with our intuition. Note that if all of the σ_i constants are equal, this estimate reduces to the simpler form given in Equation (3.10).

$\nabla\nabla\nabla$

3.3 RECURSIVE LEAST SQUARES ESTIMATION

Equation (3.15) gives us a way to compute the optimal estimate of a constant, but there is a problem. Note that the H matrix in (3.15) is a $k \times n$ matrix. If we obtain measurements sequentially and want to update our estimate of x with each new measurement, we need to augment the H matrix and completely recompute the estimate $\hat{x}$. If the number of measurements becomes large, then the computational effort could become prohibitive. For example, suppose we obtain a measurement of a satellite's altitude once per second. After one hour has passed, the number of measurements is 3600 and growing. The computational effort of least squares estimation can rapidly outgrow our resources.

In this section, we show how to *recursively* compute the weighted least squares estimate of a constant. That is, suppose we have $\hat{x}$ after $(k-1)$ measurements, and we obtain a new measurement y_k. How can we update our estimate without completely reworking Equation (3.15)?

A linear recursive estimator can be written in the form

$$\begin{aligned} y_k &= H_k x + v_k \\ \hat{x}_k &= \hat{x}_{k-1} + K_k(y_k - H_k\hat{x}_{k-1}) \end{aligned} \tag{3.20}$$

That is, we compute $\hat{x}_k$ on the basis of the previous estimate $\hat{x}_{k-1}$ and the new measurement y_k. K_k is a matrix to be determined called the estimator gain matrix. The quantity $(y_k - H_k\hat{x}_{k-1})$ is called the correction term. Note that if the correction term is zero, or if the gain matrix is zero, then the estimate does not change from time step $(k-1)$ to k.

Before we compute the optimal gain matrix K_k, let us think about the mean of the estimation error of the linear recursive estimator. The estimation error mean can be computed as

$$\begin{aligned} E(\epsilon_{x,k}) &= E(x - \hat{x}_k) \\ &= E[x - \hat{x}_{k-1} - K_k(y_k - H_k\hat{x}_{k-1})] \\ &= E[\epsilon_{x,k-1} - K_k(H_k x + v_k - H_k\hat{x}_{k-1})] \\ &= E[\epsilon_{x,k-1} - K_k H_k(x - \hat{x}_{k-1}) - K_k v_k] \\ &= (I - K_k H_k)E(\epsilon_{x,k-1}) - K_k E(v_k) \end{aligned} \tag{3.21}$$

So if $E(v_k) = 0$ and $E(\epsilon_{x,k-1}) = 0$, then $E(\epsilon_{x,k}) = 0$. In other words, if the measurement noise v_k is zero-mean for all k, and the initial estimate of x is set equal to the expected value of x [i.e., $\hat{x}_0 = E(x)$], then the expected value of $\hat{x}_k$ will be equal to x_k for all k. Because of this, the estimator of Equation (3.20) is called an unbiased estimator. Note that this property holds regardless of the value of the gain matrix K_k. This is a desirable property of an estimator because it says that, *on average*, the estimate $\hat{x}$ will be equal to the true value x.

Next we turn our attention to the determination of the optimal value of K_k. Since the estimator is unbiased regardless of what value of K_k we use, we must choose some other optimality criterion in order to determine K_k. The optimality criterion that we choose to minimize is the sum of the variances of the estimation errors at time k:

$$\begin{aligned} J_k &= E[(x_1 - \hat{x}_1)^2)] + \cdots + E[(x_n - \hat{x}_n)^2)] \\ &= E\left(\epsilon_{x1,k}^2 + \cdots + \epsilon_{xn,k}^2\right) \\ &= E\left(\epsilon_{x,k}^T \epsilon_{x,k}\right) \\ &= E\left[\mathrm{Tr}(\epsilon_{x,k}\epsilon_{x,k}^T)\right] \\ &= \mathrm{Tr} P_k \end{aligned} \tag{3.22}$$

