcal{Y}_{n-1})$$

However, the estimate $\hat{\mathbf{v}}_2(n|\mathcal{Y}_{n-1})$ of the measurement noise vector is zero since $\mathbf{v}_2(n)$ is orthogonal to the past observations $\mathbf{y}(1), \ldots, \mathbf{y}(n-1)$; see Eq. (7.28). Hence, we may simply write

$$\hat{\mathbf{y}}(n|\mathcal{Y}_{n-1}) = \mathbf{C}(n)\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1}) \tag{7.30}$$

Therefore, using Eqs. (7.22) and (7.30), we may express the innovations process in the form

$$\boldsymbol{\alpha}(n) = \mathbf{y}(n) - \mathbf{C}(n)\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1}) \tag{7.31}$$

Substituting the measurement equation (7.19) in (7.31), we get

$$\boldsymbol{\alpha}(n) = \mathbf{C}(n)\boldsymbol{\epsilon}(n, n-1) + \mathbf{v}_2(n) \tag{7.32}$$

where $\boldsymbol{\epsilon}(n, n-1)$ is the *predicted state-error vector* at time n, using data up to time

$n-1$. That is, $\boldsymbol{\epsilon}(n, n-1)$ is the difference between the state vector $\mathbf{x}(n)$ and the one-step prediction vector $\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})$, as shown by

$$\boldsymbol{\epsilon}(n, n-1) = \mathbf{x}(n) - \hat{\mathbf{x}}(n|\mathcal{Y}_{n-1}) \tag{7.33}$$

Note that the predicted state-error vector is orthogonal to both the process noise vector $\mathbf{v}_1(n)$ and the measurement noise vector $\mathbf{v}_2(n)$; see Problem 2.

The correlation matrix of the innovations process $\boldsymbol{\alpha}(n)$ is defined by

$$\mathbf{R}(n) = E[\boldsymbol{\alpha}(n)\boldsymbol{\alpha}^H(n)] \tag{7.34}$$

Therefore, substituting Eq. (7.32) in (7.34), expanding the pertinent terms, and then using the fact that the vectors $\boldsymbol{\epsilon}(n, n-1)$ and $\mathbf{v}_2(n)$ are orthogonal, we obtain the result:

$$\mathbf{R}(n) = \mathbf{C}(n)\mathbf{K}(n, n-1)\mathbf{C}^H(n) + \mathbf{Q}_2(n) \tag{7.35}$$

where $\mathbf{Q}_2(n)$ is the correlation matrix of the noise vector $\mathbf{v}_2(n)$. The M-by-M matrix $\mathbf{K}(n, n-1)$ is called the *predicted state-error correlation matrix*; it is defined by

$$\mathbf{K}(n, n-1) = E[\boldsymbol{\epsilon}(n, n-1)\boldsymbol{\epsilon}^H(n, n-1)] \tag{7.36}$$

where $\boldsymbol{\epsilon}(n, n-1)$ is the predicted state-error vector. The matrix $\mathbf{K}(n, n-1)$ is used as the statistical description of the error in the predicted estimate $\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})$.

7.4 ESTIMATION OF THE STATE USING THE INNOVATIONS PROCESS

Consider next the problem of deriving the minimum mean-square estimate of the state $\mathbf{x}(i)$ from the innovations process. From the discussion presented in Section 7.1, we deduce that this estimate may be expressed as a linear combination of the sequence of innovations processes $\boldsymbol{\alpha}(1), \boldsymbol{\alpha}(2), \ldots, \boldsymbol{\alpha}(n)$ [see Eq. (7.12) for comparison]:

$$\hat{\mathbf{x}}(i|\mathcal{Y}_n) = \sum_{k=1}^{n} \mathbf{B}_i(k)\boldsymbol{\alpha}(k) \tag{7.37}$$

where $\mathbf{B}_i(k)$, $k = 1, 2, \ldots, n$, is a set of M-by-N matrices to be determined. According to the principle of orthogonality, the predicted state-error vector is orthogonal to the innovation process, as shown by

$$\begin{aligned} E[\boldsymbol{\epsilon}(i, n)\boldsymbol{\alpha}^H(m)] &= E\{[\mathbf{x}(i) - \hat{\mathbf{x}}(i|\mathcal{Y}_n)]\boldsymbol{\alpha}^H(m)\} \\ &= \mathbf{O}, \qquad m = 1, 2, \ldots, n \end{aligned} \tag{7.38}$$

Substituting Eq. (7.37) in (7.38) and using the orthogonality property of the innovations process, namely, Eq. (7.24), we get

$$\begin{aligned} E[\mathbf{x}(i)\boldsymbol{\alpha}^H(m)] &= \mathbf{B}_i(m)E[\boldsymbol{\alpha}(m)\boldsymbol{\alpha}^H(m)] \\ &= \mathbf{B}_i(m)\mathbf{R}(m) \end{aligned} \tag{7.39}$$

Hence, postmultiplying both sides of Eq. (7.39) by the inverse matrix $\mathbf{R}^{-1}(m)$, we find that $\mathbf{B}_i(m)$ is given by

$$\mathbf{B}_i(m) = E[\mathbf{x}(i)\boldsymbol{\alpha}^H(m)]\,\mathbf{R}^{-1}(m) \tag{7.40}$$

Finally, substituting Eq. (7.40) in (7.37), we get the minimum mean-square estimate

$$\begin{aligned}\hat{\mathbf{x}}(i|\mathcal{Y}_n) &= \sum_{k=1}^{n} E[\mathbf{x}(i)\boldsymbol{\alpha}^H(k)]\,\mathbf{R}^{-1}(k)\boldsymbol{\alpha}(k)\\ &= \sum_{k=1}^{n-1} E[\mathbf{x}(i)\boldsymbol{\alpha}^H(k)]\,\mathbf{R}^{-1}(k)\,\boldsymbol{\alpha}(k)\\ &\quad + E[\mathbf{x}(i)\boldsymbol{\alpha}^H(n)]\,\mathbf{R}^{-1}(n)\boldsymbol{\alpha}(n)\end{aligned}$$

For $i = n + 1$, we may therefore write

$$\begin{aligned}\hat{\mathbf{x}}(n+1|\mathcal{Y}_n) &= \sum_{k=1}^{n-1} E[\mathbf{x}(n+1)\boldsymbol{\alpha}^H(k)]\,\mathbf{R}^{-1}(k)\boldsymbol{\alpha}(k)\\ &\quad + E[\mathbf{x}(n+1)\boldsymbol{\alpha}^H(n)]\,\mathbf{R}^{-1}(n)\boldsymbol{\alpha}(n)\end{aligned} \tag{7.41}$$

However, the state $\mathbf{x}(n + 1)$ at time $n + 1$ is related to the state $\mathbf{x}(n)$ at time n by Eq. (7.17). Therefore, using this relation, we may write for $0 \leq k \leq n$:

$$\begin{aligned}E[\mathbf{x}(n+1)\boldsymbol{\alpha}^H(k)] &= E\{[\mathbf{F}(n+1,n)\mathbf{x}(n) + \mathbf{v}_1(n)]\boldsymbol{\alpha}^H(k)\}\\ &= \mathbf{F}(n+1,n)E[\mathbf{x}(n)\boldsymbol{\alpha}^H(k)]\end{aligned} \tag{7.42}$$

where we have made use of the fact that $\boldsymbol{\alpha}(k)$ depends only on the observed data $\mathbf{y}(1), \ldots, \mathbf{y}(k)$, and therefore from Eq. (7.29) we see that $\mathbf{y}(n)$ and $\boldsymbol{\alpha}(k)$ are orthogonal for $0 \leq k \leq n$. We may thus rewrite the summation term on the right-hand side of Eq. (7.41) as follows:

$$\begin{aligned}\sum_{k=1}^{n-1} E[\mathbf{x}(n+1)\boldsymbol{\alpha}^H(k)]\,\mathbf{R}^{-1}(k)\boldsymbol{\alpha}(k) &= \mathbf{F}(n+1,n)\sum_{k=1}^{n-1} E[\mathbf{x}(n)\boldsymbol{\alpha}^H(k)]\,\mathbf{R}^{-1}(k)\boldsymbol{\alpha}(k)\\ &= \mathbf{F}(n+1,n)\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})\end{aligned} \tag{7.43}$$

To proceed further, we introduce some basic definitions, as described next.

Kalman Gain

Define the M-by-N matrix:

$$\mathbf{G}(n) = E[\mathbf{x}(n+1)\boldsymbol{\alpha}^H(n)]\,\mathbf{R}^{-1}(n) \tag{7.44}$$

Then, using this definition and the result of Eq. (7.43), we may rewrite Eq. (7.41) as follows:

$$\hat{\mathbf{x}}(n+1|\mathcal{Y}_n) = \mathbf{F}(n+1,n)\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1}) + \mathbf{G}(n)\boldsymbol{\alpha}(n) \tag{7.45}$$

Equation (7.45) is of fundamental significance. It shows that we may compute the minimum mean-square estimate $\hat{\mathbf{x}}(n+1|\mathcal{Y}_n)$ of the state of a linear dynamical system by adding to the previous estimate $\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})$, which is premultiplied by the state transition matrix $\mathbf{F}(n+1, n)$, a correction term equal to $\mathbf{G}(n)\boldsymbol{\alpha}(n)$. The correction term equals the innovations process $\boldsymbol{\alpha}(n)$ premultiplied by the matrix $\mathbf{G}(n)$. Accordingly, and in recognition of the pioneering work by Kalman, the matrix $\mathbf{G}(n)$ is called the *Kalman gain*.

There now remains only the problem of expressing the Kalman gain $\mathbf{G}(n)$ in a form convenient for computation. To do this, we first use Eqs. (7.32) and (7.42) to express the expectation of the product of $\mathbf{x}(n+1)$ and $\boldsymbol{\alpha}^H(n)$ as follows:

$$\begin{aligned} E[\mathbf{x}(n+1)\boldsymbol{\alpha}^H(n)] &= \mathbf{F}(n+1, n)E[\mathbf{x}(n)\boldsymbol{\alpha}^H(n)] \\ &= \mathbf{F}(n+1, n)E[\mathbf{x}(n)(\mathbf{C}(n)\boldsymbol{\epsilon}(n, n-1) + \mathbf{v}_2(n))^H] \\ &= \mathbf{F}(n+1, n)E[\mathbf{x}(n)\boldsymbol{\epsilon}^H(n,n-1)]\mathbf{C}^H(n) \end{aligned} \tag{7.46}$$

where we have used the fact that the state $\mathbf{x}(n)$ and noise vector $\mathbf{v}_2(n)$ are uncorrelated [see Eq. (7.27)]. We further note that the predicted state-error vector $\boldsymbol{\epsilon}(n, n-1)$ is orthogonal to the estimate $\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})$. Therefore, the expectation of the product of $\mathbf{x}(n|\mathcal{Y}_{n-1})$ and $\boldsymbol{\epsilon}^H(n, n-1)$ is zero, and so we may rewrite Eq. (7.46) by replacing the multiplying factor $\mathbf{x}(n)$ by the predicted state-error vector $\boldsymbol{\epsilon}(n, n-1)$ as follows:

$$E[\mathbf{x}(n+1)\boldsymbol{\alpha}^H(n)] = \mathbf{F}(n+1, n)E[\boldsymbol{\epsilon}(n, n-1)\boldsymbol{\epsilon}^H(n, n-1)]\mathbf{C}^H(n) \tag{7.47}$$

From Eq. (7.36), we see that the expectation on the right-hand side of Eq. (7.47) equals the predicted state-error correlation matrix. Hence, we may rewrite Eq. (7.47) as follows:

$$E[\mathbf{x}(n+1)\boldsymbol{\alpha}^H(n)] = \mathbf{F}(n+1,n)\mathbf{K}(n, n-1)\mathbf{C}^H(n) \tag{7.48}$$

We may now redefine the Kalman gain. In particular, substituting Eq. (7.48) in (7.44), we get

$$\mathbf{G}(n) = \mathbf{F}(n+1, n)\mathbf{K}(n, n-1)\mathbf{C}^H(n)\mathbf{R}^{-1}(n) \tag{7.49}$$

where the correlation matrix $\mathbf{R}(n)$ is itself defined in Eq. (7.35).

The block diagram of Fig. 7.2 shows the signal-flow graph representation of Eq. (7.49) for computing the Kalman gain $\mathbf{G}(n)$. Having computed the Kalman gain $\mathbf{G}(n)$, we may then use Eq. (7.45) to update the one-step prediction, that is, to compute $\hat{\mathbf{x}}(n+1|\mathcal{Y}_n)$ given its old value $\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})$, as illustrated in Fig. 7.3. In this figure we have also used Eq. (7.31) for the innovations process $\boldsymbol{\alpha}(n)$.

