PART II

THE KALMAN FILTER

Optimal State Estimation, First Edition. By Dan J. Simon
ISBN 0471708585 ©2006 John Wiley & Sons, Inc.

CHAPTER 5

The discrete-time Kalman filter

The Kalman filter in its various forms is clearly established as a fundamental tool for analyzing and solving a broad class of estimation problems.

—Leonard McGee and Stanley Schmidt [McG85]

This chapter forms the heart of this book. The earlier chapters were written only to provide the foundation for this chapter, and the later chapters are written only to expand and generalize the results given in this chapter.

As we will see in this chapter, the Kalman filter operates by propagating the mean and covariance of the state through time. Our approach to deriving the Kalman filter will involve the following steps.

1. We start with a mathematical description of a dynamic system whose states we want to estimate.

2. We implement equations that describe how the mean of the state and the covariance of the state propagate with time. These equations, derived in Chapter 4, themselves form a dynamic system.

3. We take the dynamic system that describes the propagation of the state mean and covariance, and implement the equations on a computer. These equations form the basis for the derivation of the Kalman filter because:

Optimal State Estimation, First Edition. By Dan J. Simon
ISBN 0471708585 ©2006 John Wiley & Sons, Inc.

   (a) The mean of the state is the Kalman filter estimate of the state.
   (b) The covariance of the state is the covariance of the Kalman filter state estimate.

4. Every time that we get a measurement, we update the mean and covariance of the state. This is similar to what we did in Chapter 3 where we used measurements to recursively update our estimate of a constant.

In Section 5.1, we derive the equations of the discrete-time Kalman filter. This includes several different-looking, but mathematically equivalent forms. Various books and papers that deal with Kalman filters present the filter equations in ways that appear very different from one another. It is not always obvious, but these different formulations are actually mathematically equivalent, and we will see this in Section 5.1. (Sections 9.1, 10.5.1, and 11.1 also derive alternate but equivalent formulations of the Kalman filter equations.) In Section 5.2, we will examine some of the theoretical properties of the Kalman filter. One remarkable aspect of the Kalman filter is that it is optimal in several different senses, as we will see in Section 5.2. In Section 5.3, we will see how the Kalman filter equations can be written with a single time update equation. Section 5.4 presents a way to obtain a closed-form equation for the time-varying Kalman filter for a scalar time-invariant system, and a way to quickly compute the steady-state Kalman filter. Section 5.5 looks at some situations in which the Kalman filter is unstable or gives state estimates that are not close to the true state. We will also look at some ways that instability and divergence can be corrected in the Kalman filter.

5.1 DERIVATION OF THE DISCRETE-TIME KALMAN FILTER

Suppose we have a linear discrete-time system given as follows:

$$\begin{aligned} x_k &= F_{k-1}x_{k-1} + G_{k-1}u_{k-1} + w_{k-1} \\ y_k &= H_k x_k + v_k \end{aligned} \tag{5.1}$$

The noise processes $\{w_k\}$ and $\{v_k\}$ are white, zero-mean, uncorrelated, and have known covariance matrices Q_k and R_k, respectively:

$$\begin{aligned} w_k &\sim (0, Q_k) \\ v_k &\sim (0, R_k) \\ E[w_k w_j^T] &= Q_k \delta_{k-j} \\ E[v_k v_j^T] &= R_k \delta_{k-j} \\ E[v_k w_j^T] &= 0 \end{aligned} \tag{5.2}$$

where δ_{k-j} is the Kronecker delta function; that is, $\delta_{k-j} = 1$ if $k = j$, and $\delta_{k-j} = 0$ if $k \neq j$. Our goal is to estimate the state x_k based on our knowledge of the system dynamics and the availability of the noisy measurements $\{y_k\}$. The amount of information that is available to us for our state estimate varies depending on the particular problem that we are trying to solve. If we have all of the measurements up to and including time k available for use in our estimate of x_k, then we can form an *a posteriori* estimate, which we denote as $\hat{x}_k^+$. The "+" superscript denotes that the estimate is *a posteriori*. One way to form the *a posteriori* state estimate is to compute the expected value of x_k conditioned on all of the measurements up to and including time k:

$$\hat{x}_k^+ = E[x_k|y_1, y_2, \cdots, y_k] = \text{ a posteriori estimate} \tag{5.3}$$

If we have all of the measurements before (but not including) time k available for use in our estimate of x_k, then we can form an *a priori* estimate, which we denote as $\hat{x}_k^-$. The "–" superscript denotes that the estimate is *a priori*. One way to form the *a priori* state estimate is to compute the expected value of x_k conditioned on all of the measurements before (but not including) time k:

$$\hat{x}_k^- = E[x_k|y_1, y_2, \cdots, y_{k-1}] = \text{ a priori estimate} \tag{5.4}$$

It is important to note that $\hat{x}_k^-$ and $\hat{x}_k^+$ are both estimates of the same quantity; they are both estimates of x_k. However, $\hat{x}_k^-$ is our estimate of x_k *before* the measurement y_k is taken into account, and $\hat{x}_k^+$ is our estimate of x_k *after* the measurement y_k is taken into account. We naturally expect $\hat{x}_k^+$ to be a better estimate than $\hat{x}_k^-$, because we use more information to compute $\hat{x}_k^+$:

$$\begin{aligned} \hat{x}_k^- &= \text{estimate of } x_k \text{ before we process the measurement at time } k \\ \hat{x}_k^+ &= \text{estimate of } x_k \text{ after we process the measurement at time } k \end{aligned} \tag{5.5}$$

If we have measurements after time k available for use in our estimate of x_k, then we can form a *smoothed* estimate. One way to form the smoothed state estimate is to compute the expected value of x_k conditioned on all of the measurements that are available:

$$\hat{x}_{k|k+N} = E[x_k|y_1, y_2, \cdots, y_k, \cdots, y_{k+N}] = \text{ smoothed estimate} \tag{5.6}$$

where N is some positive integer whose value depends on the specific problem that is being solved. If we want to find the best prediction of x_k more than one time step ahead of the available measurements, then we can form a *predicted* estimate. One way to form the predicted state estimate is to compute the expected value of x_k conditioned on all of the measurements that are available:

$$\hat{x}_{k|k-M} = E[x_k|y_1, y_2, \cdots, y_{k-M}] = \text{ predicted estimate} \tag{5.7}$$

where M is some positive integer whose value depends on the specific problem that is being solved. The relationship between the *a posteriori*, *a priori*, smoothed, and predicted state estimates is depicted in Figure 5.1.

In the notation that follows, we use $\hat{x}_0^+$ to denote our initial estimate of x_0 before any measurements are available. The first measurement is taken at time $k = 1$. Since we do not have any measurements available to estimate x_0, it is reasonable to form $\hat{x}_0^+$ as the expected value of the initial state x_0:

$$\hat{x}_0^+ = E(x_0) \tag{5.8}$$

We use the term P_k to denote the covariance of the estimation error. P_k^- denotes the covariance of the estimation error of $\hat{x}_k^-$, and P_k^+ denotes the covariance of the estimation error of $\hat{x}_k^+$:

$$\begin{aligned} P_k^- &= E[(x_k - \hat{x}_k^-)(x_k - \hat{x}_k^-)^T] \\ P_k^+ &= E[(x_k - \hat{x}_k^+)(x_k - \hat{x}_k^+)^T] \end{aligned} \tag{5.9}$$

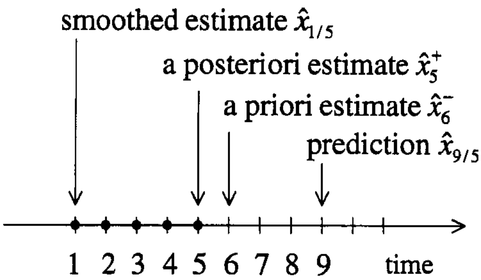
smoothed estimate $\hat{x}_{1/5}$
a posteriori estimate $\hat{x}_5^+$
a priori estimate $\hat{x}_6^-$
prediction $\hat{x}_{9/5}$
1 2 3 4 5 6 7 8 9
time

Figure 5.1 Time line showing the relationship between the *a posteriori*, *a priori*, smoothed, and predicted state estimates. In this figure, we suppose that we have received measurements at times up to and including $k = 5$. An estimate of the state at $k < 5$ is called a smoothed estimate. An estimate of the state at $k = 5$ is called the *a posteriori* estimate. An estimate of the state at $k = 6$ is called the *a priori* estimate. An estimate of the state at $k > 6$ is called the prediction.

These relationships are depicted in Figure 5.2. The figure shows that after we process the measurement at time $(k-1)$, we have an estimate of x_{k-1} (denoted $\hat{x}_{k-1}^+$) and the covariance of that estimate (denoted P_{k-1}^+). When time k arrives, before we process the measurement at time k we compute an estimate of x_k (denoted $\hat{x}_k^-$) and the covariance of that estimate (denoted P_k^-). Then we process the measurement at time k to refine our estimate of x_k. The resulting estimate of x_k is denoted $\hat{x}_k^+$, and its covariance is denoted P_k^+.