where P_k, the estimation-error covariance, is defined by the above equation. We can use a process similar to that followed in Equation (3.21) to obtain a recursive formula for the calculation of P_k:

$$\begin{aligned} P_k &= E(\epsilon_{x,k}\epsilon_{x,k}^T) \\ &= E\left\{[(I - K_k H_k)\epsilon_{x,k-1} - K_k v_k][\cdots]^T\right\} \\ &= (I - K_k H_k) E(\epsilon_{x,k-1}\epsilon_{x,k-1}^T)(I - K_k H_k)^T - \\ &\quad K_k E(v_k \epsilon_{x,k-1}^T)(I - K_k H_k)^T - (I - K_k H_k) E(\epsilon_{x,k-1} v_k^T) K_k^T + \\ &\quad K_k E(v_k v_k^T) K_k^T \end{aligned} \tag{3.23}$$

Now note that $\epsilon_{x,k-1}$ [the estimation error at time $(k-1)$] is independent of v_k (the measurement noise at time k). Therefore,

$$\begin{aligned} E(v_k \epsilon_{x,k-1}^T) &= E(v_k) E(\epsilon_{x,k-1}) \\ &= 0 \end{aligned} \tag{3.24}$$

since both expected values are zero. Therefore, Equation (3.23) becomes

$$P_k = (I - K_k H_k) P_{k-1} (I - K_k H_k)^T + K_k R_k K_k^T \tag{3.25}$$

where R_k is the covariance of v_k. This is the recursive formula for the covariance of the least squares estimation error. This is consistent with intuition in the sense that as the measurement noise increases (i.e., R_k increases) the uncertainty in our estimate also increases (i.e., P_k increases). Note that P_k should be positive definite since it is a covariance matrix, and the form of Equation (3.25) guarantees that P_k will be positive definite, assuming that P_{k-1} and R_k are positive definite.

Now we need to find the value of K_k that makes the cost function in Equation (3.22) as small as possible. The mean of the estimation error is zero for any value of K_k. So if we choose K_k to make the cost function (i.e., the trace of P_k) small then the estimation error will not only be zero-mean, but it will also be consistently close to zero. In order to find the best value of K_k, first we need to recall from Equation (1.66) that $\frac{\partial \mathrm{Tr}(ABA^T)}{\partial A} = 2AB$ if B is symmetric. With this in mind we can use Equations (3.22), (3.25), and the chain rule to obtain

$$\frac{\partial J_k}{\partial K_k} = 2(I - K_k H_k) P_{k-1}(-H_k^T) + 2K_k R_k \tag{3.26}$$

In order to find the value of K_k that minimizes J_k, we set the above derivative equal to zero and then solve for K_k as follows:

$$\begin{aligned} K_k R_k &= (I - K_k H_k) P_{k-1} H_k^T \\ K_k(R_k + H_k P_{k-1} H_k^T) &= P_{k-1} H_k^T \\ K_k &= P_{k-1} H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1} \end{aligned} \tag{3.27}$$

Equations (3.20), (3.25), and (3.27) form the recursive least squares estimator. The recursive least squares estimator can be summarized as follows.

Recursive least squares estimation

1. Initialize the estimator as follows:

$$\begin{aligned} \hat{x}_0 &= E(x) \\ P_0 &= E[(x - \hat{x}_0)(x - \hat{x}_0)^T] \end{aligned} \tag{3.28}$$

   If no knowledge about x is available before measurements are taken, then $P_0 = \infty I$. If perfect knowledge about x is available before measurements are taken, then $P_0 = 0$.

2. For $k = 1, 2, \cdots$, perform the following.

   (a) Obtain the measurement y_k, assuming that y_k is given by the equation

$$y_k = H_k x + v_k \tag{3.29}$$

   where v_k is a zero-mean random vector with covariance R_k. Further assume that the measurement noise at each time step k is independent, that is, $E(v_i v_k) = R_k \delta_{k-i}$. This implies that the measurement noise is white.