Riccati Equation

As it stands, the formula of Eq. (7.49) is not particularly useful for computing the Kalman gain $\mathbf{G}(n)$, since it requires that the predicted state-error correlation matrix $\mathbf{K}(n, n-1)$ be known. To overcome this difficulty, we derive a formula for the recursive computation of $\mathbf{K}(n, n-1)$.

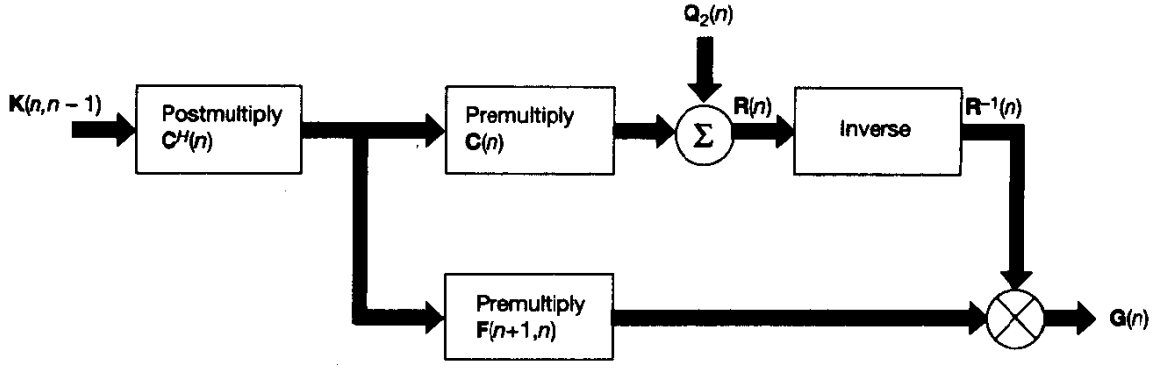

Figure 7.2 Kalman gain computer.

The predicted state-error vector $\boldsymbol{\epsilon}(n+1, n)$ equals the difference between the state $\mathbf{x}(n+1)$ and the one-step prediction $\hat{\mathbf{x}}(n+1|\mathcal{Y}_n)$[see Eq. (7.33)]:

$$\boldsymbol{\epsilon}(n+1, n) = \mathbf{x}(n+1) - \hat{\mathbf{x}}(n+1|\mathcal{Y}_n) \tag{7.50}$$

Substituting Eqs. (7.17) and (7.45) in (7.50), and using Eq. (7.31) for the innovations process $\boldsymbol{\alpha}(n)$, we get

$$\begin{aligned}\boldsymbol{\epsilon}(n+1, n) = \mathbf{F}(n+1, n)[\mathbf{x}(n) - \hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})] \\ - \mathbf{G}(n)\,[\mathbf{y}(n) - \mathbf{C}(n)\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})] + \mathbf{v}_1(n)\end{aligned} \tag{7.51}$$

Next, using the measurement equation (7.19) to eliminate $\mathbf{y}(n)$ in Eq. (7.51), we get the following difference equation for recursive computation of the predicted state-error vector:

$$\begin{aligned}\boldsymbol{\epsilon}(n+1, n) = [\mathbf{F}(n+1, n) - \mathbf{G}(n)\mathbf{C}(n)]\,\boldsymbol{\epsilon}(n, n-1) \\ + \mathbf{v}_1(n) - \mathbf{G}(n)\,\mathbf{v}_2(n)\end{aligned} \tag{7.52}$$

The correlation matrix of the predicted state-error vector $\boldsymbol{\epsilon}(n+1, n)$ equals [see Eq. (7.36)]

$$\mathbf{K}(n+1, n) = E[\boldsymbol{\epsilon}(n+1, n)\boldsymbol{\epsilon}^H(n+1, n)] \tag{7.53}$$

Substituting Eq. (7.52) in (7.53), and recognizing that the error vector $\boldsymbol{\epsilon}(n, n-1)$ and the noise vectors $\mathbf{v}_1(n)$ and $\mathbf{v}_2(n)$ are mutually uncorrelated, we may express the predicted state-error correlation matrix as follows:

$$\begin{aligned}\mathbf{K}(n+1, n) = [\mathbf{F}(n+1, n) - \mathbf{G}(n)\mathbf{C}(n)]\mathbf{K}(n, n-1)\,[\mathbf{F}(n+1, n) - \mathbf{G}(n)\mathbf{C}(n)]^H \\ + \mathbf{Q}_1(n) + \mathbf{G}(n)\mathbf{Q}_2(n)\mathbf{G}^H(n)\end{aligned} \tag{7.54}$$

where $\mathbf{Q}_1(n)$ and $\mathbf{Q}_2(n)$ are the correlation matrices of $\mathbf{v}_1(n)$ and $\mathbf{v}_2(n)$, respectively. By expanding the right-hand side of Eq. (7.54), and then using Eqs. (7.49) and (7.35) for the

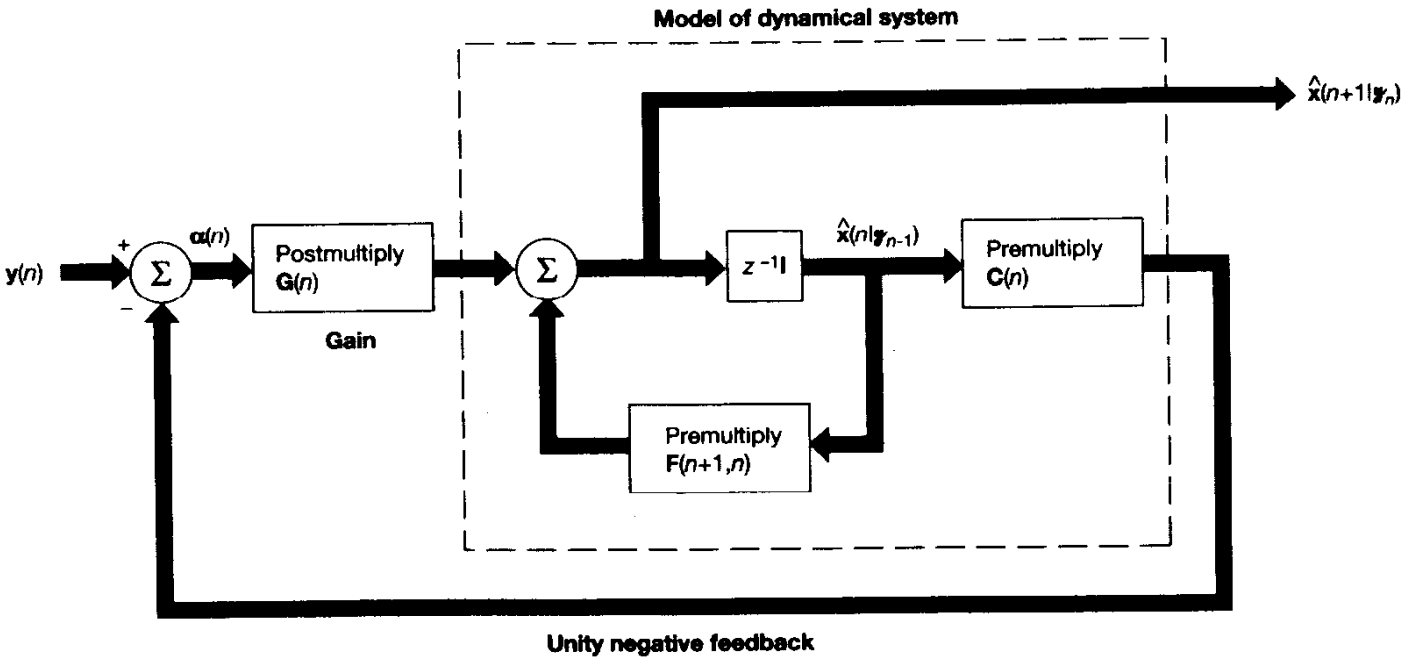

Figure 7.3 One-step predictor.

Kalman gain, we get the *Riccati difference equation*[1] for the recursive computation of the predicted state-error correlation matrix:

$$\mathbf{K}(n+1, n) = \mathbf{F}(n+1, n)\mathbf{K}(n)\mathbf{F}^H(n+1, n) + \mathbf{Q}_1(n) \tag{7.55}$$

The M-by-M matrix $\mathbf{K}(n)$ is described by the recursion:

$$\mathbf{K}(n) = \mathbf{K}(n, n-1) - \mathbf{F}(n, n+1)\mathbf{G}(n)\mathbf{C}(n)\mathbf{K}(n, n-1) \tag{7.56}$$

Here we have used the property

$$\mathbf{F}(n+1,n)\,\mathbf{F}(n, n+1) = \mathbf{I} \tag{7.57}$$

where $\mathbf{I}$ is the identity matrix. This property follows from the definition of the transition matrix. The mathematical significance of the matrix $\mathbf{K}(n)$ in Eq. (7.56) will be explained later in Section 7.5.

Figure 7.4 is a signal-flow graph representation of Eqs. (7.56) and (7.55), in that order. This diagram may be viewed as a representation of the *Riccati equation solver* in that, given $\mathbf{K}(n, n-1)$, it computes the updated value $\mathbf{K}(n+1, n)$.

Equations (7.49), (7.35), (7.31), (7.45), (7.56), and (7.55), in that order, define Kalman's one-step prediction algorithm.

Comments

The process applied to the input of the Kalman filter consists of the observed data $\mathbf{y}(1)$, $\mathbf{y}(2), \ldots, \mathbf{y}(n)$ that span the space $\mathcal{Y}_n$. The resulting filter output equals the predicted state vector $\hat{\mathbf{x}}(n+1|\mathcal{Y}_n)$. Given that the matrices $\mathbf{F}(n+1, n)$, $\mathbf{C}(n)$, $\mathbf{Q}_1(n)$, and $\mathbf{Q}_2(n)$ are all known quantities, we find from Eqs. (7.44), (7.55), and (7.56) that the predicted state-error correlation matrix $\mathbf{K}(n+1, n)$ is actually independent of the input $\mathbf{y}(n)$, which it has to be. The Kalman gain $\mathbf{G}(n)$ is also independent of the input $\mathbf{y}(n)$. Consequently, the predicted state-error correlation matrix $\mathbf{K}(n+1, n)$ and the Kalman gain $\mathbf{G}(n)$ may be computed before the Kalman filter is actually put into operation. With the correlation matrix $\mathbf{K}(n+1, n)$ providing a statistical description of the error in the predicted state vector $\hat{\mathbf{x}}(n+1|\mathcal{Y}(n))$, we may examine this matrix before actually using the Kalman filter to produce a realization of a physical system of interest; in this way, we may determine whether the solution supplied by the Kalman filter is indeed satisfactory.

As already mentioned, the Kalman filter theory assumes knowledge of the matrices $\mathbf{F}(n+1, n)$, $\mathbf{C}(n)$, $\mathbf{Q}_1(n)$ and $\mathbf{Q}_2(n)$. However, the theory may be *generalized* to include a situation where one or more of these matrices may assume values that depend on the input $\mathbf{y}(n)$. In such a situation, we find that although $\hat{\mathbf{x}}(n+1|\mathcal{Y}_n)$ and $\mathbf{K}(n+1, n)$ are still given by Eqs. (7.45) and (7.55), respectively, the Kalman gain $\mathbf{G}(n)$ and the predicted state-error correlation matrix $\mathbf{K}(n+1, n)$ are *not* precomputable (Anderson and Moore, 1979).

[1]The Riccati difference equation is named in honor of Count Jacopo Francisco Riccati. This equation has become of particular importance in control theory.