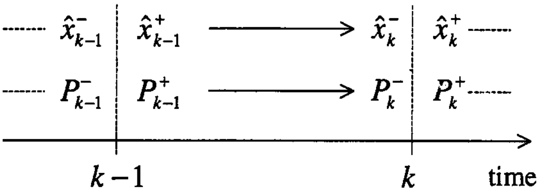

Figure 5.2 Timeline showing *a priori* and *a posteriori* state estimates and estimation-error covariances.

We begin the estimation process with $\hat{x}_0^+$, our best estimate of the initial state x_0. Given $\hat{x}_0^+$, how should we compute $\hat{x}_1^-$? We want to set $\hat{x}_1^- = E(x_1)$. But note that $\hat{x}_0^+ = E(x_0)$, and recall from Equation (4.2) how the mean of x propagates with time: $\bar{x}_k = F_{k-1}\bar{x}_{k-1} + G_{k-1}u_{k-1}$. We therefore obtain

$$\hat{x}_1^- = F_0\hat{x}_0^+ + G_0u_0 \tag{5.10}$$

This is a specific equation that shows how to obtain $\hat{x}_1^-$ from $\hat{x}_0^+$. However, the reasoning can be extended to obtain the following more general equation:

$$\hat{x}_k^- = F_{k-1}\hat{x}_{k-1}^+ + G_{k-1}u_{k-1} \tag{5.11}$$

This is called the time update equation for $\hat{x}$. From time $(k-1)^+$ to time k^-, the state estimate propagates the same way that the mean of the state propagates. This makes sense intuitively. We do not have any additional measurements available to help us update our state estimate between time $(k-1)^+$ and time k^-, so we should just update the state estimate based on our knowledge of the system dynamics.

Next we need to compute the time update equation for P, the covariance of the state estimation error. We begin with P_0^+, which is the covariance of our initial estimate of x_0. If we know the initial state perfectly, then $P_0^+ = 0$. If we have absolutely no idea of the value of x_0, then $P_0^+ = \infty I$. In general, P_0^+ represents the uncertainty in our initial estimate of x_0:

$$\begin{aligned} P_0^+ &= E[(x_0 - \bar{x}_0)(x_0 - \bar{x}_0)^T] \\ &= E[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T] \end{aligned} \tag{5.12}$$

Given P_0^+, how can we compute P_1^-? Recall from Equation (4.4) how the covariance of the state of a linear discrete-time system propagates with time: $P_k = F_{k-1}P_{k-1}F_{k-1}^T + Q_{k-1}$. We therefore obtain

$$P_1^- = F_0 P_0^+ F_0^T + Q_0 \tag{5.13}$$

This is a specific equation that shows how to obtain P_1^- from P_0^+. However, the reasoning can be extended to obtain the following more general equation:

$$P_k^- = F_{k-1}P_{k-1}^+F_{k-1}^T + Q_{k-1} \tag{5.14}$$

This is called the time-update equation for P.

We have derived the time-update equations for $\hat{x}$ and P. Now we need to derive the measurement-update equations for $\hat{x}$ and P. Given $\hat{x}_k^-$, how should we compute $\hat{x}_k^+$? The quantity $\hat{x}_k^-$ is an estimate of x_k, and the quantity $\hat{x}_k^+$ is also an estimate of x_k. The only difference between $\hat{x}_k^-$ and $\hat{x}_k^+$ is that $\hat{x}_k^+$ takes the measurement y_k into account. Recall from the recursive least squares development in Section 3.3 that the availability of the measurement y_k changes the estimate of a constant x as follows:

$$\begin{aligned} K_k &= P_{k-1}H_k^T(H_kP_{k-1}H_k^T + R_k)^{-1} \\ &= P_kH_k^TR_k^{-1} \\ \hat{x}_k &= \hat{x}_{k-1} + K_k(y_k - H_k\hat{x}_{k-1}) \\ P_k &= (I - K_kH_k)P_{k-1}(I - K_kH_k)^T + K_kR_kK_k^T \\ &= (P_{k-1}^{-1} + H_k^TR_k^{-1}H_k)^{-1} \\ &= (I - K_kH_k)P_{k-1} \end{aligned} \tag{5.15}$$

where $\hat{x}_{k-1}$ and P_{k-1} are the estimate and its covariance *before* the measurement y_k is processed, and $\hat{x}_k$ and P_k are the estimate and its covariance *after* the measurement y_k is processed. In this chapter, $\hat{x}_k^-$ and P_k^- are the estimate and its covariance before the measurement y_k is processed, and $\hat{x}_k^+$ and P_k^+ are the estimate and its covariance after the measurement y_k is processed. These relationships are shown in Table 5.1.[1]

We can now generalize from the formulas for the estimation of a constant in Section 3.3, to the measurement update equations required in this section. In

[1] We need to use minus and plus superscripts on $\hat{x}_k$ and P_k in order to distinguish between quantities before y_k is taken into account, and quantities after y_k is taken into account. In Chapter 3, we did not need superscripts because x was a constant.

Table 5.1 Relationships between estimates and covariances in Sections 3.3 and 5.1

Section 3.3 Least squares estimation		Section 5.1 Kalman filtering
$\hat{x}_{k-1}$ = estimate before y_k is processed	$\Longrightarrow$	$\hat{x}_k^-$ = *a priori* estimate
P_{k-1} = covariance before y_k is processed	$\Longrightarrow$	P_k^- = *a priori* covariance
$\hat{x}_k$ = estimate after y_k is processed	$\Longrightarrow$	$\hat{x}_k^+$ = *a posteriori* estimate
P_k = covariance after y_k is processed	$\Longrightarrow$	P_k^+ = *a posteriori* covariance

Equation (5.15), we replace $\hat{x}_{k-1}$ with $\hat{x}_k^-$, we replace P_{k-1} with P_k^-, we replace $\hat{x}_k$ with $\hat{x}_k^+$, and we replace P_k with P_k^+. This results in

$$\begin{aligned}
K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \\
&= P_k^+ H_k^T R_k^{-1} \\
\hat{x}_k^+ &= \hat{x}_k^- + K_k (y_k - H_k \hat{x}_k^-) \\
P_k^+ &= (I - K_k H_k) P_k^- (I - K_k H_k)^T + K_k R_k K_k^T \\
&= \left[(P_k^-)^{-1} + H_k^T R_k^{-1} H_k\right]^{-1} \\
&= (I - K_k H_k) P_k^- \qquad (5.16)
\end{aligned}$$

These are the measurement-update equations for $\hat{x}_k$ and P_k. The matrix K_k in the above equations is called the Kalman filter gain.

The discrete-time Kalman filter

Here we summarize the discrete-time Kalman filter by combining the above equations into a single algorithm.

1. The dynamic system is given by the following equations:

$$\begin{aligned}
x_k &= F_{k-1} x_{k-1} + G_{k-1} u_{k-1} + w_{k-1} \\
y_k &= H_k x_k + v_k \\
E(w_k w_j^T) &= Q_k \delta_{k-j} \\
E(v_k v_j^T) &= R_k \delta_{k-j} \\
E(w_k v_j^T) &= 0 \qquad (5.17)
\end{aligned}$$

2. The Kalman filter is initialized as follows:

$$\begin{aligned}
\hat{x}_0^+ &= E(x_0) \\
P_0^+ &= E[(x_0 - \hat{x}_0^+)(x_0 - \hat{x}_0^+)^T] \qquad (5.18)
\end{aligned}$$

3. The Kalman filter is given by the following equations, which are computed for each time step $k = 1, 2, \cdots$:

$$\begin{aligned}
P_k^- &= F_{k-1} P_{k-1}^+ F_{k-1}^T + Q_{k-1} \\
K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1}
\end{aligned}$$

$$\begin{aligned}
&= P_k^+ H_k^T R_k^{-1} \\
\hat{x}_k^- &= F_{k-1}\hat{x}_{k-1}^+ + G_{k-1}u_{k-1} = \text{ \textit{a priori} state estimate} \\
\hat{x}_k^+ &= \hat{x}_k^- + K_k(y_k - H_k\hat{x}_k^-) = \text{ \textit{a posteriori} state estimate} \\
P_k^+ &= (I - K_kH_k)P_k^-(I - K_kH_k)^T + K_kR_kK_k^T \\
&= \left[(P_k^-)^{-1} + H_k^T R_k^{-1} H_k\right]^{-1} \\
&= (I - K_kH_k)P_k^- \qquad (5.19)
\end{aligned}$$

The first expression for P_k^+ above is called the Joseph stabilized version of the covariance measurement update equation. It was formulated by Peter Joseph in the 1960s and can be shown to be more stable and robust than the third expression for P_k^+ [Buc68, Cra04] (see Problem 5.2). The first expression for P_k^+ guarantees that P_k^+ will always be symmetric positive definite, as long as P_k^- is symmetric positive definite. The third expression for P_k^+ is computationally simpler than the first expression, but its form does not guarantee symmetry or positive definiteness for P_k^+. The second form for P_k^+ is rarely implemented as written above but will be useful in our derivation of the information filter in Section 6.2.