   (b) Update the estimate of x and the estimation-error covariance P as follows:

$$\begin{aligned} K_k &= P_{k-1} H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1} \\ \hat{x}_k &= \hat{x}_{k-1} + K_k (y_k - H_k \hat{x}_{k-1}) \\ P_k &= (I - K_k H_k) P_{k-1} (I - K_k H_k)^T + K_k R_k K_k^T \end{aligned} \tag{3.30}$$

3.3.1 Alternate estimator forms

Sometimes it is useful to write the equations for P_k and K_k in alternate forms. Although these alternate forms are mathematically identical, they can be beneficial from a computational point of view. They can also lead to new results, which we will discover in later chapters.

First we will find an alternate form for the expression for the estimation-error covariance. Substituting for K_k from Equation (3.27) into Equation (3.25) we obtain

$$P_k = \left[I - P_{k-1} H_k^T S_k^{-1} H_k\right] P_{k-1} [\cdots]^T + K_k R_k K_k^T \tag{3.31}$$

where we have introduced the auxiliary variable $S_k = (H_k P_{k-1} H_k^T + R_k)$. We again substitute for K_k at the end of this equation, and expand terms to obtain

$$\begin{aligned} P_k &= P_{k-1} - P_{k-1}H_k^T S_k^{-1} H_k P_{k-1} - P_{k-1}H_k^T S_k^{-1} H_k P_{k-1} + \\ &\quad P_{k-1}H_k^T S_k^{-1} H_k P_{k-1} H_k^T S_k^{-1} H_k P_{k-1} + P_{k-1}H_k^T S_k^{-1} R_k S_k^{-1} H_k P_{k-1} \end{aligned} \tag{3.32}$$

Combining the last two terms in this equation gives

$$\begin{aligned} P_k &= P_{k-1} - 2P_{k-1}H_k^T S_k^{-1} H_k P_{k-1} + P_{k-1}H_k^T S_k^{-1} S_k S_k^{-1} H_k P_{k-1} \\ &= P_{k-1} - 2P_{k-1}H_k^T S_k^{-1} H_k P_{k-1} + P_{k-1}H_k^T S_k^{-1} H_k P_{k-1} \\ &= P_{k-1} - P_{k-1}H_k^T S_k^{-1} H_k P_{k-1} \end{aligned} \tag{3.33}$$

Now notice from the expression for K_k in Equation (3.27) that K_k appears implicitly in the above equation. We can therefore rewrite this equation as

$$\begin{aligned} P_k &= P_{k-1} - K_k H_k P_{k-1} \\ &= (I - K_k H_k) P_{k-1} \end{aligned} \tag{3.34}$$

This is a simpler equation for P_k [compared with Equation (3.25)] but numerical computing problems (i.e., scaling issues) may cause this expression for P_k to not be positive definite, even when P_{k-1} and R_k are positive definite.

We can also use the matrix inversion lemma from Section 1.1.2 to rewrite the measurement update equation for P_k. Starting with Equation (3.33) we obtain

$$P_k = P_{k-1} - P_{k-1}H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1} H_k P_{k-1} \tag{3.35}$$

Taking the inverse of both sides of this equation gives

$$P_k^{-1} = [P_{k-1} - P_{k-1}H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1} H_k P_{k-1}]^{-1} \tag{3.36}$$

Applying the matrix inversion lemma to this equation gives

$$\begin{aligned} P_k^{-1} &= P_{k-1}^{-1} + P_{k-1}^{-1} P_{k-1} H_k^T \left[(H_k P_{k-1} H_k^T + R_k) - \right. \\ &\quad \left. H_k P_{k-1} P_{k-1}^{-1} (P_{k-1} H_k^T) \right]^{-1} H_k P_{k-1} P_{k-1}^{-1} \\ &= P_{k-1}^{-1} + H_k^T R_k^{-1} H_k \end{aligned} \tag{3.37}$$

Inverting both sides of this equation gives

$$P_k = \left[P_{k-1}^{-1} + H_k^T R_k^{-1} H_k \right]^{-1} \tag{3.38}$$

This equation for P_k is more complicated in that it requires three matrix inversions, but it may be computationally advantageous in some situations, as will be discussed in Section 6.2.