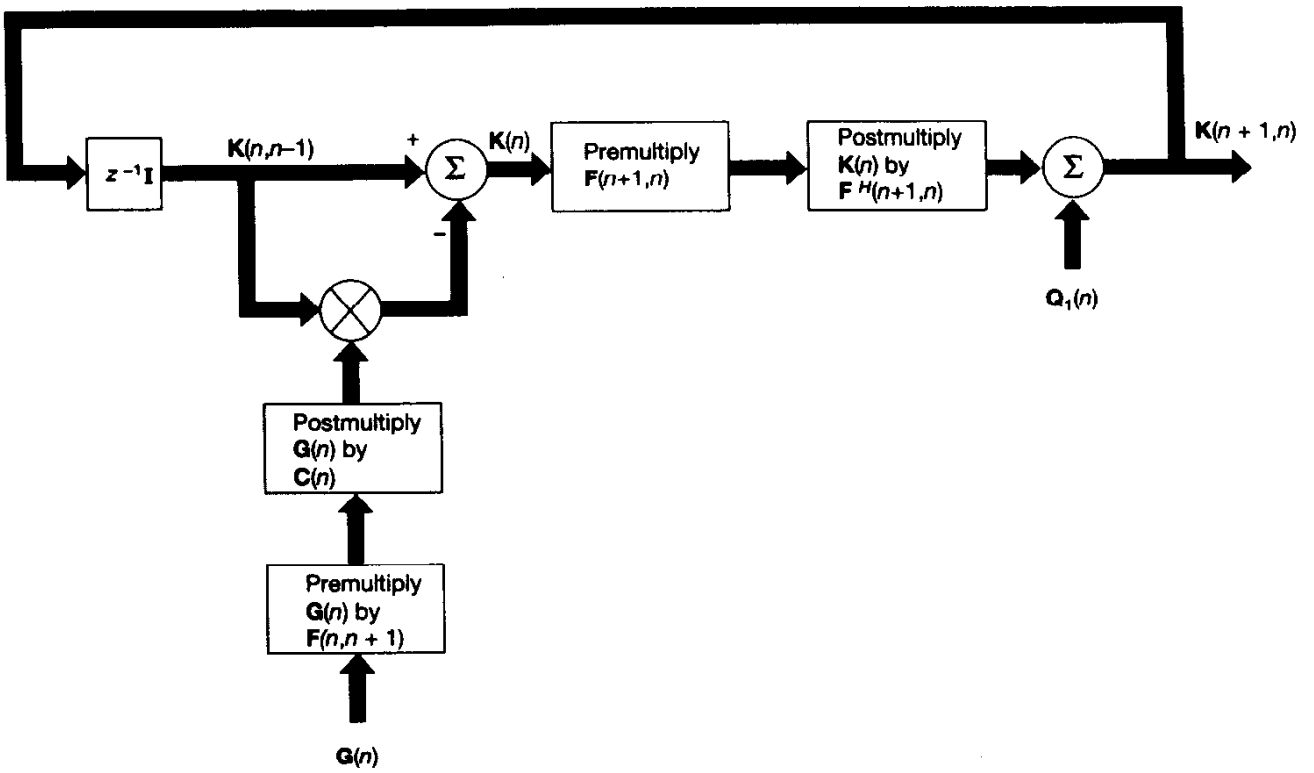

Figure 7.4 Riccati equation solver.

Rather, they both now depend on the input $\mathbf{y}(n)$. This means that $\mathbf{K}(n+1, n)$ is a *conditional* error-correlation matrix, conditional on the input $\mathbf{y}(n)$.

7.5 FILTERING

The next signal-processing operation we wish to consider is that of filtering. In particular, we wish to compute the *filtered estimate* $\hat{\mathbf{x}}(n|\mathcal{Y}_n)$ by using the one-step prediction algorithm described previously.

We first note that the state $\mathbf{x}(n)$ and the noise vector $\mathbf{v}_1(n)$ are independent of each other. Hence, from the state equation (7.17) we find that the minimum mean-square estimate of the state $\mathbf{x}(n+1)$ at time $n+1$, given the observed data up to and including time n [i.e., given $\mathbf{y}(1), \ldots, \mathbf{y}(n)$], equals

$$\hat{\mathbf{x}}(n+1|\mathcal{Y}_n) = \mathbf{F}(n+1, n)\hat{\mathbf{x}}(n|\mathcal{Y}_n) + \hat{\mathbf{v}}_1(n|\mathcal{Y}_n) \tag{7.58}$$

Since the noise vector $\mathbf{v}_1(n)$ is independent of the observed data $\mathbf{y}(1), \ldots, \mathbf{y}(n)$, it follows that the corresponding minimum mean-square estimate $\hat{\mathbf{v}}_1(n|\mathcal{Y}_n)$ is zero. Accordingly, Eq. (7.58) simplifies to

$$\hat{\mathbf{x}}(n+1|\mathcal{Y}_n) = \mathbf{F}(n+1, n)\,\hat{\mathbf{x}}(n|\mathcal{Y}_n) \tag{7.59}$$

To find the filtered estimate $\hat{\mathbf{x}}(n|\mathcal{Y}_n)$, we premultiply both sides of Eq. (7.59) by the inverse of the transition matrix $\mathbf{F}(n+1, n)$, and thus write

$$\hat{\mathbf{x}}(n|\mathcal{Y}_n) = \mathbf{F}^{-1}(n+1, n)\hat{\mathbf{x}}(n+1|\mathcal{Y}_n) \tag{7.60}$$

Using the property of the state transition matrix given in Eq. (7.57), we have

$$\mathbf{F}^{-1}(n+1, n) = \mathbf{F}(n, n+1) \tag{7.61}$$

We may therefore rewrite Eq. (7.60) in the equivalent form:

$$\hat{\mathbf{x}}(n|\mathcal{Y}_n) = \mathbf{F}(n, n+1)\,\hat{\mathbf{x}}(n+1|\mathcal{Y}_n) \tag{7.62}$$

This shows that knowing the solution to the one-step prediction problem, that is, the minimum mean-square estimate $\hat{\mathbf{x}}(n+1|\mathcal{Y}_n)$, we may determine the corresponding filtered estimate $\hat{\mathbf{x}}(n|\mathcal{Y}_n)$ simply by multiplying $\hat{\mathbf{x}}(n+1|\mathcal{Y}_n)$ by the state transition matrix $\mathbf{F}(n, n+1)$.

Filtered Estimation Error and Conversion Factor

In a filtering framework, it is natural that we define a *filtered estimation error vector* in terms of the filtered estimate of the state as follows:

$$\mathbf{e}(n) = \mathbf{y}(n) - \mathbf{C}(n)\hat{\mathbf{x}}(n|\mathcal{Y}_n) \tag{7.63}$$

This definition is similar to that of Eq. (7.31) for the innovations vector $\boldsymbol{\alpha}(n)$, except that we have substituted the filtered estimate $\hat{\mathbf{x}}(n|\mathcal{Y}_n)$ for the predicted estimate $\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})$. Using Eqs. (7.45) and (7.62) in (7.63), we get

$$\begin{aligned}\mathbf{e}(n) &= \mathbf{y}(n) - \mathbf{C}(n)\hat{\mathbf{x}}(n\,|\mathcal{Y}_{n-1}) - \mathbf{C}(n)\mathbf{F}(n, n+1)\mathbf{G}(n)\boldsymbol{\alpha}(n)\\ &= \boldsymbol{\alpha}(n) - \mathbf{C}(n)\mathbf{F}(n, n+1)\,\mathbf{G}(n)\boldsymbol{\alpha}(n)\\ &= [\mathbf{I} - \mathbf{C}(n)\mathbf{F}(n, n+1)\mathbf{G}(n)]\boldsymbol{\alpha}(n)\end{aligned} \tag{7.64}$$

The matrix-valued quantity inside the square brackets in Eq. (7.64) is called the *conversion factor*, which provides a formula for converting the innovations vector $\boldsymbol{\alpha}(n)$ into the filtered estimation error vector $\mathbf{e}(n)$. Using Eq. (7.49) to eliminate the Kalman gain $\mathbf{G}(n)$ from this definition and canceling common terms, we may rewrite Eq. (7.64) in the equivalent form:

$$\mathbf{e}(n) = \mathbf{Q}_2(n)\,\mathbf{R}^{-1}(n)\boldsymbol{\alpha}(n) \tag{7.65}$$

where $\mathbf{Q}_2(n)$ is the correlation matrix of the measurement noise process $\mathbf{v}_2(n)$, and the matrix $\mathbf{R}(n)$ is itself defined in Eq. (7.35) as the correlation matrix of the innovations process $\boldsymbol{\alpha}(n)$. Thus, except for a premultiplication by $\mathbf{Q}_2(n)$, Eq. (7.65) shows that the inverse matrix $\mathbf{R}^{-1}(n)$ plays the role of a conversion factor in the Kalman filter theory. Indeed, for the special case of $\mathbf{Q}_2(n)$ equal to the identity matrix, the inverse matrix $\mathbf{R}^{-1}$ is exactly the conversion factor defined herein.

Filtered State-Error Correlation Matrix

Earlier we introduced the M-by-M matrix $\mathbf{K}(n)$ in the formulation of the Riccati difference equation (7.55). We conclude our present discussion of the standard Kalman filter theory by showing that this matrix equals the correlation matrix of the error inherent in the filtered estimate $\hat{\mathbf{x}}(n|\mathcal{Y}_n)$.

Define the *filtered state-error vector* $\boldsymbol{\epsilon}(n)$ as the difference between the state $\mathbf{x}(n)$ and the filtered estimate $\hat{\mathbf{x}}(n|\mathcal{Y}_n)$, as shown by

$$\boldsymbol{\epsilon}(n) = \mathbf{x}(n) - \hat{\mathbf{x}}(n|\mathcal{Y}_n) \tag{7.66}$$

Substituting Eqs. (7.45) and (7.62) in (7.66), and recognizing that the product of $\mathbf{F}(n, n+1)$ and $\mathbf{F}(n+1, n)$ equals the identity matrix, we get

$$\begin{aligned}\boldsymbol{\epsilon}(n) &= \mathbf{x}(n) - \hat{\mathbf{x}}(n|\mathcal{Y}_{n-1}) - \mathbf{F}(n, n+1)\,\mathbf{G}(n)\,\boldsymbol{\alpha}(n)\\ &= \boldsymbol{\epsilon}(n, n-1) - \mathbf{F}(n, n+1)\mathbf{G}(n)\,\boldsymbol{\alpha}(n)\end{aligned} \tag{7.67}$$

where $\boldsymbol{\epsilon}(n, n, -1)$ is the predicted state-error vector at time n, using data up to time $n-1$, and $\boldsymbol{\alpha}(n)$ is the innovations process.

By definition, the correlation matrix of the filtered state-error vector $\boldsymbol{\epsilon}(n)$ equals the expectation $E[\boldsymbol{\epsilon}(n)\boldsymbol{\epsilon}^H(n)]$. Hence, using Eq. (7.67), we may express this expectation as follows:

$$E[\boldsymbol{\epsilon}(n)\boldsymbol{\epsilon}^H(n)] = E[\boldsymbol{\epsilon}(n, n-1)\,\boldsymbol{\epsilon}^H(n, n-1)] + \mathbf{F}(n, n+1)\,\mathbf{G}(n)\,E[\boldsymbol{\alpha}(n)\boldsymbol{\alpha}^H(n)]\,\mathbf{G}^H(n)\,\mathbf{F}^H(n, n+1) - 2E[\boldsymbol{\epsilon}(n, n-1)\boldsymbol{\alpha}^H(n)]\,\mathbf{G}^H(n)\,\mathbf{F}^H(n, n+1) \tag{7.68}$$

Examining the right-hand side of Eq. (7.68), we find that the three expectations contained in it may be interpreted individually as follows:

1. The first expectation equals the predicted state-error correlation matrix:

$$\mathbf{K}(n, n-1) = E[\boldsymbol{\epsilon}(n, n-1)\,\boldsymbol{\epsilon}^H(n, n-1)]$$

2. The expectation in the second term equals the correlation matrix of the innovations process $\boldsymbol{\alpha}(n)$:

$$\mathbf{R}(n) = E[\boldsymbol{\alpha}(n)\boldsymbol{\alpha}^H(n)]$$

3. The expectation in the third term may be expressed as follows:

$$E[\boldsymbol{\epsilon}(n, n-1)\,\boldsymbol{\alpha}^H(n)] = E[(\mathbf{x}(n) - \hat{\mathbf{x}}(n|\mathcal{Y}_{n-1}))\boldsymbol{\alpha}^H(n)] = E[\mathbf{x}(n)\,\boldsymbol{\alpha}^H(n)]$$

where, in the last line, we have used the fact that the estimate $\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})$ is orthogonal to the innovations process $\boldsymbol{\alpha}(n)$ acting as input. Next, from Eq. (7.42) we see, by putting $k = n$ and then premultiplying both sides by the inverse matrix $\mathbf{F}^{-1}(n+1, n) = \mathbf{F}(n, n+1)$, that

$$E[\mathbf{x}(n)\,\boldsymbol{\alpha}^H(n)] = \mathbf{F}(n, n+1)\,E[\mathbf{x}(n+1)\boldsymbol{\alpha}^H(n)] = \mathbf{F}(n, n+1)\mathbf{G}(n)\,\mathbf{R}(n)$$

where, in the last line, we have made use of Eq. (7.44). Hence,

$$E[\boldsymbol{\epsilon}(n, n-1)\,\boldsymbol{\alpha}^H(n)] = \mathbf{F}(n, n+1)\,\mathbf{G}(n)\,\mathbf{R}(n)$$