If the second expression for K_k is used, then the second expression for P_k^+ must be used. This is because the second expression for K_k depends on P_k^+, so we need to use an expression for P_k^+ that does not depend on K_k.

Note that if x_k is a constant, then $F_k = I$, $Q_k = 0$, and $u_k = 0$. In this case, the Kalman filter of Equation (5.19) reduces to the recursive least squares algorithm for the estimation of a constant vector as given in Equation (3.47).

Finally we mention one more important practical aspect of the Kalman filter. We see from Equation (5.19) that the calculation of P_k^-, K_k, and P_k^+ does not depend on the measurements y_k, but depends only on the system parameters F_k, H_k, Q_k, and R_k. That means that the Kalman gain K_k can be calculated offline before the system operates and saved in memory. Then when it comes time to operate the system in real time, only the $\hat{x}_k$ equations need to be implemented in real time. The computational effort of calculating K_k can be saved during real-time operation by precomputing it. If the Kalman filter is implemented in an embedded system with strict computational requirements, this can make the difference between whether or not the system can be implemented in real time. Furthermore, the performance of the filter can be investigated and evaluated before the filter is actually run. This is because P_k indicates the estimation accuracy of the filter, and it can be computed offline since it does not depend on the measurements. In contrast, as we will see in Chapter 13, the filter gain and covariance for nonlinear systems cannot (in general) be computed offline because they depend on the measurements.

5.2 KALMAN FILTER PROPERTIES

In this section, we summarize some of the interesting and important properties of the Kalman filter. Suppose we are given the linear system of Equation (5.17) and we want to find a causal filter that results in a state estimate $\hat{x}_k$. The error between the true state and the estimated state is denoted as $\tilde{x}_k$:

$$\tilde{x}_k = x_k - \hat{x}_k \qquad (5.20)$$

Since the state is partly determined by the stochastic process $\{w_k\}$, x_k is a random variable. Since the state estimate is determined by the measurement sequence $\{y_k\}$, which in turn is partly determined by the stochastic process $\{v_k\}$, $\hat{x}_k$ is a random variable. Therefore, $\tilde{x}_k$ is also a random variable.

Suppose we want to find the estimator that minimizes (at each time step) a weighted two-norm of the expected value of the estimation error $\tilde{x}_k$:

$$\min E\left[\tilde{x}_k^T S_k \tilde{x}_k\right] \tag{5.21}$$

where S_k is a positive definite user-defined weighting matrix. If S_k is diagonal with elements $S_k(1), \cdots, S_k(n)$, then the weighted sum is equal to $S_k(1)E[\tilde{x}_k^2(1)] + \cdots + S_k(n)E[\tilde{x}_k^2(n)]$.

- If $\{w_k\}$ and $\{v_k\}$ are Gaussian, zero-mean, uncorrelated, and white, then the Kalman filter is the solution to the above problem.

- If $\{w_k\}$ and $\{v_k\}$ are zero-mean, uncorrelated, and white, then the Kalman filter is the best linear solution to the above problem. That is, the Kalman filter is the best filter that is a linear combination of the measurements. There may be a nonlinear filter that gives a better solution, but the Kalman filter is the best linear filter. It is often asserted in books and papers that the Kalman filter is not optimal unless the noise is Gaussian. However, as our derivation in this chapter has shown, that is simply untrue. Such statements arise from erroneous interpretations of Kalman filter derivations. Even if the noise is not Gaussian, the Kalman filter is still the optimal *linear* filter.

- If $\{w_k\}$ and $\{v_k\}$ are correlated or colored, then the Kalman filter can be modified to solve the above problem. This will be shown in Chapter 7.

- For nonlinear systems, various formulations of nonlinear Kalman filters approximate the solution to the above problem. This will be discussed further in Chapters 13–15.

Recall the measurement update equation from Equation (5.19):

$$\hat{x}_k^+ = \hat{x}_k^- + K_k(y_k - H_k\hat{x}_k^-) \tag{5.22}$$

The quantity $(y_k - H_k\hat{x}_k^-)$ is called the innovations. This is the part of the measurement that contains new information about the state. In Section 10.1, we will prove that the innovations is zero-mean and white with covariance $(H_kP_k^-H_k^T + R_k)$. In fact, the Kalman filter can actually be derived as a filter that whitens the measurement and hence extracts the maximum possible amount of information from the measurement. This was first proposed in [Kai68]. When a Kalman filter is used for state estimation, the innovations can be measured and its mean and covariance can be approximated using statistical methods. If the mean and covariance of the innovations are not as expected, that means something is wrong with the filter. Perhaps the assumed system model is incorrect, or the assumed noise statistics are incorrect. This can be used in real time to verify Kalman filter performance and parameters, and even to adjust Kalman filter parameters in order to improve performance. An application of this idea will be explored in Section 10.2.

5.3 ONE-STEP KALMAN FILTER EQUATIONS

In this section, we will see how the *a priori* and *a posteriori* Kalman filter equations can be combined into a single equation. This may simplify computer implementation of the equations. We start with the *a priori* state estimate expression from Equation (5.19), with the time index increased by one:

$$\hat{x}_{k+1}^- = F_k \hat{x}_k^+ + G_k u_k \tag{5.23}$$

Now take the *a posteriori* expression for $\hat{x}_k^+$ from Equation (5.19), and substitute it into the above equation to obtain

$$\begin{aligned} \hat{x}_{k+1}^- &= F_k \left[\hat{x}_k^- + K_k (y_k - H_k \hat{x}_k^-) \right] + G_k u_k \\ &= F_k (I - K_k H_k) \hat{x}_k^- + F_k K_k y_k + G_k u_k \end{aligned} \tag{5.24}$$

This shows that the *a priori* state estimate can be computed directly from its value at the previous time step, without computing the *a posteriori* state estimate in between. A similar procedure can be followed in order to obtain a one-step expression for the *a priori* covariance. We start with the *a priori* covariance expression from Equation (5.19), with the time index increased by one:

$$P_{k+1}^- = F_k P_k^+ F_k^T + Q_k \tag{5.25}$$

Now take the expression for P_k^+ from Equation (5.19), and substitute it into the above equation to obtain

$$\begin{aligned} P_{k+1}^- &= F_k (P_k^- - K_k H_k P_k^-) F_k^T + Q_k \\ &= F_k P_k^- F_k^T - F_k K_k H_k P_k^- F_k^T + Q_k \\ &= F_k P_k^- F_k^T - F_k P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} H_k P_k^- F_k^T + Q_k \end{aligned} \tag{5.26}$$

This equation, called a discrete Riccati equation, shows how P_{k+1}^- can be computed on the basis of P_k^- without an intermediate calculation of P_k^+.

Similar manipulations can be performed to obtain one-step expressions for the *a posteriori* state estimate and covariance. This results in

$$\begin{aligned} \hat{x}_k^+ &= (I - K_k H_k)(F_{k-1} \hat{x}_{k-1}^+ + G_{k-1} u_{k-1}) + K_k y_k \\ P_k^+ &= (I - K_k H_k)(F_{k-1} P_{k-1}^+ F_{k-1}^T + Q_{k-1}) \end{aligned} \tag{5.27}$$

One could imagine many different ways of combining the two expressions for K_k and the three expressions for P_k^+ in Equation (5.19). This would result in a number of different expressions for one-step updates for the *a priori* and *a posteriori* covariance.