We can use Equation (3.38) to derive an equivalent equation for the estimator gain K_k. Starting with Equation (3.27) we have

$$K_k = P_{k-1} H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1} \tag{3.39}$$

Premultiplying the right side by $P_k P_k^{-1}$, which is equal to the identity matrix, gives

$$K_k = P_k P_k^{-1} P_{k-1} H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1} \tag{3.40}$$

Substituting for P_k^{-1} from Equation (3.38) gives

$$K_k = P_k(P_{k-1}^{-1} + H_k^T R_k^{-1} H_k) P_{k-1} H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1} \tag{3.41}$$

Note the $P_{k-1}H_k^T$ factor that is on the right of the first term in parentheses. We can multiply this factor inside the first term in parentheses to obtain

$$K_k = P_k(H_k^T + H_k^T R_k^{-1} H_k P_{k-1} H_k^T)(H_k P_{k-1} H_k^T + R_k)^{-1} \tag{3.42}$$

Now bring H_k^T out to the left side of the parentheses to obtain

$$K_k = P_k H_k^T (I + R_k^{-1} H_k P_{k-1} H_k^T)(H_k P_{k-1} H_k^T + R_k)^{-1} \tag{3.43}$$

Now premultiply the first parenthetical expression by R_k^{-1}, and multiply on the inside of the parenthetical expression by R_k, to obtain

$$\begin{aligned} K_k &= P_k H_k^T R_k^{-1} (R_k + H_k P_{k-1} H_k^T)(H_k P_{k-1} H_k^T + R_k)^{-1} \\ &= P_k H_k^T R_k^{-1} \end{aligned} \tag{3.44}$$

General recursive least squares estimation

The recursive least squares algorithm can be summarized with the following equations. The measurement equations are given as

$$\begin{aligned} y_k &= H_k x + v_k \\ x &= \text{constant} \\ E(v_k) &= 0 \\ E(v_k v_i^T) &= R_k \delta_{k-i} \end{aligned} \tag{3.45}$$

The initial estimate of the constant vector x, along with the uncertainty in that estimate, is given as

$$\begin{aligned} \hat{x}_0 &= E(x) \\ P_0 &= E[(x - \hat{x}_0)(x - \hat{x}_0)^T] \end{aligned} \tag{3.46}$$

The recursive least squares algorithm is given as follows.
For $k = 1, 2, \cdots$,

$$\begin{aligned} K_k &= P_{k-1} H_k^T (H_k P_{k-1} H_k^T + R_k)^{-1} \\ &= P_k H_k^T R_k^{-1} \\ \hat{x}_k &= \hat{x}_{k-1} + K_k(y_k - H_k \hat{x}_{k-1}) \\ P_k &= (I - K_k H_k) P_{k-1} (I - K_k H_k)^T + K_k R_k K_k^T \\ &= (P_{k-1}^{-1} + H_k^T R_k^{-1} H_k)^{-1} \\ &= (I - K_k H_k) P_{k-1} \end{aligned} \tag{3.47}$$

■ EXAMPLE 3.3

Once again we revisit the problem of trying to estimate the resistance x of an unmarked resistor on the basis of noisy measurements from a multimeter. However, we do not want to wait until we have all the measurements in order to have an estimate. We want to recursively modify our estimate of x each time we obtain a new measurement. At sample time k our measurement is

$$\begin{aligned} y_k &= H_k x + v_k \\ H_k &= 1 \\ R_k &= E(v_k^2) \end{aligned} \tag{3.48}$$