We may now use these results in Eq. (7.68), and so obtain

$$E[\boldsymbol{\epsilon}(n)\boldsymbol{\epsilon}^H(n)] = \mathbf{K}(n, n-1) - \mathbf{F}(n, n+1)\,\mathbf{G}(n)\,\mathbf{R}(n)\mathbf{G}^H(n)\,\mathbf{F}^H(n, n+1) \tag{7.69}$$

We may further simplify this result by noting that [see Eq. (7.49)]

$$\mathbf{G}(n)\,\mathbf{R}(n) = \mathbf{F}(n+1, n)\,\mathbf{K}(n, n-1)\,\mathbf{C}^H(n) \tag{7.70}$$

Accordingly, using Eqs. (7.69) and (7.70), and recognizing that the product of $\mathbf{F}(n, n+1)$ and $\mathbf{F}(n+1, n)$ equals the identity matrix, we get the desired result for the filtered state-error correlation matrix:

$$E[\boldsymbol{\epsilon}(n)\boldsymbol{\epsilon}^H(n)] = \mathbf{K}(n, n-1) - \mathbf{K}(n, n-1)\mathbf{C}^H(n)\,\mathbf{G}^H(n)\mathbf{F}^H(n, n+1)$$

Equivalently, using the Hermitian property of $E[\boldsymbol{\epsilon}(n)\boldsymbol{\epsilon}^H(n)]$ and that of $\mathbf{K}(n, n-1)$, we may write

$$E[\boldsymbol{\epsilon}(n)\boldsymbol{\epsilon}^H(n)] = \mathbf{K}(n, n-1) - \mathbf{F}(n, n+1)\mathbf{G}(n)\,\mathbf{C}(n)\,\mathbf{K}(n, n-1) \tag{7.71}$$

Comparing Eq. (7.71) with (7.56), we readily see that

$$E[\boldsymbol{\epsilon}(n)\,\boldsymbol{\epsilon}^H(n)] = \mathbf{K}(n)$$

This shows that the matrix $\mathbf{K}(n)$ used in the Riccati difference equation (7.55) is in fact the *filtered state-error correlation matrix*. The matrix $\mathbf{K}(n)$ is used as the statistical description of the error in the filtered estimate $\hat{\mathbf{x}}(n|\mathcal{Y}_n)$.

7.6 INITIAL CONDITIONS

To operate the one-step prediction and filtering algorithms described in Sections 7.4 and 7.5, we obviously need to specify the *initial conditions*. We now address this issue.

The initial state of the process equation (7.17) is not known precisely. Rather, it is usually described by its mean and correlation matrix. In the absence of any observed data at time $n = 0$, we may choose the *initial predicted estimate* as

$$\hat{\mathbf{x}}(1|\mathcal{Y}_0) = E[\mathbf{x}(1)] \tag{7.72}$$

and its *correlation matrix*

$$\begin{aligned}\mathbf{K}(1,0) &= E[(\mathbf{x}(1) - E[\mathbf{x}(1)])(\mathbf{x}(1) - E[\mathbf{x}(1)])^H] \\ &= \boldsymbol{\Pi}_0\end{aligned} \tag{7.73}$$

This choice for the initial conditions is not only intuitively satisfying but also has the advantage of yielding a filtered estimate of the state $\hat{\mathbf{x}}(n|\mathcal{Y}_n)$ that is *unbiased* (see Problem 10). Assuming that the state vector $\mathbf{x}(n)$ has *zero mean*, we may simplify Eqs. (7.72) and (7.73) by setting

$$\hat{\mathbf{x}}(1|\mathcal{Y}_0) = \mathbf{0}$$

and

$$\mathbf{K}(1,0) = E[\mathbf{x}(1)\,\mathbf{x}^H(1)] = \boldsymbol{\Pi}_0$$

7.7 SUMMARY OF THE KALMAN FILTER

Table 7.1 presents a summary of the variables used to formulate the solution to the Kalman filtering problem. The input of the filter is the vector process $\mathbf{y}(n)$, represented by the vector space $\mathcal{Y}_n$, and the output is the filtered estimate $\hat{\mathbf{x}}(n|\mathcal{Y}_n)$ of the state vector. In Table 7.2, we present a summary of the Kalman filter (including initial conditions) based on the one-step prediction algorithm.

TABLE 7.1 SUMMARY OF THE KALMAN VARIABLES

Variable	Definition	Dimension
$\mathbf{x}(n)$	State vector at time n	M-by-1
$\mathbf{y}(n)$	Observation vector at time n	N-by-1
$\mathbf{F}(n+1, n)$	State transition matrix from time n to $n+1$	M-by-M
$\mathbf{C}(n)$	Measurement matrix at time n	N-by-M
$\mathbf{Q}_1(n)$	Correlation matrix of process noise vector $\mathbf{v}_1(n)$	M-by-M
$\mathbf{Q}_2(n)$	Correlation matrix of measurement noise vector $\mathbf{v}_2(n)$	N-by-N
$\hat{\mathbf{x}}(n+1\|\mathcal{Y}_n)$	Predicted estimate of the state vector at time $n+1$, given the observation vectors $\mathbf{y}(1), \mathbf{y}(2), \ldots, \mathbf{y}(n)$	M-by-1
$\hat{\mathbf{x}}(n\|\mathcal{Y}_n)$	Filtered estimate of the state vector at time n, given the observation vectors $\mathbf{y}(1), \mathbf{y}(2), \ldots, \mathbf{y}(n)$	M-by-1
$\mathbf{G}(n)$	Kalman gain at time n	M-by-N
$\boldsymbol{\alpha}(n)$	Innovations vector at time n	N-by-1
$\mathbf{R}(n)$	Correlation matrix of the innovations vector $\boldsymbol{\alpha}(n)$	N-by-N
$\mathbf{K}(n+1, n)$	Correlation matrix of the error in $\hat{\mathbf{x}}(n+1\|\mathcal{Y}_n)$	M-by-M
$\mathbf{K}(n)$	Correlation matrix of the error in $\hat{\mathbf{x}}(n\|\mathcal{Y}_n)$	M-by-M

TABLE 7.2 SUMMARY OF THE KALMAN FILTER BASED ON ONE-STEP PREDICTION

Input vector process

Observations = $\{\mathbf{y}(1), \mathbf{y}(2), \ldots, \mathbf{y}(n)\}$

Known parameters

State transition matrix = $\mathbf{F}(n+1, n)$

Measurement matrix = $\mathbf{C}(n)$

Correlation matrix of process noise vector = $\mathbf{Q}_1(n)$

Correlation matrix of measurement noise vector = $\mathbf{Q}_2(n)$

Computation: $n = 1, 2, 3, \ldots$

$$\mathbf{G}(n) = \mathbf{F}(n+1, n)\mathbf{K}(n, n-1)\mathbf{C}^H(n)[\mathbf{C}(n)\mathbf{K}(n, n-1)\mathbf{C}^H(n) + \mathbf{Q}_2(n)]^{-1}$$

$$\boldsymbol{\alpha}(n) = \mathbf{y}(n) - \mathbf{C}(n)\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})$$

$$\hat{\mathbf{x}}(n+1|\mathcal{Y}_n) = \mathbf{F}(n+1, n)\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1}) + \mathbf{G}(n)\boldsymbol{\alpha}(n)$$

$$\mathbf{K}(n) = \mathbf{K}(n, n-1) - \mathbf{F}(n, n+1)\mathbf{G}(n)\mathbf{C}(n)\mathbf{K}(n, n-1)$$

$$\mathbf{K}(n+1, n) = \mathbf{F}(n+1, n)\mathbf{K}(n)\mathbf{F}^H(n+1, n) + \mathbf{Q}_1(n)$$

Initial conditions:

$$\hat{\mathbf{x}}(1|\mathcal{Y}_0) = E[\mathbf{x}(1)]$$

$$\mathbf{K}(1, 0) = E[(\mathbf{x}(1) - E[\mathbf{x}(1)])(\mathbf{x}(1) - E[\mathbf{x}(1)])^H] = \Pi_0$$

A block diagram representation of the Kalman filter is given in Fig. 7.5, which is based on three functional blocks:

- Kalman gain computer, described in Fig. 7.2
- One-step predictor, described in Fig. 7.3.
- Riccati equation solver, described in Fig. 7.4

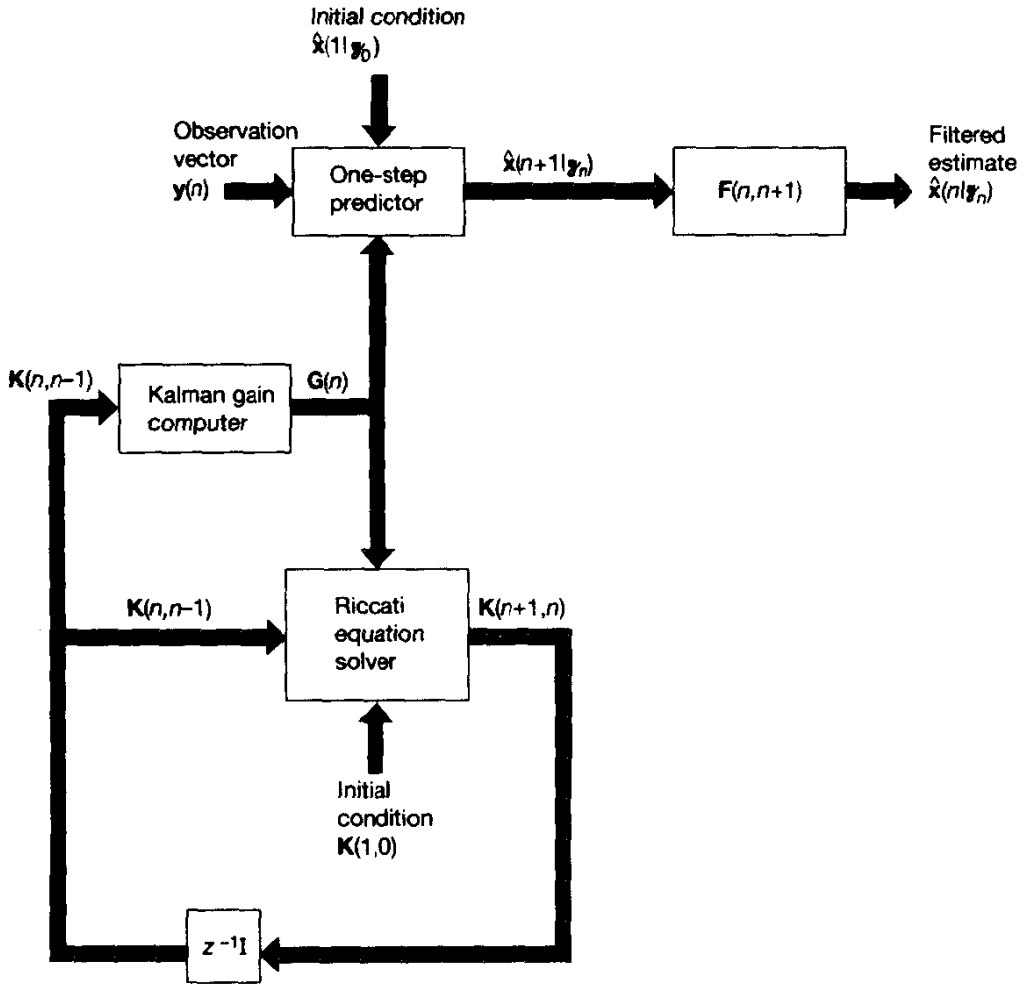

Figure 7.5 Block diagram of the Kalman filter based on one-step prediction.

7.8 VARIANTS OF THE KALMAN FILTER

As mentioned in the introductory remarks to this chapter, the main reason for our interest in Kalman filter theory in this book is that it provides a general framework for the derivation of certain adaptive filtering algorithms known collectively as the family of recursive least-squares (RLS) algorithms.

The application of Kalman filter theory to adaptive filtering was apparently first reported in the literature by Lawrence and Kaufman (1971); see Problem 8. This was followed by Godard (1974), who used an approach different from that of Lawrence and Kaufman. In particular, Godard formulated the adaptive filtering problem (using a tap-delay-line structure) as the estimation of a state vector in Gaussian noise, which represents

a classical Kalman filtering problem. Godard's paper prompted many other investigators to explore the application of Kalman filter theory to adaptive filtering problems.

However, we had to await the paper by Sayed and Kailath (1994) to discover how indeed the Riccati-based Kalman filtering algorithm and its variants can be correctly framed into one-to-one correspondences with all the known algorithms in the RLS family. We will take up the details of this unifying framework later in the book. For now, we will focus our attention on a special dynamical model that befits our future needs.