■ EXAMPLE 5.1

Suppose we have a noise-free Newtonian system[2] with position r, velocity v, and constant acceleration a. The system can be described as

$$\begin{aligned} \begin{bmatrix} \dot{r} \\ \dot{v} \\ \dot{a} \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} r \\ v \\ a \end{bmatrix} \\ \dot{x} &= Ax \end{aligned} \tag{5.28}$$

The discretized version of this system (with a sample time of T) can be written as

$$x_{k+1} = Fx_k \tag{5.29}$$

where F is given as

$$\begin{aligned} F &= \exp(AT) \\ &= I + AT + \frac{(AT)^2}{2!} + \cdots \\ &= \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{5.30}$$

The Kalman filter for this system is

$$\begin{aligned} \hat{x}_k^- &= F\hat{x}_{k-1}^+ \\ P_k^- &= FP_{k-1}^+F^T + \underbrace{Q_{k-1}}_{0} \\ &= FP_{k-1}^+F^T \end{aligned} \tag{5.31}$$

We see that the covariance of the estimation error increases between time $(k-1)^+$ [that is, time $(k-1)$ after the measurement at that time is processed], and time k^- (i.e., time k before the measurement at that time is processed). Since we do not obtain any measurements between time $(k-1)^+$ and time k^-, it makes sense that our estimation uncertainty increases. Now suppose that we measure position with a variance of σ^2:

$$\begin{aligned} y_k &= H_k x_k + v_k \\ &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} x_k + v_k \\ v_k &\sim (0, R_k) \\ R_k &= \sigma^2 \end{aligned} \tag{5.32}$$

The Kalman gain can be obtained from Equation (5.19) as

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \tag{5.33}$$

If we write out the 3×3 matrix P_k^- in terms of its individual elements, and substitute for H_k and R_k in the above equation, we obtain

[2] The system described in this example is called Newtonian because it has its roots in the mathematical work of Isaac Newton. That is, velocity is the derivative of position, and acceleration is the derivative of velocity.

$$K_k = \begin{bmatrix} P_{k,11}^- \\ P_{k,12}^- \\ P_{k,13}^- \end{bmatrix} \frac{1}{P_{k,11}^- + \sigma^2} \tag{5.34}$$

The *a posteriori* covariance can be obtained from Equation (5.19) as

$$P_k^+ = P_k^- - K_k H_k P_k^- \tag{5.35}$$

If we write out the 3×3 matrix P_k^- in terms of its individual elements, and substitute for H_k and K_k in the above equation, we obtain

$$\begin{aligned} P_k^+ &= P_k^- - \frac{1}{P_{k,11}^- + \sigma^2} \begin{bmatrix} P_{k,11}^- & 0 & 0 \\ P_{k,12}^- & 0 & 0 \\ P_{k,13}^- & 0 & 0 \end{bmatrix} P_k^- \\ &= P_k^- - \frac{1}{P_{k,11}^- + \sigma^2} \begin{bmatrix} (P_{k,11}^-)^2 & P_{k,11}^- P_{k,21}^- & P_{k,11}^- P_{k,31}^- \\ P_{k,12}^- P_{k,11}^- & (P_{k,12}^-)^2 & P_{k,12}^- P_{k,31}^- \\ P_{k,13}^- P_{k,11}^- & P_{k,13}^- P_{k,12}^- & (P_{k,13}^-)^2 \end{bmatrix} \end{aligned} \tag{5.36}$$

We will use this expression to show that from time k^- to time k^+ the trace of the estimation-error covariance decreases. To see this first note that the trace of P_k^- is given as

$$\text{Tr}(P_k^-) = P_{k,11}^- + P_{k,22}^- + P_{k,33}^- \tag{5.37}$$

From Equation (5.36) we see that the trace of P_k^+ is given as

$$\begin{aligned} \text{Tr}(P_k^+) &= P_{k,11}^+ + P_{k,22}^+ + P_{k,33}^+ \\ &= \left(P_{k,11}^- - \frac{(P_{k,11}^-)^2}{P_{k,11}^- + \sigma^2} \right) + \left(P_{k,22}^- - \frac{(P_{k,12}^-)^2}{P_{k,11}^- + \sigma^2} \right) + \\ & \quad \left(P_{k,33}^- - \frac{(P_{k,13}^-)^2}{P_{k,11}^- + \sigma^2} \right) \\ &= \text{Tr}(P_k^-) - \frac{(P_{k,11}^-)^2 + (P_{k,12}^-)^2 + (P_{k,13}^-)^2}{P_{k,11}^- + \sigma^2} \end{aligned} \tag{5.38}$$

When we get a new measurement, we expect our state estimate to improve. That is, we expect the covariance to decrease, and the above equation shows that it does indeed decrease. That is, the trace of P_k^+ is less than the trace of P_k^-.

This system was simulated with five time units between discretization steps ($T = 5$), and a position-measurement standard deviation of 30 units. Figure 5.3 shows the variance of the position estimate ($P_{k,11}^-$ and $P_{k,11}^+$) for the first five time steps of the Kalman filter. It can be seen that the variance (uncertainty) increases from one time step to the next, but then decreases at each time step as the measurement is processed.

Figure 5.4 shows the variance of the position estimate ($P_{k,11}^-$ and $P_{k,11}^+$) for the first 60 time steps of the Kalman filter. This shows that the variance increases between time steps, and then decreases at each time step. But it

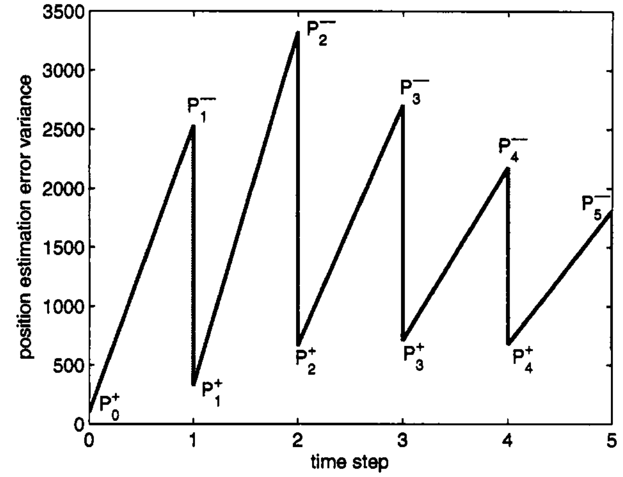

Figure 5.3 The first five time steps of the *a priori* and *a posteriori* position-estimation-error variances for Example 5.1.

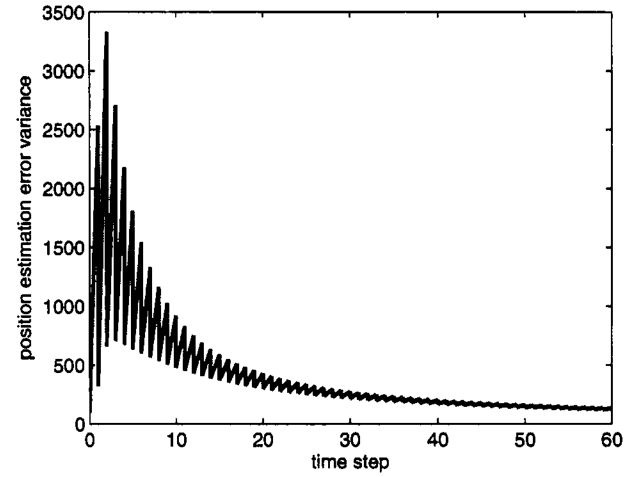

Figure 5.4 The first 60 time steps of the *a priori* and *a posteriori* position-estimation-error variances for Example 5.1.

can also be seen from this figure that the variance converges to a steady-state value.

Figure 5.5 shows the position-measurement error (with a standard deviation of 30) and the error of the *a posteriori* position estimate. The estimation error starts out with a standard deviation close to 30, but by the end of the simulation the standard deviation is about 11.

∇∇∇

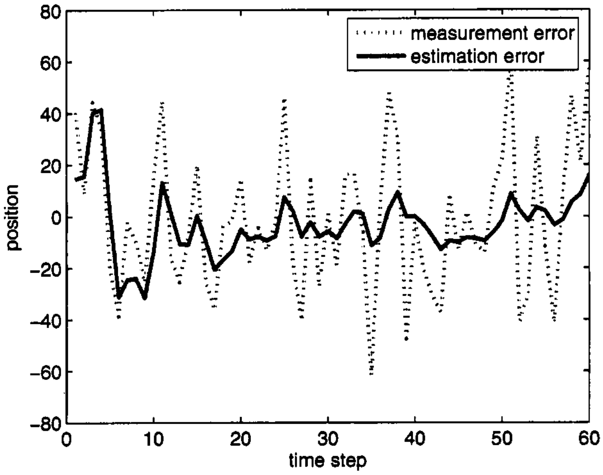

Figure 5.5 The position-measurement error and position estimation error for Example 5.1.

5.4 ALTERNATE PROPAGATION OF COVARIANCE

In this section, we derive an alternate equation for the propagation of the estimation-error covariance P. This alternate equation, based on [Gre01], can be used to find a closed-form equation for a scalar Kalman filter.[3] It can also be used to find a fast solution to the steady-state estimation-error covariance.

5.4.1 Multiple state systems

Recall from Equation (5.19) the update equations for the estimation-error covariance:

$$\begin{aligned} P_k^- &= F_{k-1}P_{k-1}^+F_{k-1}^T + Q_{k-1} \\ P_k^+ &= P_k^- - P_k^-H_k^T(H_kP_k^-H_k^T + R_k)^{-1}H_kP_k^- \end{aligned} \tag{5.39}$$

If the $n \times n$ matrix P_k^- can be factored as

$$P_k^- = A_kB_k^{-1} \tag{5.40}$$

where A_k and B_k are $n \times n$ matrices to be determined, then P_{k+1}^- satisfies

$$P_{k+1}^- = A_{k+1}B_{k+1}^{-1} \tag{5.41}$$

where A and B are propagated as follows:

$$\begin{bmatrix} A_{k+1} \\ B_{k+1} \end{bmatrix} = \begin{bmatrix} (F_k + Q_kF_k^{-T}H_k^TR_k^{-1}H_k) & Q_kF_k^{-T} \\ F_k^{-T}H_k^TR_k^{-1}H_k & F_k^{-T} \end{bmatrix} \begin{bmatrix} A_k \\ B_k \end{bmatrix} \tag{5.42}$$

[3] The equations given in [Gre01] have some typographical errors that have been corrected in this section.