For this scalar problem, the measurement matrix H_k is a scalar, and the measurement noise covariance R_k is also a scalar. We will suppose that each measurement has the same covariance so the measurement covariance R_k is not a function of k, and can be written as R. Initially, before we have any measurements, we have some idea about the value of the resistance x, and this forms our initial estimate. We also have some uncertainty about our initial estimate, and this forms our initial covariance:

$$\begin{aligned} \hat{x}_0 &= E(x) \\ P_0 &= E[(x-\hat{x}_0)(x-\hat{x}_0)^T] \\ &= E[(x-\hat{x}_0)^2] \end{aligned} \tag{3.49}$$

If we have absolutely no idea about the resistance value, then $P(0) = \infty$. If we are 100% certain about the resistance value before taking any measurements, then $P(0) = 0$ (but then, of course, there would not be any need to take measurements). Equation (3.47) tells us how to obtain the estimator gain, the estimate of x, and the estimation covariance, after the first measurement ($k = 1$):

$$\begin{aligned} K_k &= P_{k-1}H_k^T(H_kP_{k-1}H_k^T + R_k)^{-1} \\ K_1 &= P_0(P_0+R)^{-1} \\ \hat{x}_k &= \hat{x}_{k-1} + K_k(y_k - H_k\hat{x}_{k-1}) \\ \hat{x}_1 &= \hat{x}_0 + \frac{P_0}{P_0+R}(y_1 - \hat{x}_0) \\ P_k &= (I - K_kH_k)P_{k-1}(I - K_kH_k)^T + K_kR_kK_k^T \\ P_1 &= \frac{P_0R}{P_0+R} \end{aligned} \tag{3.50}$$

Repeating these calculations to find these quantities after the second measurement ($k = 2$) gives

$$\begin{aligned} K_2 &= \frac{P_1}{P_1+R} = \frac{P_0}{2P_0+R} \\ P_2 &= \frac{P_1R}{P_1+R} = \frac{P_0R}{2P_0+R} \\ \hat{x}_2 &= \hat{x}_1 + \frac{P_1}{P_1+R}(y_2 - \hat{x}_1) \\ &= \frac{P_0+R}{2P_0+R}\hat{x}_1 + \frac{P_0}{2P_0+R}y_2 \end{aligned} \tag{3.51}$$

By induction, we can find general expressions for P_{k-1}, K_k, and $\hat{x}_k$ as follows:

$$\begin{aligned}
P_{k-1} &= \frac{P_0 R}{(k-1)P_0 + R} \\
K_k &= \frac{P_0}{kP_0 + R} \\
\hat{x}_k &= \hat{x}_{k-1} + K_k(y_k - \hat{x}_{k-1}) \\
&= (1 - K_k)\hat{x}_{k-1} + K_k y_k \\
&= \frac{(k-1)P_0 + R}{kP_0 + R}\hat{x}_{k-1} + \frac{P_0}{kP_0 + R} y_k
\end{aligned} \tag{3.52}$$

Note that if x is known perfectly *a priori* (i.e., before any measurements are obtained) then $P_0 = 0$, and the above equations show that $K_k = 0$ and $\hat{x}_k = \hat{x}_0$. That is, the optimal estimate of x is independent of any measurements that are obtained. On the other hand, if x is completely unknown *a priori*, then $P_0 \to \infty$, and the above equations show that

$$\begin{aligned}
\hat{x}_k &= \frac{(k-1)P_0}{kP_0}\hat{x}_{k-1} + \frac{P_0}{kP_0} y_k \\
&= \frac{(k-1)}{k}\hat{x}_{k-1} + \frac{1}{k} y_k \\
&= \frac{1}{k}[(k-1)\hat{x}_{k-1} + y_k]
\end{aligned} \tag{3.53}$$