Special Case: Unforced Dynamics

Consider a linear dynamical system whose state-space model is described by the following pair of state equations (Sayed and Kailath, 1994):

$$\mathbf{x}(n+1) = \lambda^{-1/2}\mathbf{x}(n) \tag{7.74}$$

$$y(n) = \mathbf{u}^H(n)\mathbf{x}(n) + v(n) \tag{7.75}$$

where λ is a positive real scalar. According to this model, the process noise is zero, and the measurement noise, denoted by the scalar $v(n)$, is a zero-mean white noise process with unit variance, as shown by

$$E[v(n)v^*(k)] = \begin{cases} 1, & n = k \\ 0, & n \neq k \end{cases} \tag{7.76}$$

Thus, comparing this model with the general model described by Eqs. (7.17) to (7.21), we note the following:

$$\mathbf{F}(n+1, n) = \lambda^{-1/2}\mathbf{I} \tag{7.77}$$

$$\mathbf{Q}_1(n) = \mathbf{O} \tag{7.78}$$

$$\mathbf{C}(n) = \mathbf{u}^H(n) \tag{7.79}$$

$$\mathbf{Q}_2(n) = 1 \tag{7.80}$$

The state-space model described by Eqs. (7.74) to (7.76) is referred to as an *unforced dynamical model* by virtue of the fact that the process equation (7.74) is free of an external force. Most importantly, the state transition matrix of the model is equal to the identity matrix $\mathbf{I}$ scaled by the constant $\lambda^{-1/2}$. Consequently, the predicted state-error correlation matrix $\mathbf{K}(n+1, n)$ and the filtered state-error correlation matrix $\mathbf{K}(n)$ assume a common value; see Problem 9.

This special unforced dynamical model holds the key to the formulation of a general framework for deriving the RLS family of adaptive filtering algorithms. As we shall see later in the book, the constant λ has a significant role in the operation of these algorithms. For now we content ourselves by considering variants of the Kalman filtering algorithm based on this model.

TABLE 7.3 SUMMARY OF THE COVARIANCE (KALMAN) FILTERING ALGORITHM FOR THE SPECIAL UNFORCED DYNAMICAL MODEL

Input scalar process:
Observations: $y(1), y(2), \ldots y(n)$
Known parameters:
state transition matrix $= \lambda^{-1/2}\mathbf{I}$, $\mathbf{I}$ = identity matrix
measurement matrix $= \mathbf{u}^H(n)$
variance of measurement noise $v(n) = 1$
Initial conditions:

$$\hat{\mathbf{x}}(1\,|\mathcal{Y}_0) = E[\mathbf{x}(1)]$$

$$\mathbf{K}(1,0) = E[(\mathbf{x}(1) - E[\mathbf{x}(1)])(\mathbf{x}(1) - E[\mathbf{x}(1)])^H] = \Pi_0$$

Computation: $n = 1, 2, 3, \ldots$

$$\mathbf{g}(n) = \frac{\lambda^{-1/2}\mathbf{K}(n-1)\mathbf{u}(n)}{\mathbf{u}^H(n)\mathbf{K}(n-1)\mathbf{u}(n) + 1}$$

$$\alpha(n) = y(n) - \mathbf{u}^H(n)\hat{\mathbf{x}}(n\,|\mathcal{Y}_{n-1})$$

$$\hat{\mathbf{x}}(n+1\,|\mathcal{Y}_n) = \lambda^{-1/2}\hat{\mathbf{x}}(n\,|\mathcal{Y}_{n-1}) + \mathbf{g}(n)\alpha(n)$$

$$\mathbf{K}(n) = \lambda^{-1}\mathbf{K}(n-1) - \lambda^{-1/2}\mathbf{g}(n)\mathbf{u}^H(n)\mathbf{K}(n-1)$$

Covariance (Kalman) Filtering Algorithm

The Kalman filtering algorithm summarized in Table 7.2 is designed to propagate the correlation (covariance) matrix $\mathbf{K}(n+1, n)$ that refers to the error in the state's estimate $\hat{\mathbf{x}}(n+1|\mathcal{Y}_n)$. This algorithm is therefore commonly referred to as the *covariance (Kalman) filtering algorithm*. For the unforced dynamical model at hand, we find that substituting Eqs. (7.77) to (7.80) in Table 7.2 yields the simplified covariance filtering algorithm summarized in Table 7.3. In this table we have used $\mathbf{g}(n)$ to denote the Kalman gain, as it takes the form of a vector here.

Information Filtering Algorithm

The Kalman filter may also be implemented by propagating the inverse matrix $\mathbf{K}^{-1}(n)$ which accentuates the recursive least-squares nature of the filtering process. The *inverse state-error correlation matrix*, $\mathbf{K}^{-1}(n)$, is related to *Fisher's information matrix*[2], which permits an interpretation of filter performance in information-theoretic terms. For this reason, an implementation of the Kalman filtering algorithm based on $\mathbf{K}^{-1}(n)$ is termed the *information filtering algorithm* (Fraser, 1967).

For the derivation of the information filtering algorithm, we may proceed in the manner described next.

Step 1. We start with the Riccati difference equation which, for the special unforced dynamical model, has the form (see the last line of the algorithm in Table 7.3):

[2]Fisher's information matrix is discussed in Appendix D.

$$\mathbf{K}(n) = \lambda^{-1}\mathbf{K}(n-1) - \lambda^{-1/2}\mathbf{g}(n)\mathbf{u}^H(n)\mathbf{K}(n-1) \tag{7.81}$$

Solving this equation for the matrix product $\mathbf{g}(n)\mathbf{u}^H(n)\mathbf{K}(n-1)$, we get

$$\mathbf{g}(n)\mathbf{u}^H(n)\mathbf{K}(n-1) = \lambda^{-1/2}\mathbf{K}(n-1) - \lambda^{1/2}\mathbf{K}(n) \tag{7.82}$$

Next, from the first line of the algorithm in Table 7.3, the Kalman gain for the unforced dynamical model of interest is defined by

$$\mathbf{g}(n) = \frac{\lambda^{-1/2}\mathbf{K}(n-1)\mathbf{u}(n)}{\mathbf{u}^H(n)\mathbf{K}(n-1)\mathbf{u}(n) + 1} \tag{7.83}$$

Cross-multiplying and rearranging terms, we may rewrite Eq. (7.83) as

$$\mathbf{g}(n) = \lambda^{-1/2}\mathbf{K}(n-1)\mathbf{u}(n) - (\mathbf{g}(n)\mathbf{u}^H(n)\mathbf{K}(n-1))\mathbf{u}(n) \tag{7.84}$$

Substituting Eq. (7.82) in (7.84), and then canceling common terms, we get a new definition for the Kalman gain:

$$\mathbf{g}(n) = \lambda^{1/2}\mathbf{K}(n)\mathbf{u}(n) \tag{7.85}$$

Next, eliminating $\mathbf{g}(n)$ between Eqs. (7.82) and (7.85), and multiplying the result by $\lambda^{1/2}$, we get

$$\mathbf{K}(n-1) = \lambda\mathbf{K}(n)\mathbf{u}(n)\mathbf{u}^H(n)\mathbf{K}(n-1) + \lambda\mathbf{K}(n) \tag{7.86}$$

Premultiplying Eq. (7.86) by the inverse matrix $\mathbf{K}^{-1}(n)$ and postmultiplying it by $\mathbf{K}^{-1}(n-1)$, we get the first recursion of the information filtering algorithm:

$$\mathbf{K}^{-1}(n) = \lambda\mathbf{K}^{-1}(n-1) + \lambda\mathbf{u}(n)\mathbf{u}^H(n) \tag{7.87}$$

Step 2. From the second and third lines of the algorithm summarized in Table 7.3, we have, respectively,

$$\alpha(n) = y(n) - \mathbf{u}^H(n)\hat{\mathbf{x}}(n \mid \mathcal{Y}_{n-1}) \tag{7.88}$$

and

$$\hat{\mathbf{x}}(n+1 \mid \mathcal{Y}_n) = \lambda^{-1/2}\hat{\mathbf{x}}(n \mid \mathcal{Y}_{n-1}) + \mathbf{g}(n)\alpha(n) \tag{7.89}$$

Therefore, substituting Eq. (7.85) in (7.89), we get

$$\hat{\mathbf{x}}(n+1 \mid \mathcal{Y}_n) = \lambda^{-1/2}\hat{\mathbf{x}}(n \mid \mathcal{Y}_{n-1}) + \lambda^{1/2}\mathbf{K}(n)\mathbf{u}(n)\alpha(n) \tag{7.90}$$

Next, eliminating $\alpha(n)$ between Eqs. (7.88) and (7.90) yields

$$\hat{\mathbf{x}}(n+1 \mid \mathcal{Y}_n) = [\lambda^{-1/2}\mathbf{I} - \lambda^{1/2}\mathbf{K}(n)\mathbf{u}(n)\mathbf{u}^H(n)]\hat{\mathbf{x}}(n \mid \mathcal{Y}_{n-1}) + \lambda^{1/2}\mathbf{K}(n)\mathbf{u}(n)y(n) \tag{7.91}$$

But, from Eq. (7.86), we readily deduce the following relation:

$$\lambda^{-1/2}\mathbf{I} - \lambda^{1/2}\mathbf{K}(n)\mathbf{u}(n)\mathbf{u}^H(n) = \lambda^{1/2}\mathbf{K}(n)\mathbf{K}^{-1}(n-1) \tag{7.92}$$

Accordingly, we may simplify Eq. (7.91) as follows:

$$\hat{\mathbf{x}}(n+1 \mid \mathcal{Y}_n) = \lambda^{1/2}\mathbf{K}(n)\mathbf{K}^{-1}(n-1)\hat{\mathbf{x}}(n \mid \mathcal{Y}_{n-1}) + \lambda^{1/2}\mathbf{K}(n)\mathbf{u}(n)y(n)$$

Premultiplying this equation by the inverse matrix $\mathbf{K}^{-1}(n)$, we get the second recursion of the information filtering algorithm:

$$\mathbf{K}^{-1}(n)\hat{\mathbf{x}}(n+1 \mid \mathcal{Y}_n) = \lambda^{1/2}[\mathbf{K}^{-1}(n-1)\hat{\mathbf{x}}(n \mid \mathcal{Y}_{n-1}) + \mathbf{u}(n)y(n)] \tag{7.93}$$

Note that in Eq. (7.93) the algorithm propagates the product $\mathbf{K}^{-1}(n-1)\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})$ rather than the estimate $\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})$ by itself.

Step 3. Finally, the updated value of the state's estimate is computed by combining the results of steps 1 and 2 as follows:

$$\begin{aligned}\hat{\mathbf{x}}(n+1|\mathcal{Y}_n) &= \mathbf{K}(n)\,(\mathbf{K}^{-1}(n)\hat{\mathbf{x}}(n+1|\mathcal{Y}_n)) \\ &= [\mathbf{K}^{-1}(n)]^{-1}(\mathbf{K}^{-1}(n)\hat{\mathbf{x}}(n+1|\mathcal{Y}_n))\end{aligned} \tag{7.94}$$

Equations (7.87), (7.93), and (7.94), in that order, constitute the information-filtering algorithm for the unforced dynamical model of Eqs. (7.74) to (7.76). A summary of the algorithm is presented in Table 7.4.

Although the covariance and information implementations of the Kalman filter, as described herein, are algebraically equivalent, the numerical properties of these two algorithms may differ substantially from each other (Kaminski et al., 1971). However, both algorithms require the same number of algebraic operations (i.e., multiplications and additions), which, for the special model at hand, is $O(M^2)$, where M is the state dimension.