This can be seen by noting from Equation (5.42) that

$$\begin{aligned} B_{k+1}^{-1} &= [F_k^{-T}H_k^TR_k^{-1}H_kA_k + F_k^{-T}B_k]^{-1} \\ &= \left[F_k^{-T}(H_k^TR_k^{-1}H_kA_kB_k^{-1} + I)B_k\right]^{-1} \\ &= B_k^{-1}[H_k^TR_k^{-1}H_kA_kB_k^{-1} + I]^{-1}F_k^T \end{aligned} \tag{5.43}$$

From Equation (5.42) we see that

$$A_{k+1}B_{k+1}^{-1} = [(F_k + Q_kF_k^{-T}H_k^TR_k^{-1}H_k)A_k + Q_kF_k^{-T}B_k]B_{k+1}^{-1} \tag{5.44}$$

Substituting the expression for B_{k+1}^{-1} into this equation gives

$$\begin{aligned} A_{k+1}B_{k+1}^{-1} &= [(F_k + Q_kF_k^{-T}H_k^TR_k^{-1}H_k)A_k + Q_kF_k^{-T}B_k] \times \\ &\quad B_k^{-1}[H_k^TR_k^{-1}H_kA_kB_k^{-1} + I]^{-1}F_k^T \\ &= [(F_k + Q_kF_k^{-T}H_k^TR_k^{-1}H_k)A_kB_k^{-1} + Q_kF_k^{-T}] \times \\ &\quad [H_k^TR_k^{-1}H_kA_kB_k^{-1} + I]^{-1}F_k^T \end{aligned} \tag{5.45}$$

Substituting P_k^- for $A_kB_k^{-1}$ in the above equation gives

$$\begin{aligned} A_{k+1}B_{k+1}^{-1} &= [(F_k + Q_kF_k^{-T}H_k^TR_k^{-1}H_k)P_k^- + Q_kF_k^{-T}] \times \\ &\quad [H_k^TR_k^{-1}H_kP_k^- + I]^{-1}F_k^T \\ &= [F_kP_k^- + Q_kF_k^{-T}(H_k^TR_k^{-1}H_kP_k^- + I)] \times \\ &\quad [H_k^TR_k^{-1}H_kP_k^- + I]^{-1}F_k^T \\ &= F_kP_k^-[H_k^TR_k^{-1}H_kP_k^- + I]^{-1}F_k^T + Q_kF_k^{-T}F_k^T \end{aligned} \tag{5.46}$$

Applying the matrix inversion lemma to the term in brackets gives

$$\begin{aligned} A_{k+1}B_{k+1}^{-1} &= F_kP_k^-[I - H_k^T(H_kP_k^-H_k^T + R_k)^{-1}H_kP_k^-]F_k^T + Q_k \\ &= F_k[P_k^- - P_k^-H_k^T(H_kP_k^-H_k^T + R_k)^{-1}H_kP_k^-]F_k^T + Q_k \\ &= F_kP_k^+F_k^T + Q_k \\ &= P_{k+1}^- \end{aligned} \tag{5.47}$$

So we see that $A_{k+1}B_{k+1}^{-1} = P_{k+1}^-$.

Equation (5.42) can be used to obtain a quick solution to the steady-state covariance for multidimensional systems (although not a closed-form solution). Suppose that F, Q, H, and R are constant matrices. From Equation (5.42) we obtain

$$\begin{aligned} \begin{bmatrix} A_{k+1} \\ B_{k+1} \end{bmatrix} &= \begin{bmatrix} (F + QF^{-T}H^TR^{-1}H) & QF^{-T} \\ F^{-T}H^TR^{-1}H & F^{-T} \end{bmatrix} \begin{bmatrix} A_k \\ B_k \end{bmatrix} \qquad (5.48) \\ &= \Psi \begin{bmatrix} A_k \\ B_k \end{bmatrix} \\ \begin{bmatrix} A_k \\ B_k \end{bmatrix} &= \Psi^{k-1} \begin{bmatrix} P_1^- \\ I \end{bmatrix} \end{aligned}$$

where we used the fact that $A_1 = P_1^-$ and $B_1 = I$ satisfies the original factoring of Equation (5.40). Now we can successively square Ψ a total of p times to obtain Ψ^2, Ψ^4, Ψ^8, and so on, until Ψ^{2^p} converges to a steady-state value:

$$\begin{bmatrix} A_\infty \\ B_\infty \end{bmatrix} \approx \Psi^{2^p} \begin{bmatrix} P_1^- \\ I \end{bmatrix} \quad \text{for large } p \tag{5.49}$$

The steady-state covariance is $P_\infty^- = A_\infty B_\infty^{-1}$. We can also find the steady-state Kalman gain by simply iterating the filter equations from Equation (5.19), but the method in this section could be a much quicker way to find the steady-state gain. Once we find P_∞^- as shown above, we compute $K_\infty = P_\infty^- H^T(HP_\infty^- H^T + R)^{-1}$ as the steady-state Kalman filter gain. More discussion of steady-state Kalman filtering is given in Section 7.3.

5.4.2 Scalar systems

Equation (5.42) can be used to obtain a closed-form solution for the scalar Kalman filter for time-invariant systems. Suppose that F, Q, H, and R are constant scalars. Then from Equation (5.42) we obtain

$$\begin{aligned} \begin{bmatrix} A_{k+1} \\ B_{k+1} \end{bmatrix} &= \begin{bmatrix} F + \frac{H^2Q}{FR} & \frac{Q}{F} \\ \frac{H^2}{FR} & \frac{1}{F} \end{bmatrix} \begin{bmatrix} A_k \\ B_k \end{bmatrix} \\ &= \Psi \begin{bmatrix} A_k \\ B_k \end{bmatrix} \end{aligned} \tag{5.50}$$

where Ψ is defined by the above equation. Now find the eigendata of Ψ. Suppose that the eigenvalues of Ψ are λ_1 and λ_2, and the eigenvectors of Ψ are combined to create the 2×2 matrix M. Then

$$\Psi = M \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} M^{-1} \tag{5.51}$$

and we obtain

$$\begin{aligned} \begin{bmatrix} A_k \\ B_k \end{bmatrix} &= \Psi^{k-1} \begin{bmatrix} A_1 \\ B_1 \end{bmatrix} \\ &= M \begin{bmatrix} \lambda_1^{k-1} & 0 \\ 0 & \lambda_2^{k-1} \end{bmatrix} M^{-1} \begin{bmatrix} P_1^- \\ 1 \end{bmatrix} \end{aligned} \tag{5.52}$$

where we used the fact that $A_1 = P_1^-$ and $B_1 = 1$ satisfies the original factoring of Equation (5.40). Working through the math to obtain λ_1, λ_2, and M gives the following.

$$\begin{aligned} P_k^- &= \frac{\tau_1 \mu_1^{k-1}(2RH^2P_1^- - \tau_2) - \tau_2 \mu_2^{k-1}(2H^2P_1^- - \tau_1)}{2H^2\mu_1^{k-1}(2RH^2P_1^- - \tau_2) - 2H^2\mu_2^{k-1}(2H^2P_1^- - \tau_1)} \\ \lambda_1 &= \frac{H^2Q + R(F^2+1) + \sigma}{2FR} \\ \lambda_2 &= \frac{H^2Q + R(F^2+1) - \sigma}{2FR} \\ \sigma &= \sqrt{H^2Q + R(F+1)^2}\sqrt{H^2Q + R(F-1)^2} \\ \tau_1 &= H^2Q + R(F^2-1) + \sigma \\ \tau_2 &= H^2Q + R(F^2-1) - \sigma \\ \mu_1 &= H^2Q + R(F^2+1) + \sigma \\ \mu_2 &= H^2Q + R(F^2+1) - \sigma \end{aligned}$$

$$\begin{aligned} M &= \begin{bmatrix} \frac{\tau_1}{2H^2} & \frac{\tau_2}{2H^2} \\ 1 & 1 \end{bmatrix} \\ M^{-1} &= \frac{1}{\tau_1(R-1)+2\sigma} \begin{bmatrix} 2RH^2 & -\tau_1 \\ -2RH^2 & R\tau_1 \end{bmatrix} \end{aligned} \tag{5.53}$$

This is a closed-form equation for the time-varying Kalman filter for a scalar time-invariant system. This can easily be used to obtain the steady-state value of P_k^-. Note that $\mu_2 < \mu_1$. As k increases, μ_2^k gets smaller and smaller relative to μ_1^k. Therefore

$$\begin{aligned} \lim_{k\to\infty} P_k^- &= \lim_{k\to\infty} \frac{\tau_1\mu_1^{k-1}(2RH^2P_1^- - \tau_2) - \tau_2\mu_2^{k-1}(2H^2P_1^- - \tau_1)}{2H^2\mu_1^{k-1}(2RH^2P_1^- - \tau_2) - 2H^2\mu_2^{k-1}(2H^2P_1^- - \tau_1)} \\ &= \lim_{k\to\infty} \frac{\tau_1\mu_1^{k-1}(2RH^2P_1^- - \tau_2)}{2H^2\mu_1^{k-1}(2RH^2P_1^- - \tau_2)} \\ &= \frac{\tau_1}{2H^2} \end{aligned} \tag{5.54}$$

This gives the steady-state covariance for a scalar system.