In other words, the optimal estimate of x is equal to the running average of the measurements y_k, which can be written as

$$\begin{aligned}
\bar{y}_k &= \frac{1}{k}\sum_{j=1}^{k} y_j \\
&= \frac{1}{k}\left(\sum_{j=1}^{k-1} y_j + y_k\right) \\
&= \frac{1}{k}\left[(k-1)\left(\frac{1}{k-1}\sum_{j=1}^{k-1} y_j\right) + y_k\right] \\
&= \frac{1}{k}\left[(k-1)\bar{y}_{k-1} + y_k\right]
\end{aligned} \tag{3.54}$$

∇∇∇

■ EXAMPLE 3.4

In this example, we illustrate the computational advantages of the first form of the covariance update in Equation (3.47) compared with the third form. Suppose we have a scalar parameter x and a perfect measurement of it. That is, $H_1 = 1$ and $R_1 = 0$. Further suppose that our initial estimation covariance $P_0 = 6$, and our computer provides precision of three digits to the right of the decimal point for each quantity that it computes. The estimator gain K_1 is computed as

$$\begin{aligned} K_1 &= P_0(P_0 + R_1)^{-1} \\ &= (6)\left(\frac{1}{6}\right) \\ &= (6)(0.167) \\ &= 1.002 \end{aligned} \tag{3.55}$$

If we use the third form of the covariance update in Equation (3.47) we obtain

$$\begin{aligned} P_1 &= (1 - K_1)P_0 \\ &= (-0.002)(6) \\ &= -0.012 \end{aligned} \tag{3.56}$$

The covariance after the first measurement is negative, which is physically impossible. However, if we use the first form of the covariance update in Equation (3.47) we obtain

$$\begin{aligned} P_1 &= (1 - K_1)P_0(1 - K_1) + K_1 R_1 K_1 \\ &= (1 - K_1)^2 P_0 + K_1^2 R_1 \\ &= 0 \end{aligned} \tag{3.57}$$

The reason we get zero is because $(1 - K_1)^2 = 0.000004$, but our computer retains only three digits to the right of the decimal point. Zero is the theoretically correct value of P_1. The form of the above expression for P_1 guarantees that it will never be negative, regardless of any numerical errors in P_0, R_1, and K_1.

∇∇∇

■ EXAMPLE 3.5

Suppose that a tank contains a concentration x_1 of chemical 1, and a concentration x_2 of chemical 2. You have some instrumentation that can detect the combined concentration $(x_1 + x_2)$ of the two chemicals, but your instrumentation cannot distinguish between the two chemicals. Chemical 2 is removed from the tank through a leaching process so that its concentration decreases by 1% from one measurement time to the next. The measurement equation is therefore given as

$$\begin{aligned} y_k &= x_1 + 0.99^{k-1} x_2 + v_k \\ &= \begin{bmatrix} 1 & 0.99^{k-1} \end{bmatrix} x + v_k \end{aligned} \tag{3.58}$$

where v_k is the measurement noise, which is a zero-mean random variable with a variance of $R = 0.01$. Suppose that $x_1 = 10$ and $x_2 = 5$. Further suppose that your initial estimates are $\hat{x}_1 = 8$ and $\hat{x}_2 = 7$, with an initial estimation-error variance P_0 that is equal to the identity matrix. A recursive least squares algorithm can be implemented as shown in Equation (3.47) to estimate the two concentrations. Figure 3.1 shows the estimate of x_1 and x_2 as measurements are obtained, along with the variance of the estimation errors. It can be seen that after a couple dozen measurements the estimates become quite close to their true values of 10 and 5. The variances of the estimation errors asymptotically approach zero, which means that we have increasingly more confidence in our estimates as we obtain more measurements.