Square-root Filtering

The covariance implementation of the Kalman filter, summarized in Table 7.2, is the optimal solution to the linear filtering problem posed in Section 7.2. However, this algorithm is prone to serious numerical difficulties that are well documented in the literature (Kamin-

TABLE 7.4 SUMMARY OF THE INFORMATION-FILTERING ALGORITHM FOR THE SPECIAL UNFORCED DYNAMICAL MODEL

Input scalar process:
observations = $y(1), y(2), \ldots, y(n)$
Known parameters:
state transition matrix $= \lambda^{-1/2}\mathbf{I}$, $\mathbf{I}$ = identity matrix
measurement matrix $= \mathbf{u}^H(n)$
variance of measurement noise $v(n) = 1$
Initial conditions:

$$\hat{\mathbf{x}}(1 \mid \mathcal{Y}_0) = E[\mathbf{x}(1)]$$

$$\mathbf{K}(1,0) = E[(\mathbf{x}(1) - E[\mathbf{x}(1)])(\mathbf{x}(1) - E[\mathbf{x}(1)])^H] = \Pi_0$$

Computation: $n = 1, 2, 3 \ldots$

$$\mathbf{K}^{-1}(n) = \lambda[\mathbf{K}^{-1}(n-1) + \mathbf{u}(n)\mathbf{u}^H(n)]$$

$$\mathbf{K}^{-1}(n)\hat{\mathbf{x}}(n+1 \mid \mathcal{Y}_n) = \lambda^{1/2}[\mathbf{K}^{-1}(n-1)\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1}) + \mathbf{u}(n)y(n)]$$

$$\hat{\mathbf{x}}(n+1 \mid \mathcal{Y}_n) = [\mathbf{K}^{-1}(n)]^{-1}\mathbf{K}^{-1}(n)\hat{\mathbf{x}}(n+1|\mathcal{Y}_n)$$

ski et al., 1971; Bierman and Thornton, 1977). For example, according to Eq. (7.56) the matrix $\mathbf{K}(n)$ is defined as the difference between two nonnegative definite matrices; hence, unless the numerical accuracy employed at every iteration of the algorithm is high enough, the matrix $\mathbf{K}(n)$ resulting from this computation may *not* be nonnegative definite. Such a situation is clearly unacceptable, because $\mathbf{K}(n)$ represents a correlation matrix. The unstable behavior of the Kalman filter, which results from numerical inaccuracies due to the use of finite wordlength arithmetic, is called the *divergence phenomenon.*

This problem may be overcome by using numerically stable unitary transformations at every iteration of the Kalman filtering algorithm (Potter, 1963; Kaminski et al., 1971; Morf and Kailath, 1975). In particular, the matrix $\mathbf{K}(n)$ is propagated in a square-root form by using the *Cholesky factorization*[3]:

$$\mathbf{K}(n) = \mathbf{K}^{1/2}(n)\mathbf{K}^{H/2}(n) \tag{7.95}$$

where $\mathbf{K}^{1/2}(n)$ is reserved for a lower triangular matrix, and $\mathbf{K}^{H/2}$ is its Hermitian transpose. In linear algebra, the Cholesky factor $\mathbf{K}^{1/2}(n)$ is commonly referred to as the *square root* of the matrix $\mathbf{K}(n)$. Accordingly, any variant of the Kalman filtering algorithm based on the Cholesky factorization is referred to as *square-root filtering*. The important point to note here is that the matrix product $\mathbf{K}^{1/2}(n)\mathbf{K}^{H/2}(n)$ is much less likely to become indefinite, because the product of any square matrix and its Hermitian transpose is always positive definite. Indeed, even in the presence of roundoff errors, the numerical conditioning of the Cholesky factor $\mathbf{K}^{1/2}(n)$ is generally much better than that of $\mathbf{K}(n)$ itself; see Problem 12.

The information filtering algorithm may also be implemented in a square-root form of its own by propagating the square root $\mathbf{K}^{-1/2}(n)$ rather than the inverse matrix $\mathbf{K}^{-1}(n)$ itself (Kaminski et al., 1971; Bierman, 1977). In this variant of the Kalman filter, the Cholesky factorization is used to express the inverse matrix $\mathbf{K}^{-1}(n)$ as follows:

$$\mathbf{K}^{-1}(n) = \mathbf{K}^{-H/2}(n)\mathbf{K}^{-1/2}(n) \tag{7.96}$$

where $\mathbf{K}^{-1/2}(n)$ is a lower triangular matrix, and $\mathbf{K}^{-H/2}$ is its Hermitian transpose.

UD-factorization

The square-root implementation of a Kalman filter requires more computation than the conventional Kalman filter. This problem of computational efficiency led to the development of a modified version of the square-root filtering algorithm known as the *UD-factorization algorithm* (Bierman, 1977). In this second approach, the filtered state-error correlation matrix $\mathbf{K}(n)$ is factored into an upper triangular matrix $\mathbf{U}(n)$ with 1's along its main diagonal and a real diagonal matrix $\mathbf{D}(n)$, as shown by

$$\mathbf{K}(n) = \mathbf{U}(n)\mathbf{D}(n)\mathbf{U}^{H}(n) \tag{7.97}$$

[3]The Cholesky factorization was also discussed in Section 6.7 in the context of linear prediction.

Equivalently, the factorization may be written as

$$\mathbf{K}(n) = (\mathbf{U}(n)\mathbf{D}^{1/2}(n))\,(\mathbf{U}(n)\mathbf{D}^{1/2}(n))^H \tag{7.98}$$

where $\mathbf{D}^{1/2}(n)$ is the *square-root* of $\mathbf{D}(n)$. The nonnegative definiteness of the computed matrix $\mathbf{K}(n)$ is guaranteed by updating the factors $\mathbf{U}(n)$ and $\mathbf{D}(n)$ instead of $\mathbf{K}(n)$ itself. However, a Kalman filter based on the UD-factorization does *not* possess the numerical advantage of a standard square-root Kalman filter. Moreover, a Kalman filter using UD-factorization may suffer from serious overflow/underflow problems (Stewart and Chapman, 1990). When an arithmetic operation produces a resultant number with too large or too small a characteristic, it is said to suffer from *overflow* or *underflow*, respectively.

One final comment is in order. With the ever-increasing improvements in digital technology, the old argument that square roots are expensive and awkward to calculate is no longer as compelling as it used to be. Accordingly, to avoid the divergence of a Kalman filter, we will only pursue a detailed discussion of square-root filtering in this book. This we do in Chapter 14, after equipping ourselves with certain unitary transformations in Chapter 12.

7.9 THE EXTENDED KALMAN FILTER

The Kalman filtering problem considered up to this point in the discussion has addressed the estimation of a state vector in a linear model of a dynamical system. If, however, the model is *nonlinear*, we may extend the use of Kalman filtering through a linearization procedure. The resulting filter is naturally referred to as the *extended Kalman filter* (EKF). Such an extension is feasible by virtue of the fact that the Kalman filter is described in terms of differential equations (in the case of continuous-time systems) or difference equations (in the case of discrete-time systems). This is in contrast to the Wiener filter that is limited to linear systems, since the notion of an impulse response (on which the Wiener filter is based) is meaningful only in the context of linear systems. Here is another important advantage of the Kalman filter over the Wiener filter.

To set the stage for a development of the extended Kalman filter in the discrete-time domain, consider first the standard linear state-space model that we studied in the earlier part of this chapter [Eqs. (7.17) and (7.19)], reproduced here for convenience of presentation:

$$\mathbf{x}(n+1) = \mathbf{F}(n+1, n)\mathbf{x}(n) + \mathbf{v}_1(n) \tag{7.99}$$

$$\mathbf{y}(n) = \mathbf{C}(n)\mathbf{x}(n) + \mathbf{v}_2(n) \tag{7.100}$$

where $\mathbf{v}_1(n)$ and $\mathbf{v}_2(n)$ are uncorrelated zero-mean white-noise processes with correlation matrices $\mathbf{Q}_1(n)$ and $\mathbf{Q}_2(n)$, respectively, as defined in Eqs. (7.18), (7.20), and (7.21). The corresponding Kalman filter equations are summarized in Table 7.2. In this section, how-

ever, we will rewrite these equations in a slightly modified form that is more convenient for our present discussion. Specifically, the update of the state estimate is performed in two steps. The first step updates $\hat{\mathbf{x}}(n|\mathcal{Y}_n)$ to $\hat{\mathbf{x}}(n+1|\mathcal{Y}_n)$; this update equation is simply (7.59). The second step updates $\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})$ to $\hat{\mathbf{x}}(n|\mathcal{Y}_n)$ and is obtained by substituting Eq. (7.45) into Eq. (7.60), and by defining a new gain matrix:

$$\mathbf{G}_f(n) = \mathbf{F}^{-1}(n+1, n)\mathbf{G}(n): \tag{7.101}$$

We may thus write

$$\hat{\mathbf{x}}(n+1|\mathcal{Y}_n) = \mathbf{F}(n+1, n)\hat{\mathbf{x}}(n|\mathcal{Y}_n) \tag{7.102}$$

$$\hat{\mathbf{x}}(n|\mathcal{Y}_n) = \hat{\mathbf{x}}(n|\mathcal{Y}_{n-1}) + \mathbf{G}_f(n)\boldsymbol{\alpha}(n) \tag{7.103}$$

$$\boldsymbol{\alpha}(n) = \mathbf{y}(n) - \mathbf{C}(n)\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1}) \tag{7.104}$$

$$\mathbf{G}_f(n) = \mathbf{K}(n, n-1)\mathbf{C}^H(n)[\mathbf{C}(n)\mathbf{K}(n, n-1)\mathbf{C}^H(n) + \mathbf{Q}_2(n)]^{-1} \tag{7.105}$$

$$\mathbf{K}(n+1, n) = \mathbf{F}(n+1, n)\mathbf{K}(n)\mathbf{F}^H(n+1, n) + \mathbf{Q}_1(n) \tag{7.106}$$

$$\mathbf{K}(n) = [\mathbf{I} - \mathbf{G}_f(n)\mathbf{C}(n)]\mathbf{K}(n, n-1) \tag{7.107}$$

We next make the following observation. Suppose that instead of the state equations (7.99) and (7.100), we are given the alternative state-space model

$$\mathbf{x}(n+1) = \mathbf{F}(n+1, n)\mathbf{x}(n) + \mathbf{v}_1(n) + \mathbf{d}(n) \tag{7.108}$$

$$\mathbf{y}(n) = \mathbf{C}(n)\mathbf{x}(n) + \mathbf{v}_2(n) \tag{7.109}$$

where $\mathbf{d}(n)$ is a known (i.e., nonrandom) vector. In this case, it is easily verified that the same Kalman equations (7.103) through (7.107) apply except for a modification in the first equation (7.102), which now reads as follows:

$$\hat{\mathbf{x}}(n+1|\mathcal{Y}_n) = \mathbf{F}(n+1, n)\hat{\mathbf{x}}(n|\mathcal{Y}_n) + \mathbf{d}(n) \tag{7.110}$$

This modification arises in the derivation of the extended Kalman filter, as discussed in the sequel.

As mentioned previously, the extended Kalman filter (EKF) is an *approximate* solution that allows us to extend the Kalman filtering idea to *nonlinear* state-space models (Jazwinski, 1970; Maybeck, 1982; Ljung and Söderstrom, 1983). In particular, the nonlinear model considered here has the following form:

$$\mathbf{x}(n+1) = \mathbf{F}(n,\mathbf{x}(n)) + \mathbf{v}_1(n) \tag{7.111}$$

$$\mathbf{y}(n) = \mathbf{C}(n,\mathbf{x}(n)) + \mathbf{v}_2(n) \tag{7.112}$$

where, as before, $\mathbf{v}_1(n)$ and $\mathbf{v}_2(n)$ are uncorrelated zero-mean white-noise processes with correlation matrices $\mathbf{Q}_1(n)$ and $\mathbf{Q}_2(n)$, respectively. Here, however, the functional

$\mathbf{F}(n,\mathbf{x}(n))$ denotes a nonlinear transition matrix function that is possibly time-variant. In the linear case,we simply have

$$\mathbf{F}(n,\mathbf{x}(n)) = \mathbf{F}(n+1,n)\mathbf{x}(n)$$

But in a general nonlinear setting, the entries of the state vector $\mathbf{x}(n)$ may be combined nonlinearly by the action of the functional $\mathbf{F}(n,\mathbf{x}(n))$. Moreover, this nonlinear operation may vary with time. Likewise, the functional $\mathbf{C}(n,\mathbf{x}(n))$ denotes a *nonlinear measurement matrix* that may be time-variant too.

As an example, consider the following two-dimensional nonlinear state-space model:

$$\begin{bmatrix} x_1(n+1) \\ x_2(n+1) \end{bmatrix} = \begin{bmatrix} x_1(n) + x_2^2(n) \\ nx_1(n) - x_1(n)x_2(n) \end{bmatrix} + \begin{bmatrix} v_{1,1}(n) \\ v_{1,2}(n) \end{bmatrix}$$

$$y(n) = x_1(n)x_2^2(n) + v_2(n)$$

In this example, we have

$$\mathbf{F}(n,\mathbf{x}(n)) = \begin{bmatrix} x_1(n) + x_2^2(n) \\ nx_1(n) - x_1(n)x_2(n) \end{bmatrix}$$

and

$$\mathbf{C}(n,\mathbf{x}(n)) = x_1(n)x_2^2(n)$$

The basic idea of the extended Kalman filter is to *linearize* the state-space model of Eqs. (7.111) and (7.112) at each time instant around the most recent state estimate, which is taken to be either $\hat{\mathbf{x}}(n|\mathcal{Y}_n)$ or $\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})$, depending on which particular functional is being considered. Once a linear model is obtained, the standard Kalman filter equations are applied.