■ EXAMPLE 5.2

In this example, we will show how a scalar covariance can be propagated. Consider the following scalar system:

$$\begin{aligned} x_{k+1} &= x_k + w_k \\ y_k &= x_k + v_k \\ w_k &\sim (0,1) \\ v_k &\sim (0,1) \end{aligned} \tag{5.55}$$

This is a very simple system but one that arises in many applications. For example, it may represent some slowly varying parameter x_k that we measure directly. The process noise term w_k accounts for the variations in x_k, and the measurement noise term v_k accounts for measurement errors. In this system, we have $F = H = Q = R = 1$. Substituting these values in Equation (5.53) gives

$$\begin{aligned} \tau_1 &= 1+\sqrt{5} \\ \tau_2 &= 1-\sqrt{5} \\ \mu_1 &= 3+\sqrt{5} \\ \mu_2 &= 3-\sqrt{5} \\ P_k^- &= \frac{\tau_1\mu_1^{k-1}(2P_1^- - \tau_2) - \tau_2\mu_2^{k-1}(2P_1^- - \tau_1)}{2\mu_1^{k-1}(2P_1^- - \tau_2) - 2\mu_2^{k-1}(2P_1^- - \tau_1)} \end{aligned} \tag{5.56}$$

Taking the limit as $k \to \infty$ gives the steady-state value of P_k^-:

$$\begin{aligned} P_\infty^- &= \frac{\tau_1}{2} \\ &= \frac{1+\sqrt{5}}{2} \\ &\approx 1.62 \end{aligned} \tag{5.57}$$

Now we can use Equation (5.19) to find the steady-state value of K_k:

$$
\begin{aligned}
K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \\
&= \frac{P_k^-}{P_k^- + 1} \\
K_\infty &= \frac{1+\sqrt{5}}{3+\sqrt{5}} \\
&\approx 0.62
\end{aligned} \tag{5.58}
$$

Figure 5.6 shows the *a priori* estimation covariance and the Kalman gain as a function of time, and illustrates their convergence to steady-state values. From the equation for the *a posteriori* estimation covariance, we know that $P_k^+ = (I - K_k H_k) P_k^-$. For this example we therefore see that the steady-state value of P_k^+ is given as

$$
\begin{aligned}
P_\infty^+ &= \left(1 - \frac{1+\sqrt{5}}{3+\sqrt{5}}\right) \frac{1+\sqrt{5}}{2} \\
&= \frac{1+\sqrt{5}}{3+\sqrt{5}}
\end{aligned} \tag{5.59}
$$

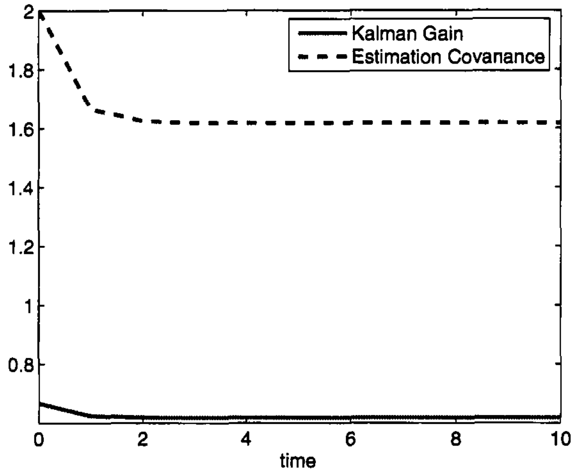

Figure 5.6 Estimation covariance and Kalman gain as a function of time for Example 5.2. The covariance and gain converge to steady-state values.

▽▽▽

5.5 DIVERGENCE ISSUES

The theory presented in this chapter makes the Kalman filter an attractive choice for state estimation. But when a Kalman filter is implemented on a real system it may not work, even though the theory is correct. Two of the primary causes for the failure of Kalman filtering are finite precision arithmetic and modeling errors [Fit71].

The theory presented in this chapter assumes that the Kalman filter arithmetic is infinite precision. In digital microprocessors the arithmetic is finite precision – only a certain number of bits are used to represent the numbers in the Kalman filter equations. This may cause divergence or even instability in the implementation of the Kalman filter.

The theory presented also assumes that the system model is precisely known. It is assumed that the F, Q, H, and R matrices are exactly known, and it is assumed that the noise sequences $\{w_k\}$ and $\{v_k\}$ are pure white, zero-mean, and completely uncorrelated. If any of these assumptions are violated, as they always are in real implementations, then the Kalman filter assumptions are violated and the theory may not work.

In order to improve filter performance in the face of these realities, the designer can use several strategies:

1. Increase arithmetic precision
2. Use some form of square root filtering
3. Symmetrize P at each time step: $P = (P + P^T)/2$
4. Initialize P appropriately to avoid large changes in P
5. Use a fading-memory filter
6. Use fictitious process noise (especially for estimating "constants")

These strategies are often problem dependent and need to be explored via simulation or experimentation in order to obtain good results. Some of these strategies may be more attractive than others, depending on the specific problem.

Item 1 above, increasing arithmetic precision, simply forces the digital implementation of the filter to more closely match the analog theory. In a PC-based implementation, it may require only a trivial effort to increase the arithmetic precision – change all the variables to double precision. This trivial change may make the difference between divergence and convergence. However, in a microcontroller implementation it may not be feasible to increase the arithmetic precision.

Item 2 above, square root filtering, is a way of reformulating the filter equations. Even though the physical precision of the implementation does not change, square root filtering effectively increases arithmetic precision. This will be discussed further in Sections 6.3, 6.4, and 8.3. But square root filtering requires more computational effort, which may or may not be a major consideration for a given application. Square root filtering also adds a lot of complication to the filter equations, which invites software bugs.

Items 3 and 4 above involve forcing P to be symmetric and initializing P appropriately. These are easy solutions, but they usually do not result in major improvements to the convergence properties of the filter. However, these steps should always be implemented since they are straightforward and easy, and since they may prevent numerical problems. Note from Equation (5.19) that the P_k^- expression is already symmetric, and so there is no point to forcing symmetry for P_k^-. However, depending on which equation is used, P_k^+ may or may not be symmetric. The expressions for P_k^+ in Equation (5.19) are mathematically equivalent, but they are not numerically equivalent. One of them has a built-in symmetry, but the others do not. If an equation for P_k^+ is used that does not have a built-in symmetry, then it is very easy and may pay large dividends to force symmetry. This has been done several different ways in the literature. One way is as described in Item 3 above; that is, after P is calculated, set $P = (P + P^T)/2$. Other ways involve forcing the terms below the diagonal to be equal to the terms above the diagonal, or forcing the eigenvalues of P to be positive.

Item 5 above is a simple way of forcing the filter to "forget" measurements in the distant past and place more emphasis on recent measurements. This causes the filter to be more responsive to measurements. It theoretically results in the loss of optimality of the Kalman filter, but it may restore convergence and stability. It is better to have a theoretically suboptimal filter that works rather than a theoretically optimal filter that does not work due to modeling errors. The greater responsiveness of the fading-memory filter to recent measurements makes the filter less sensitive to modeling errors, and hence more robust. This approach will be discussed further in Section 7.4.

Item 6 above, the use of fictitious process noise, is also easy to implement. In fact, it can be implemented in a way that is mathematically equivalent to the fading-memory filter of Item 5. Adding fictitious process noise is a way of telling the filter that you have less confidence in your system model. This causes the filter to place more emphasis on the measurements, and less emphasis on the process model (which may be incorrect) [Jaz69].