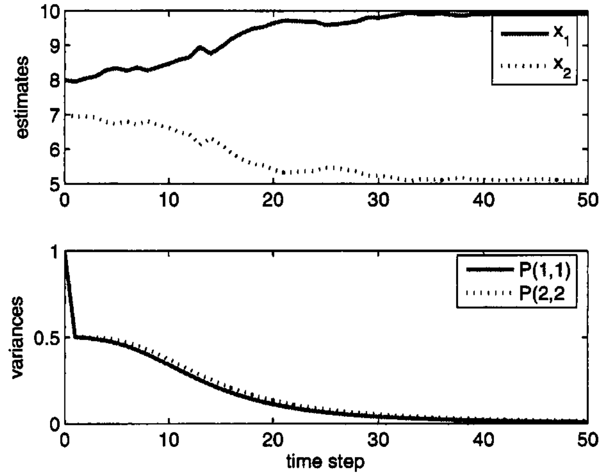

Figure 3.1 Parameter estimates and estimation variances for Example 3.5.

∇∇∇

3.3.2 Curve fitting

In this section, we will apply recursive least squares theory to the curve fitting problem. In the recursive curve fitting problem, we measure data one sample at a time (y_1, y_2, $\cdots$) and want to find the best fit of a curve to the data. The curve that we want to fit to the data could be constrained to be linear, or quadratic, or sinusoid, or some other shape, depending on the underlying problem.

■ **EXAMPLE 3.6**

Suppose that we want to fit a straight line to a set of data points. The linear data fitting problem can be written as

$$\begin{aligned} y_k &= x_1 + x_2 t_k + v_k \\ E(v_k^2) &= R_k \end{aligned} \tag{3.59}$$

t_k is the independent variable (perhaps time), y_k is the noisy data, and we want to find the linear relationship between y_k and t_k. In other words, we want to estimate the constants x_1 and x_2. The measurement matrix can be written as

$$H_k = \begin{bmatrix} 1 & t_k \end{bmatrix} \tag{3.60}$$

so that Equation (3.59) can be written as

$$y_k = H_k x + v_k \tag{3.61}$$

Our recursive estimator is initialized as

$$\begin{aligned}
\hat{x}_0 &= E(x) \\
\begin{bmatrix} \hat{x}_{1,0} \\ \hat{x}_{2,0} \end{bmatrix} &= \begin{bmatrix} E(x_1) \\ E(x_2) \end{bmatrix} \\
P_0 &= E[(x-\hat{x}_0)(x-\hat{x}_0)^T] \\
&= \begin{bmatrix} E[x_1-\hat{x}_{1,0})^2] & E[(x_1-\hat{x}_{1,0})(x_2-\hat{x}_{2,0})] \\ E[(x_1-\hat{x}_{1,0})(x_2-\hat{x}_{2,0})] & E[x_2-\hat{x}_{2,0})^2] \end{bmatrix}
\end{aligned} \tag{3.62}$$

The recursive estimate of the two-element vector x is then obtained from Equation (3.47) as follows:

For $k = 1, 2, \cdots$,

$$\begin{aligned}
K_k &= P_{k-1}H_k^T(H_kP_{k-1}H_k^T + R_k)^{-1} \\
\hat{x}_k &= \hat{x}_{k-1} + K_k(y_k - H_k\hat{x}_{k-1}) \\
P_k &= (I - K_kH_k)P_{k-1}(I - K_kH_k)^T + K_kR_kK_k^T
\end{aligned} \tag{3.63}$$