More explicitly, the approximation proceeds in two stages.

Stage 1. The following two matrices are constructed

$$\mathbf{F}(n+1,n) = \left.\frac{\partial \mathbf{F}(n,\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(n|\mathcal{Y}_n)} \tag{7.113}$$

and

$$\mathbf{C}(n) = \left.\frac{\partial \mathbf{C}(n,\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})} \tag{7.114}$$

That is, the ijth entry of $\mathbf{F}(n+1,n)$ is equal to the partial derivative of the ith component of $\mathbf{F}(n,\mathbf{x})$ *with respect to the* jth component of $\mathbf{x}$. Likewise, the ijth entry of $\mathbf{C}(n)$ is equal to the partial derivative of the ith component of $\mathbf{C}(n,\mathbf{x})$ with respect to the jth component of $\mathbf{x}$. In the former case, the derivatives are evaluated at $\hat{\mathbf{x}}(n|\mathcal{Y}_n)$, while in the latter case

the derivatives are evaluated at $\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})$. The entries of the matrices $\mathbf{F}(n+1,n)$ and $\mathbf{C}(n)$ are all known (i.e., computable), since $\hat{\mathbf{x}}(n|\mathcal{Y}_n)$ and $\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})$ are made available as described later.

Applying the definitions of Eqs. (7.113) and (7.114) to the previous example, we get

$$\frac{\partial \mathbf{F}(n, \mathbf{x})}{\partial \mathbf{x}} = \begin{bmatrix} 1 & 2x_2 \\ n - x_2 & -x_1 \end{bmatrix}$$

$$\frac{\partial \mathbf{C}(n, \mathbf{x})}{\partial \mathbf{x}} = [x_2^2 \quad 2x_1x_2]$$

which leads to

$$\mathbf{F}(n+1, n) = \begin{bmatrix} 1 & 2\hat{x}_2(n|\mathcal{Y}_n) \\ n - \hat{x}_2(n|\mathcal{Y}_n) & -\hat{x}_1(n|\mathcal{Y}_n) \end{bmatrix}$$

and

$$\mathbf{C}(n) = [\hat{x}_2^2(n|\mathcal{Y}_{n-1}) \quad 2\hat{x}_1(n|\mathcal{Y}_{n-1})\,\hat{x}_2(n|\mathcal{Y}_{n-1})]$$

Stage 2. Once the matrices $\mathbf{F}(n+1,n)$ and $\mathbf{C}(n)$ are evaluated, they are then employed in a *first-order Taylor approximation* of the nonlinear functionals $\mathbf{F}(n,\mathbf{x}(n))$ and $\mathbf{C}(n,\mathbf{x}(n))$ around $\hat{\mathbf{x}}(n|\mathcal{Y}_n)$ and $\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})$, respectively. Specifically, $\mathbf{F}(n,\mathbf{x}(n))$ and $\mathbf{C}(n,\mathbf{x}(n))$ are approximated as follows, respectively:

$$\mathbf{F}(n,\mathbf{x}(n)) \simeq \mathbf{F}(n, \hat{\mathbf{x}}(n|\mathcal{Y}_n)) + \mathbf{F}(n+1,n)\,[\mathbf{x}(n) - \hat{\mathbf{x}}(n|\mathcal{Y}_n)] \tag{7.115}$$

$$\mathbf{C}(n, \mathbf{x}(n)) \simeq \mathbf{C}(n, \hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})) + \mathbf{C}(n)[\mathbf{x}(n) - \hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})] \tag{7.116}$$

With the above approximate expressions at hand, we may now proceed to approximate the nonlinear state-equations (7.111) and (7.112) as shown by, respectively,

$$\mathbf{x}(n+1) \simeq \mathbf{F}(n+1, n)\mathbf{x}(n) + \mathbf{v}_1(n) + \mathbf{d}(n) \tag{7.117}$$

$$\bar{\mathbf{y}}(n) \simeq \mathbf{C}(n)\mathbf{x}(n+\mathbf{v}_2(n) \tag{7.118}$$

where we have introduced two new quantities:

$$\bar{\mathbf{y}}(n) = \mathbf{y}(n) - [\mathbf{C}(n,\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})) - \mathbf{C}(n)\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})] \tag{7.119}$$

and

$$\mathbf{d}(n) = \mathbf{F}(n,\hat{\mathbf{x}}(n|\mathcal{Y}_n)) - \mathbf{F}(n+1,n)\hat{\mathbf{x}}(n|\mathcal{Y}_n) \tag{7.120}$$

The entries in the term $\bar{\mathbf{y}}(n)$ are all known at time n, and, therefore, $\bar{\mathbf{y}}(n)$ can be regarded as an observation vector at time n. Likewise, the entries in the term $\mathbf{d}(n)$ are all known at time n.

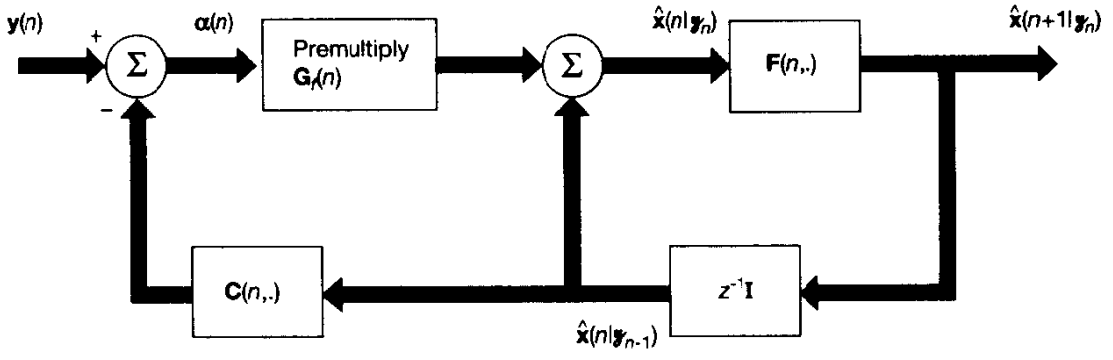

Figure 7.6 One-step predictor for the extended Kalman filter.

The approximate state-space model of Eqs. (7.117) and (7.118) is a linear model of the same mathematical form as that described in Eqs. (7.108) and (7.109); indeed, it is with this objective in mind that earlier on we formulated the state-space model of Eqs. (7.108) and (7.109). The extended Kalman filter equations simply correspond to applying the standard Kalman equations (7.103) through (7.109) and (7.110) to the above linear model. This leads to the following set of equations:

$$\begin{aligned}
\hat{\mathbf{x}}(n+1|\mathcal{Y}_n) &= \mathbf{F}(n+1,n)\hat{\mathbf{x}}(n|\mathcal{Y}_n) + \mathbf{d}(n) \\
&= \mathbf{F}(n+1,n)\hat{\mathbf{x}}(n|\mathcal{Y}_n) + [\mathbf{F}(n,\hat{\mathbf{x}}(n|\mathcal{Y}_n)) - \mathbf{F}(n+1,n)\hat{\mathbf{x}}(n|\mathcal{Y}_n)] \\
&= \mathbf{F}(n,\hat{\mathbf{x}}(n|\mathcal{Y}_n)) \qquad (7.121) \\
\hat{\mathbf{x}}(n|\mathcal{Y}_n) &= \hat{\mathbf{x}}(n|\mathcal{Y}_{n-1}) + \mathbf{G}_f(n)\boldsymbol{\alpha}(n) \\
\boldsymbol{\alpha}(n) &= \tilde{\mathbf{y}}(n) - \mathbf{C}(n)\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1}) \\
&= \mathbf{y}(n) - \mathbf{C}(n,\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})) + \mathbf{C}(n)\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1}) - \mathbf{C}(n)\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1}) \\
&= \mathbf{y}(n) - \mathbf{C}(n,\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})) \qquad (7.122)
\end{aligned}$$

On the basis of Eqs. (7.121) and (7.122), we may formulate the signal-flow graph of Fig. 7.6 for updating the one-step prediction in the extended Kalman filter.

In Table 7.5 we present a summary of the extended Kalman filtering algorithm, where the linearized matrices $\mathbf{F}(n+1, n)$ and $\mathbf{C}(n)$ are computed from their respective nonlinear counterparts using Eqs. (7.113) and (7.114). Given a nonlinear state-space model of the form described in Eqs. (7.111) and (7.112), we may thus use this algorithm to compute state estimates recursively. Comparing the equations of the extended Kalman filter

TABLE 7.5 SUMMARY OF THE EXTENDED KALMAN FILTER

Input vector process

Observations = $\{\mathbf{y}(1), \mathbf{y}(2), \ldots, \mathbf{y}(n)\}$

Known parameters

Nonlinear state transition matrix = $\mathbf{F}(n, \mathbf{x}(n))$

Nonlinear measurement matrix = $\mathbf{C}(n, \mathbf{x}(n))$

Correlation matrix of process noise vector = $\mathbf{Q}_1(n)$

Correlation matrix of measurement noise vector = $\mathbf{Q}_2(n)$

Computation: $n = 1, 2, 3, \ldots$.

$$\mathbf{G}_f(n) = \mathbf{K}(n, n-1)\mathbf{C}^H(n)[\mathbf{C}(n)\mathbf{K}(n, n-1)\mathbf{C}^H(n) + \mathbf{Q}_2(n)]^{-1}$$

$$\boldsymbol{\alpha}(n) = \mathbf{y}(n) - \mathbf{C}(n, \hat{\mathbf{x}}(n|\mathcal{Y}_{n-1}))$$

$$\hat{\mathbf{x}}(n|\mathcal{Y}_n) = \hat{\mathbf{x}}(n|\mathcal{Y}_{n-1}) + \mathbf{G}_f(n)\boldsymbol{\alpha}(n)$$

$$\hat{\mathbf{x}}(n+1|\mathcal{Y}_n) = \mathbf{F}(n, \hat{\mathbf{x}}(n|\mathcal{Y}_n))$$

$$\mathbf{K}(n) = [\mathbf{I} - \mathbf{G}_f(n)\mathbf{C}(n)]\mathbf{K}(n, n-1)$$

$$\mathbf{K}(n+1, n) = \mathbf{F}(n+1, n)\mathbf{K}(n)\mathbf{F}^H(n+1, n) + \mathbf{Q}_1(n)$$

Note: The linearized matrices $\mathbf{F}(n+1, n)$ and $\mathbf{C}(n)$ are computed from their nonlinear counterparts $\mathbf{F}(n, \mathbf{x}(n))$ and $\mathbf{C}(n, \mathbf{x}(n))$ using Eqs. (7.113) and (7.114), respectively.

Initial conditions

$$\hat{\mathbf{x}}(1|\mathcal{Y}_0) = E[\mathbf{x}(1)]$$

$$\mathbf{K}(1,0) = E[(\mathbf{x}(1) - E[\mathbf{x}(1)])(\mathbf{x}(1) - E[\mathbf{x}(1)])^H] = \boldsymbol{\Pi}_0$$

summarized herein with those of the standard Kalman filter given in Eqs. (7.102) through (7.107), we see that the only differences between them arise in the computations of the innovations vector $\boldsymbol{\alpha}(n)$ and the updated estimate $\hat{\mathbf{x}}(n+1|\mathcal{Y}_n)$. Specifically, the linear terms $\mathbf{F}(n+1, n)\hat{\mathbf{x}}(n|\mathcal{Y}_n)$ and $\mathbf{C}(n)\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})$ in the standard Kalman filter are replaced by the approximate terms $\mathbf{F}(n, \hat{\mathbf{x}}(n|\mathcal{Y}_n))$ and $\mathbf{C}(n, \hat{\mathbf{x}}(n|\mathcal{Y}_{n-1}))$, respectively, in the extended Kalman filter. These differences also show up in comparing the signal-flow graph of Fig. 7.3 for one-step prediction in the standard Kalman filter with that of Fig. 7.6 for one-step prediction in the extended Kalman filter.

7.10 SUMMARY AND DISCUSSION

The Kalman filter is a linear, discrete-time, finite-dimensional system, the implementation of which is well suited for a digital computer. A key property of the Kalman filter is that it leads to minimization of the trace of the filtered state error correlation matrix $\mathbf{K}(n)$. This, in turn, means that the Kalman filter is the *linear minimum variance estimator* of the state vector $\mathbf{x}(n)$ (Anderson and Moore, 1979; Goodwin and Sin, 1984).