■ EXAMPLE 5.3

Let us illustrate the use of fictitious process noise with an example. Suppose we are trying to estimate a state that we think is a constant, but in reality is a ramp. In other words, we have a modeling error. Our assumed (but incorrect) model, upon which we base the Kalman filter, is given as follows:

$$\begin{aligned} x_{k+1} &= x_k + w_k \\ y_k &= x_k + v_k \\ w_k &\sim (0,0) \\ v_k &\sim (0,1) \end{aligned} \tag{5.60}$$

The assumed process noise is zero, which means that we are modeling x_k as a constant. From Equation (5.19) we derive the Kalman filter equations for this system as

$$\begin{aligned} P_k^- &= F_{k-1}P_{k-1}^+F_{k-1}^T + Q_{k-1} \\ &= P_{k-1}^+ \\ K_k &= P_k^-H_k^T(H_kP_k^-H_k^T + R_k)^{-1} \\ &= \frac{P_k^-}{P_k^- + 1} \\ \hat{x}_k^- &= F_{k-1}\hat{x}_{k-1}^+ \\ &= \hat{x}_{k-1}^+ \end{aligned}$$

$$\begin{aligned}
\hat{x}_k^+ &= \hat{x}_k^- + K_k(y_k - H_k\hat{x}_k^-) \\
&= \hat{x}_k^- + K_k(y_k - \hat{x}_k^-) \\
P_k^+ &= (I - K_kH_k)P_k^-(I - K_kH_k)^T + K_kR_kK_k^T \\
&= (1 - K_k)^2P_k^- + K_k^2 \qquad (5.61)
\end{aligned}$$

Suppose that the true system, although unknown to the Kalman filter designer, is given as the following two-state model:

$$\begin{aligned}
x_{1,k+1} &= x_{1,k} + x_{2,k} \\
x_{2,k+1} &= x_{2,k} \\
y_k &= x_{1,k} + v_k \\
v_k &\sim (0,1) \qquad (5.62)
\end{aligned}$$

The first state is a ramp, which we assumed incorrectly in our system model to be a constant. Figure 5.7 shows the true state $x_{1,k}$ and the estimated state $\hat{x}_{1,k}$. It can be seen that the estimate is diverging from the true state, and the estimation error is growing without bound.

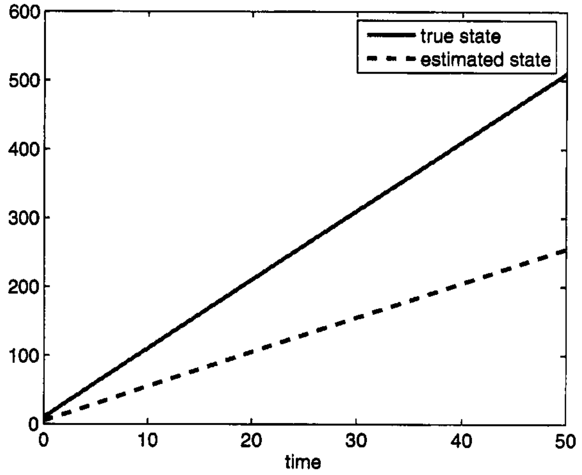

Figure 5.7 Kalman filter divergence due to mismodeling.

However, if we add fictitious process noise to the Kalman filter, then the filter will place more emphasis on the measurements, which will improve the filter performance. Figure 5.8 shows the true state and the estimated state when various values of Q are used in the Kalman filter. As the fictitious process noise gets larger, the estimation error becomes smaller. Of course, this is at the price of poorer performance in case the assumed system model is actually correct. The designer needs to add an appropriate amount of fictitious process noise to balance performance under nominal conditions with performance under mismodel conditions.

Figure 5.9 shows the time history of the Kalman gain K_k for this example for various values of Q. As expected, the gain K_k converges to a larger steady-state value when Q is larger, making the filter more responsive to

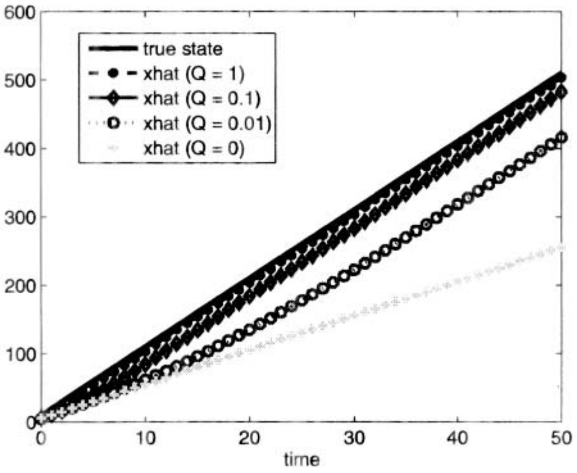

Figure 5.8 Kalman filter improvement due to fictitious process noise.

measurements [see the $\hat{x}_k^+$ expression in Equation (5.61)]. This compensates for modeling errors. As shown later in Section 7.4, the fading-memory filter accomplishes the same thing in a different way. Also note from Figure 5.9 that the steady-state Kalman gain is approximately 0.62 when $Q = 1$. This matches the results of Example 5.2.

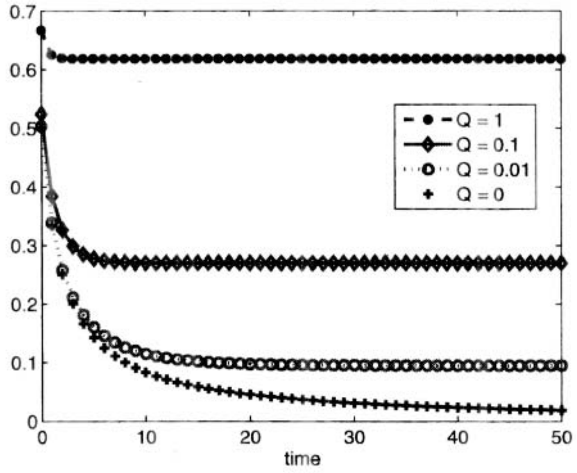

Figure 5.9 Kalman gain for various values of process noise.

This example illustrates the general principle that model noise is good, but only to a certain extent. If a system model has too much noise then it is difficult to estimate its state. But if a system model has too little noise then our state estimator might be overly susceptible to modeling errors.[4] When designing a model for a Kalman filter, we need to balance our confidence in our model (low noise resulting in close model tracking; i.e., low bandwidth) with a healthy self-doubt (high noise resulting in filter responsiveness; i.e., high bandwidth).

$\nabla\nabla\nabla$

Examination of the filter equations shows why adding fictitious process noise compensates for modeling errors. Recall the Kalman filter equations from Equation (5.19), some of which we repeat here:

$$\begin{aligned} P_k^- &= F_{k-1}P_{k-1}^+F_{k-1}^T + Q_{k-1} \\ K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \\ \hat{x}_k^+ &= \hat{x}_k^- + K_k(y_k - H_k \hat{x}_k^-) \end{aligned} \tag{5.63}$$

If Q_k is small then the covariance may not increase very much between time samples. In Example 5.3 we had $F_k = 1$, so $P_k^- = P_{k-1}^+$ when $Q_k = 0$. But the covariance will decrease from P_k^- down to P_k^+ every time a measurement is obtained due to the measurement-update equation for the covariance. Eventually P_k^- will converge to zero. This can be seen by looking at Equation (5.26), which shows the one-step equation for P_k^-:

$$P_{k+1}^- = F_k P_k^- F_k^T - F_k K_k H_k P_k^- F_k^T + Q_k \tag{5.64}$$

If $Q_k = 0$ then this equation has a steady solution of zero. A zero value for P_k^- will result in $K_k = 0$, as seen from Equation (5.63). A zero value for K_k means that the measurement-update equation (5.63) for $\hat{x}$ will not take any account of the measurement – that is, the measurement y_k will be completely ignored in the computation of $\hat{x}^+$. This is because the measurement noise covariance R_k (assuming it is greater than zero) will be infinitely times larger than the process noise $Q_k = 0$. The filter will become sluggish in the sense that it will not respond to measurements.

On the other hand, if Q_k is larger, then the covariance will always increase between time samples – that is, P_k^- will always be larger than P_{k-1}^+. When P_k^- converges, it will converge to a larger value. This will make K_k converge to a larger value. A larger K_k means that the measurement update for $\hat{x}$ in Equation (5.63) will include a larger emphasis on the measurement – that is, the filter will pay more attention to the measurements.

5.6 SUMMARY

In this chapter, we have presented the essence of the discrete-time Kalman filter. Over the past few decades, this estimation algorithm has found applications in virtually every area of engineering. We have seen that the Kalman filter equations can be written in several different ways, each of which may appear quite different than the others, although they are all mathematically equivalent. We have seen that

[4] Noise, like most things in life, is beneficial in moderate amounts. We also see this in human psychological responses to noise. Too much noise will drive humans insane, but too little noise might also result in a loss of sanity. Noise is especially beneficial for controls engineers, who would not only lose their sanity but would also lose their research funding if not for noise [Bar01, p. 179].

the Kalman filter is optimal even when the noise is not Gaussian. The Kalman filter is the optimal estimator when the noise is Gaussian, and it is the optimal *linear* estimator when the noise is not Gaussian. We have seen that the Kalman filter may not perform well if the underlying assumptions do not hold, and we briefly mentioned some ways to compensate for violated assumptions. The later chapters of this book will expand and generalize the results presented in this chapter.