$\nabla\nabla\nabla$

■ EXAMPLE 3.7

Suppose that we know *a priori* that the underlying data is a quadratic function of time. In this case, we have a quadratic data fitting problem. For example, suppose we are measuring the altitude of a free-falling object. We know from our understanding of physics that altitude r is a function of the acceleration due to gravity, the initial altitude and velocity of the object r_0 and v_0, and time t, as given by the equation $r = r_0 + v_0t + (a/2)t^2$. So if we measure r at various time instants and fit a quadratic to the resulting r versus t curve, then we have an estimate of the parameters r_0, v_0, and $a/2$. In general, the quadratic data fitting problem can be written as

$$\begin{aligned}
y_k &= x_1 + x_2t_k + x_3t_k^2 + v_k \\
E(v_k^2) &= R_k
\end{aligned} \tag{3.64}$$

t_k is the independent variable, y_k is the noisy measurement, and we want to find the quadratic relationship between y_k and t_k. In other words, we want to estimate the constants x_1, x_2, and x_3. The measurement matrix can be written as

$$H_k = \begin{bmatrix} 1 & t_k & t_k^2 \end{bmatrix} \tag{3.65}$$

so that Equation (3.64) can be written as

$$y_k = H_k x + v_k \tag{3.66}$$

Our recursive estimator is initialized as

$$\begin{aligned}
\hat{x}_0 &= E(x) \\
P_0 &= E[(x-\hat{x}_0)(x-\hat{x}_0)^T]
\end{aligned} \tag{3.67}$$

where P_0 is a 3×3 matrix. The recursive estimate of the three-element vector x is then obtained from Equation (3.47) as follows:

For $k = 1, 2, \cdots$,

$$\begin{aligned} K_k &= P_{k-1}H_k^T(H_kP_{k-1}H_k^T + R_k)^{-1} \\ \hat{x}_k &= \hat{x}_{k-1} + K_k(y_k - H_k\hat{x}_{k-1}) \\ P_k &= (I - K_kH_k)P_{k-1}(I - K_kH_k)^T + K_kR_kK_k^T \end{aligned} \qquad (3.68)$$

$\nabla\nabla\nabla$

3.4 WIENER FILTERING

In this section, we will give a brief review of Wiener filtering. The rest of this book does not assume any knowledge on the reader's part of Wiener filtering. However, Wiener filtering is important from a historical perspective, and it still has a lot of applications in signal processing and communication theory. But since it is not used much for state estimation anymore, the reader can safely skip this section if desired.

Wiener filtering addresses the problem of designing a linear, time-invariant filter to extract a signal from noise, approaching the problem from the frequency domain perspective. Norbert Wiener invented his filter as part of the World War II effort for the United States. He published his work on the problem in 1942, but it was not available to the public until 1949 [Wie64]. His book was known as the "yellow peril" because of its mathematical difficulty and its yellow cover [Deu65, page 176]. Andrey Kolmogorov actually solved a more general problem earlier (1941), and Mark Krein also worked on the same problem (1945). Kolmogorov's and Krein's work was independent of Wiener's work, and Wiener acknowledges that Kolmogorov's work predated his own work [Wie56]. However, Kolmogorov's and Krein's work did not become well known in the Western world until later, since it was published in Russian [Kol41]. A nontechnical account of Wiener's work is given in his autobiography [Wie56].

To set up the presentation of the Wiener filter, we first need to ask the following question: How does the power spectrum of a stochastic process $x(t)$ change when it goes through an LTI system with impulse response $g(t)$? The output $y(t)$ of the system is given by the convolution of the impulse response with the input:

$$y(t) = g(t) * x(t) \qquad (3.69)$$

Since the system is time-invariant, a time shift in the input results in an equal time shift in the output:

$$y(t + \alpha) = g(t) * x(t + \alpha) \qquad (3.70)$$

Multiplying the above two equations and writing out the convolutions as integrals gives

$$y(t)y(t + \alpha) = \int g(\tau)x(t - \tau)\, d\tau \int g(\gamma)x(t + \alpha - \gamma)\, d\gamma \qquad (3.71)$$

Taking the expected value of both sides of the above equation gives the autocorrelation of $y(t)$ as a function of the autocorrelation of $x(t)$:

$$E[y(t)y(t + \alpha)] = \int\int g(\tau)g(\gamma)E[x(t - \tau)x(t + \alpha - \gamma)]\, d\tau\, d\gamma \qquad (3.72)$$