The Kalman filter has been successfully applied to solve many real-world problems as can be seen in the literature on control systems (Sorenson, 1985). Moreover, the Kalman

filter provides the general framework for deriving *all* of the known algorithms that constitute the recursive least-squares family of adaptive filters (Sayed and Kailath, 1994). In the intervening two decades between the paper by Sayed and Kailath and the seminal paper by Godard in 1974, many attempts were made to incorporate this important family of adaptive filtering algorithms into the framework of Kalman filter theory. However, some annoying discrepancies always remained, thereby hindering the full application of the extensive control literature on Kalman filters to adaptive filtering problems. For the first time, the paper by Sayed and Kailath has shown us how to devise a state-space model for the adaptive filtering problem that is a perfect match for the application of Kalman filter theory. It has been said by many that many of the problems encountered in signal processing and control theory are mathematically equivalent. The link between Kalman filter theory and adaptive filter theory demonstrated in the paper by Sayed and Kailath is further testimony to the validity of this mathematical equivalence.

PROBLEMS

1. The Gram–Schmidt orthogonalization procedure enables the set of observation vectors $\mathbf{y}(1)$, $\mathbf{y}(2), \ldots, \mathbf{y}(n)$ to be transformed into the set of innovations processes $\boldsymbol{\alpha}(1), \boldsymbol{\alpha}(2), \ldots, \boldsymbol{\alpha}(n)$ without loss of information, and vice versa. Illustrate this procedure for $n = 2$, and comment on the procedure for $n > 2$.

2. The predicted state-error vector is defined by

$$\boldsymbol{\epsilon}(n, n-1) = \mathbf{x}(n) - \hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})$$

where $\hat{\mathbf{x}}(n|\mathcal{Y}_{n-1})$ is the minimum mean-square estimate of the state $\mathbf{x}(n)$, given the space $\mathcal{Y}_{n-1}$ that is spanned by the observed data $\mathbf{y}(1), \ldots, \mathbf{y}(n-1)$. Let $\mathbf{v}_1(n)$ and $\mathbf{v}_2(n)$ denote the process noise and measurement noise vectors, respectively. Show that $\boldsymbol{\epsilon}(n, n-1)$ is orthogonal to both $\mathbf{v}_1(n)$ and $\mathbf{v}_2(n)$; that is,

$$E[\boldsymbol{\epsilon}(n, n-1)\mathbf{v}_1^H(n)] = \mathbf{0}$$

and

$$E[\boldsymbol{\epsilon}(n, n-1)\mathbf{v}_2^H(n)] = \mathbf{0}$$

3. Consider a set of scalar observations $y(n)$ of zero mean, which is transformed into the corresponding set of innovations $\alpha(n)$ of zero mean and variance $\sigma_\alpha^2(n)$. Let the estimate of the state vector $\mathbf{x}(i)$, given this set of data, be expressed as

$$\hat{\mathbf{x}}(i|\mathcal{Y}_n) = \sum_{k=1}^{n} \mathbf{b}_i(k)\alpha(k)$$

where $\mathcal{Y}_n$ is the space spanned by $y(1), \ldots, y(n)$, and $\mathbf{b}_i(k)$, $k = 1, 2, \ldots, n$, is a set of vectors to be determined. The requirement is to choose the $\mathbf{b}_i(k)$ so as to minimize the expected value of the squared norm of the estimated state-error vector

$$\boldsymbol{\epsilon}(i|\mathcal{Y}_n) = \mathbf{x}(i) - \hat{\mathbf{x}}(i|\mathcal{Y}_n)$$

Show that this minimization yields the result

$$\hat{\mathbf{x}}(i|\mathcal{Y}_n) = \sum_{k=1}^{n} E[\mathbf{x}(i)\phi^*(k)]\phi(k)$$

where $\phi(k)$ is the normalized innovation

$$\phi(k) = \frac{\alpha(k)}{\sigma_\alpha(k)}$$

This result may be viewed as a special case of Eqs. (7.37) and (7.40).

4. The Kalman gain $\mathbf{G}(n)$ defined in Eq. (7.49) involves the inverse matrix $\mathbf{R}^{-1}(n)$. The matrix $\mathbf{R}(n)$ is itself defined in Eq. (7.35), reproduced here for convenience:

$$\mathbf{R}(n) = \mathbf{C}(n)\mathbf{K}(n, n-1)\mathbf{C}^H(n) + \mathbf{Q}_2(n)$$

The matrix $\mathbf{C}(n)$ is nonnegative definite but not necessarily nonsingular.

(a) Why is $\mathbf{R}(n)$ nonnegative definite?

(b) What prior condition would you impose on the matrix $\mathbf{Q}_2(n)$ to ensure that the inverse matrix $\mathbf{R}^{-1}(n)$ exists?

5. In many cases the predicted state-error correlation matrix $\mathbf{K}(n+1, n)$ converges to the steady-state value $\mathbf{K}$ as the number of iterations n approaches infinity. Show that the limiting value $\mathbf{K}$ satisfies the *algebraic Riccati equation*

$$\mathbf{K}\mathbf{C}^H(\mathbf{C}\mathbf{K}\mathbf{C}^H + \mathbf{Q}_2)^{-1}\mathbf{C}\mathbf{K} - \mathbf{Q}_1 = \mathbf{0}$$

where it is assumed that the state transition matrix equals the identity matrix, and the matrices $\mathbf{C}$, $\mathbf{Q}_1$, and $\mathbf{Q}_2$ are the limiting values of $\mathbf{C}(n)$, $\mathbf{Q}_1(n)$, and $\mathbf{Q}_2(n)$, respectively.

6. Consider a stochastic process $y(n)$ represented by an autoregressive-moving average (ARMA) model of order (1, 1):

$$y(n) + ay(n-1) = v(n) + bv(n-1)$$

where a and b are the ARMA parameters and $v(n)$ is a zero-mean white-noise process of variance σ^2.

(a) Show that a state-space representation of this model is

$$\mathbf{x}(n+1) = \begin{bmatrix} -a & 1 \\ 0 & 0 \end{bmatrix}\mathbf{x}(n) + \begin{bmatrix} 1 \\ b \end{bmatrix}v(n+1)$$

$$y(n) = [1 \quad 0]\mathbf{x}(n)$$

where $\mathbf{x}(n)$ is a 2-by-1 state vector.

(b) Assume the applicability of the algebraic Riccati equation described in Problem 5. Hence, show that the solution of this equation is

$$\mathbf{K} = \sigma^2\begin{bmatrix} 1+c & b \\ b & b^2 \end{bmatrix}$$

where c is a scalar that satisfies the second-order equation

$$c = (b-a)^2 + a^2c - \frac{(b-a-ac)^2}{1+c}$$

What are the two values of c that satisfy this equation? Determine the corresponding values of the matrix $\mathbf{K}$.

(c) Show that the Kalman gain is

$$\mathbf{G} = \frac{b - a - ac}{1 + c}\begin{bmatrix}1\\0\end{bmatrix}$$

Determine the values of $\mathbf{G}$ that correspond to the solutions for the scalar c found in part (b).

7. In this problem we consider the general case of *time-varying real-valued ARMA process* $y(n)$ described by the difference equation:

$$y(n) + \sum_{k=1}^{M} a_k(n)y(n-k) = \sum_{k=1}^{N} a_{M+k}(n)v(n-k) + v(n)$$

where $a_1(n), a_2(n), \ldots, a_M(n), a_{M+1}(n), a_{M+2}(n), \ldots, a_{M+N}(n)$ are the ARMA coefficients, the process $v(n)$ is the input, and process $y(n)$ is the output. The process $v(n)$ is a white Gaussian noise process of zero mean and variance σ^2. The ARMA coefficients are subject to random fluctuations, as shown in the model

$$a_k(n+1) = a_k(n) + w_k(n),\ k = 1, \ldots, M+N$$

where $w_k(n)$ is a zero-mean, white Gaussian noise process that is independent of $w_j(n)$ for $j \neq k$, and also independent of $v(n)$. The issue of interest is to provide a technique based on the Kalman filter for identifying the coefficients of the ARMA process. To do this, we define an $(M + N)$-dimensional state vector:

$$\mathbf{x}(n) = [a_1(n), \ldots, a_M(n), \ldots, a_{M+N}(n)]^T$$

We also define the measurement matrix (actually, a row vector):

$$\mathbf{C}(n) = [-y(n-1), \ldots, -y(n-M), v(n-1), \ldots, v(n-N)]$$

On this basis, do the following:

(a) Formulate the state-space equations for the ARMA process.

(b) Find an algorithm for computing the predicted value of the state vector $\mathbf{x}(n+1)$, given the observation $y(n)$.

(c) How would you initialize the algorithm in part (b)?

8. Consider a communication channel modeled as an FIR filter of known impulse response. The channel output $y(n)$ is defined by

$$y(n) = \mathbf{h}^T\mathbf{x}(n) + w(n)$$

where $\mathbf{h}$ is an M-by-1 vector representing the channel impulse response, $\mathbf{x}(n)$ is an M-by-1 vector representing the present value $u(n)$ of the channel input and $(M-1)$ previous transmissions, and $w(n)$ is a white Gaussian noise process of zero mean and variance σ_w^2. At time n, the channel input $u(n)$ consists of a coded binary sequence of zeros and ones, statistically independent of $w(n)$. This model suggests that we may view $\mathbf{x}(n)$ as a state vector, in which case the state equation is written as[4]

$$\mathbf{x}(n+1) = \mathbf{A}\mathbf{x}(n) + \mathbf{b}v(n)$$

[4]This problem is adapted from Lawrence and Kaufman (1971).

where $v(n)$ is a white Gaussian noise process of zero mean and variance σ_v^2 which is independent of $w(n)$. The matrix $\mathbf{A}$ is an M-by-M matrix whose ijth element is defined by

$$a_{ij} = \begin{cases} 1, & i = j + 1 \\ 0, & \text{otherwise} \end{cases}$$

The vector $\mathbf{b}$ is an M-by-1 vector whose ith element is defined by

$$b_i = \begin{cases} 1, & i = 1 \\ 0, & i = 2, \ldots, M \end{cases}$$

We may now state the problem: Given the foregoing channel model and a sequence $y(n)$ of noisy measurements made at the channel output, use the Kalman filter to construct an equalizer that yields a good estimate of the channel input $u(n)$ at some delayed time $(n + D)$, where $0 \leq D \leq M - 1$. Show that the equalizer so constructed is an IIR filter where coefficients are determined by two distinct sets of parameters: (a) the M-by-1 channel impulse response vector, and (b) the Kalman gain, which (in this problem) is an M-by-1 vector.

9. For the case when the transition matrix $\mathbf{F}(n + 1, n)$ is the identity matrix and the state noise vector is zero, show that the predicted state-error correlation matrix $\mathbf{K}(n + 1, n)$ and the filtered state-error correlation matrix $\mathbf{K}(n)$ are equal.

10. Using the initial conditions described in Eqs. (7.72) and (7.73), show that the resulting filtered estimate $\hat{\mathbf{x}}(n|\mathcal{Y}_n)$ produced by the Kalman filter is unbiased; that is,

$$E[\hat{\mathbf{x}}(n)|\mathcal{Y}(n)] = \mathbf{x}(n)$$

11. In the UD-factorization algorithm, the filtered state-error correlation matrix $\mathbf{K}(n)$ is expressed as follows

$$\mathbf{K}(n) = \mathbf{U}(n)\mathbf{D}(n)\mathbf{U}^H(n)$$

where $\mathbf{U}(n)$ is an upper triangular matrix with 1's along its main diagonal, and $\mathbf{D}(n)$ is a real diagonal matrix. Let λ_{max} and λ_{min} denote the maximum and minimum eigenvalues of the matrix $\mathbf{K}(n)$. Show that the condition number of the diagonal matrix $\mathbf{D}(n)$ is governed by

$$\chi(\mathbf{D}) \geq \frac{\lambda_{max}}{\lambda_{min}} = \chi(\mathbf{K})$$

12. Let $\chi(\mathbf{K})$ denote the condition number of the filtered state-error correlation matrix $\mathbf{K}(n)$, defined as the ratio of the largest eigenvalue λ_{max} to the smallest eigenvalue λ_{min}. Show that

$$\chi(\mathbf{K}) = (\chi(\mathbf{K}^{1/2}))^2$$

where $\mathbf{K}^{1/2}(n)$ is the square-root of $\mathbf{K}(n)$. What is the computational implication of this relation?

13. Consider the state-space model described in Eqs. (7.108) and (7.109). Show that the one-step prediction $\mathbf{x}(n + 1|\mathcal{Y}_n)$ of the state vector in this model is given by Eq. (7.110).

14. (a) Figures 7.3 and 7.6 are signal-flow graph representations of the one-step predictor for the standard Kalman filter and extended Kalman filter, respectively. Show that for a linear model of a dynamical system, these two representations are equivalent.

(b) Figure 7.5 shows a block diagram representation of the standard Kalman filter. How is this block diagram modified for representation of the extended Kalman filter?