PROBLEMS

Written exercises

5.1 A radioactive mass has a half-life of τ seconds. At each time step the number of emitted particles x is half of what it was one time step ago, but there is some error w_k (zero-mean with variance Q) in the number of emitted particles due to background radiation. At each time step, the number of emitted particles is counted. The instrument used to count the number of emitted particles has a random error at time k of v_k, which is zero-mean with a variance of R. Assume that w_k and v_k are uncorrelated.

a) Write the linear system equations for this system.

b) Suppose we want to use a Kalman filter to find the optimal estimate of the number of emitted particles at each time step. Write the one-step *a posteriori* Kalman filter equations for this system.

c) Find the steady-state *a posteriori* estimation-error variance for the Kalman filter.

d) What is the steady-state Kalman gain when $Q = R$? What is the steady-state Kalman gain when $Q = 2R$? Give an intuitive explanation for why the steady-state gain changes the way it does when the ratio of Q to R changes.

5.2 This problem illustrates the robustness that is achieved by the use of the Joseph form of the covariance measurement update equation. Suppose you have a discrete-time Kalman filter for a scalar system.

a) Find $\partial P_k^+/\partial K_k$ for the third form of the covariance measurement update in Equation (5.19).

b) Find $\partial P_k^+/\partial K_k$ for the Joseph form (the first form) of the covariance measurement update in Equation (5.19). After you get your answer, substitute for K_k from the Kalman gain expression.

c) Use the above results to explain why the Joseph form of the covariance measurement-update equation is stable and robust.

5.3 Prove that $E[\hat{x}_k^+(\tilde{x}_k^+)^T] = 0$. Hint: Since $\hat{x}_0^+ = E[x_0]$ is a constant and $\tilde{x}_0^+ = x_0 - \hat{x}_0^+$ is zero-mean, we know that $E[\hat{x}_0^+(\tilde{x}_0^+)^T] = 0$. Given this information, prove that $E[\hat{x}_1^+(\tilde{x}_1^+)^T] = 0$. From this point, use induction to complete the proof.

5.4 Suppose that you have a fish tank with x_p piranhas and x_g guppies [Bay99]. Once per week, you put guppy food into the tank (which the piranhas do not eat). Each week the piranhas eat some of the guppies. The birth rate of the piranhas is proportional to the guppy population, and the death rate of the piranhas is proportional to their own population (due to overcrowding). Therefore $x_p(k+1) = x_p(k) + k_1 x_g(k) - k_2 x_p(k) + w_p(k)$, where k_1 and k_2 are proportionality constants and $w_p(k)$ is white noise with a variance of one that accounts for mismodeling. The birth rate of the guppies is proportional to the food supply u, and the death rate of the guppies is proportional to the piranha population. Therefore, $x_g(k+1) = x_g(k) + u(k) - k_3 x_p(k) + w_g(k)$, where k_3 is a proportionality constant and $w_g(k)$ is white noise with a variance of one that accounts for mismodeling. The step size for this model is one week. Every week, you count the piranhas and guppies. You can count the piranhas accurately because they are so large, but your guppy count has zero-mean noise with a variance of one. Assume that $k_1 = 1$ and $k_2 = k_3 = 1/2$.

a) Generate a linear state-space model for this system.

b) Suppose that at the initial time you have a perfect count for x_p and x_g. Using a Kalman filter to estimate the guppy population, what is the variance of your guppy population estimate after one week? What is the variance after two weeks?

c) What is the ratio of the piranha population to the guppy population when they reach steady state? Assume that the process noise is zero for this part of the problem.

5.5 The measured output of a simple moving average process is $y_k = z_k + z_{k-1}$, where $\{z_j\}$ is zero-mean white noise with a variance of one.

a) Generate a state-space description for this system with the first element of x_k equal to z_{k-1} and second element equal to z_k.

b) Suppose that the initial estimation-error covariance is equal to the identity matrix. Show that the *a posteriori* estimation-error covariance is given by

$$P_k^+ = \frac{1}{k+1}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

c) Find $E\left[||x_k - \hat{x}_k^+||_2^2\right]$ as a function of k.

5.6 In this problem, we use the auxiliary variable $S_k = H_k P_k^- H_k^T + R_k$. Note that

$$\begin{bmatrix} I & 0 \\ -P_k^- H_k^T S_k^{-1} & I \end{bmatrix}\begin{bmatrix} S_k & H_k P_k^- \\ P_k^- H_k^T & P_k^- \end{bmatrix} = \begin{bmatrix} S_k & H_k P_k^- \\ 0 & P_k^+ \end{bmatrix}$$

Use the product rule for determinants to show that

$$|P_k^+| = \frac{|P_k^-||R_k|}{|S_k|}$$

5.7 In Section 4.1, we saw that Σ_k, the covariance of the state of a discrete-time system, is given as $\Sigma_{k+1} = F_k \Sigma_k F_k^T + Q_k$. Use this along with the one-step expression for the *a priori* estimation-error covariance of the Kalman filter to show that $\Sigma_k - P_k^- \geq 0$ for all k. Give an intuitive explanation for this expression [And79].

5.8 Consider the system of Problem 5.1.

a) Use the method of Section 5.4 to find a closed-form solution for P_k^-, assuming that $Q = 1$, $R = 5$, and $P_0 = 0$.

b) Use your result from above to find the steady-state value of P_k^-.

5.9 Suppose that a Kalman filter is designed for the system

$$\begin{aligned} x_{k+1} &= x_k \\ y_k &= x_k + v_k \\ v_k &\sim (0, R) \end{aligned}$$

a) Suppose that $E(x_0^2) = 1$. Design a Kalman filter for the system and find a closed-form expression for P_k^-. What is the limit of P_k^- as $k \to \infty$?

b) Now suppose that the true process equation is actually $x_{k+1} = x_k + w_k$, where $w_k \sim (0, Q)$. Find a difference equation for the variance of the *a priori* estimation error if the Kalman filter that you designed in part (a) is used to estimate the state. What is the limit of the estimation-error variance as $k \to \infty$?

5.10 Suppose that a Kalman filter is designed for a discrete LTI system with an assumed measurement noise covariance of R, but the actual measurement noise covariance is $(R + \Delta R)$. The output of the Kalman filter will indicate that the *a priori* estimation-error covariance is P_k^-, but the actual *a priori* estimation-error covariance will be Σ_k^-. Find a difference equation for $\Delta_k = (\Sigma_k^- - P_k^-)$. Will Δ_k always be positive definite?

Computer exercises

5.11 Let p_k denote the wombat population at time k, and f_k denote the size of the wombat's food supply at time k. From one time step to the next, half of the existing wombat population dies, but the number of new wombats is added to the population is equal to twice the food supply. The food supply is constant except for zero-mean random fluctuations with a variance of 10. At each time step the wombat population is counted with an error that has zero mean and a variance of 10. The initial state is

$$\begin{aligned} p_0 &= 650 \\ f_0 &= 250 \end{aligned}$$

The initial state estimate and uncertainty is

$$\begin{aligned} \hat{p}_0 &= 600 \\ E[(\hat{p}_0 - p_0)^2] &= 500 \\ \hat{f}_0 &= 200 \\ E[(\hat{f}_0 - f_0)^2] &= 200 \end{aligned}$$

Design a Kalman filter to estimate the population and food supply.

a) Simulate the system and the Kalman filter for 10 time steps. Hand in the following.

- Source code listing.
- A plot showing the true population and the estimated population as a function of time.

- A plot showing the true food supply and the estimated food supply as a function of time.

- A plot showing the standard deviation of the population and food supply estimation error as a function of time.

- A plot showing the elements of the Kalman gain matrix as a function of time.

b) Compare the standard deviation of the estimation error of your simulation with the steady-state theoretical standard deviation based on P_k^+. Why is there such a discrepancy?

c) Run the simulation again for 1000 time steps and compare the experimental estimation error standard deviation with the theoretical standard deviation.

5.12 Consider the RLC circuit described in Problem 1.18 with $R = 3$, $L = 1$, and $C = 0.5$. The input voltage is zero-mean, unity variance white noise. Suppose that the capacitor voltage is measured at 10 Hz with zero-mean, unity variance white noise. Design a Kalman filter to estimate the inductor current, with an initial covariance $P_0^+ = 0$. Generate a plot showing the *a priori* and *a posteriori* variances of the inductor current estimate for 20 time steps. Based on the plot, what is the steady-state value of P_k^-? Use the development of Section 5.4.1 to approximate the steady-state value of P_k^- using 1, 2, 3, and 4 successive squares of the Ψ matrix.