PART III

THE H_∞ FILTER

Optimal State Estimation, First Edition. By Dan J. Simon
ISBN 0471708585

CHAPTER 11

The H_∞ filter

[Kalman filtering] assumes that the message generating process has a known dynamics and that the exogenous inputs have known statistical properties. Unfortunately, these assumptions limit the utility of minimum variance estimators in situations where the message model and/or the noise descriptions are unknown.

—Uri Shaked and Yahali Theodor [Sha92]

As we have seen in earlier chapters, the Kalman filter is an effective tool for estimating the states of a system. The early success in the 1960s of the Kalman filter in aerospace applications led to attempts to apply it to more common industrial applications in the 1970s. However, these attempts quickly made it clear that a serious mismatch existed between the underlying assumptions of Kalman filters and industrial state estimation problems. Accurate system models are not as readily available for industrial problems. The government spent millions of dollars on the space program in the 1960s (hence the accurate system models), but industry rarely has millions of dollars to spend on engineering problems (hence the inaccurate system models). In addition, engineers rarely understand the statistical nature of the noise processes that impinge on industrial processes. After a decade or so of reappraising the nature and role of Kalman filters, engineers realized they needed a new filter that could handle modeling errors and noise uncertainty. State estimators that can tolerate such uncertainty are called robust. Although robust

Optimal State Estimation, First Edition. By Dan J. Simon
ISBN 0471708585

estimators based on Kalman filter theory can be designed (as seen in Section 10.4), these approaches are somewhat ad-hoc in that they attempt to modify an already existing approach. The H_∞ filter was specifically designed for robustness.

In Section 11.1 we derive a different form of the Kalman filter and discuss the limitations of the Kalman filter. Section 11.2 discusses constrained optimization using Lagrange multipliers, which we will need later for our derivation of the H_∞ filter. In Section 11.3 we use a game theory approach to derive the discrete-time H_∞ filter, which minimizes the worst-case estimation error. This is in contrast to the Kalman filter's minimization of the expected value of the variance of the estimation error. Furthermore, the H_∞ filter does not make any assumptions about the statistics of the process and measurement noise (although this information can be used in the H_∞ filter if it is available). Section 11.4 presents the continuous-time H_∞ filter, and Section 11.5 discusses an alternative method for deriving the H_∞ filter using a transfer function approach.

11.1 INTRODUCTION

In this section we will first derive an alternate form for the Kalman filter. We do this to facilitate comparisons that we will make later in this chapter between the Kalman and H_∞ filters. After we derive an alternate Kalman filter form, we will briefly discuss the limitations of the Kalman filter.

11.1.1 An alternate form for the Kalman filter

Recall that the Kalman filter estimates the state of a linear dynamic system defined by the equations

$$\begin{aligned} x_{k+1} &= F_k x_k + w_k \\ y_k &= H_k x + v_k \end{aligned} \tag{11.1}$$

where $\{w_k\}$ and $\{v_k\}$ are stochastic processes with covariances Q_k and R_k, respectively. As derived in Section 5.1, the Kalman filter equations are given as follows:

$$\begin{aligned} \hat{x}_{k+1}^- &= F_k \hat{x}_k^- + F_k K_k (y_k - H_k \hat{x}_k^-) \\ K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \\ P_k^- &= F_{k-1} P_{k-1}^+ F_{k-1}^T + Q_{k-1} \\ P_k^+ &= (I - K_k H_k) P_k^- \end{aligned} \tag{11.2}$$

Using the matrix inversion lemma from Section 1.1.2 we see that

$$\begin{aligned} (H_k P_k^- H_k^T + R_k)^{-1} &= R_k^{-1} - R_k^{-1} H_k (\mathcal{I}_k^- + H_k^T R_k^{-1} H_k)^{-1} H_k^T R_k^{-1} \\ &= R_k^{-1} - R_k^{-1} H_k (I + P_k^- H_k^T R_k^{-1} H_k)^{-1} P_k^- H_k^T R_k^{-1} \end{aligned} \tag{11.3}$$

where $\mathcal{I}_k$ is the information matrix (i.e., the inverse of the covariance matrix P_k). The Kalman gain can therefore be written as follows:

$$
\begin{aligned}
K_k &= P_k^- H_k^T (H_k P_k^- H_k^T + R_k)^{-1} \\
&= P_k^- H_k^T R_k^{-1} - P_k^- H_k^T R_k^{-1} H_k (I + P_k^- H_k^T R_k^{-1} H_k)^{-1} P_k^- H_k^T R_k^{-1} \\
&= \left[I - P_k^- H_k^T R_k^{-1} H_k (I + P_k^- H_k^T R_k^{-1} H_k)^{-1}\right] P_k^- H_k^T R_k^{-1} \\
&= \left[(I + P_k^- H_k^T R_k^{-1} H_k) - P_k^- H_k^T R_k^{-1} H_k\right] (I + P_k^- H_k^T R_k^{-1} H_k)^{-1} P_k^- H_k^T R_k^{-1} \\
&= (I + P_k^- H_k^T R_k^{-1} H_k)^{-1} P_k^- H_k^T R_k^{-1} \qquad (11.4)
\end{aligned}
$$

Substituting this into the expression for P_{k+1}^- in Equation (11.2) we get

$$
\begin{aligned}
P_{k+1}^- &= F_k P_k^+ F_k^T + Q_k \\
&= F_k (I - K_k H_k) P_k^- F_k^T + Q_k \\
&= F_k P_k^- F_k^T - F_k K_k H_k P_k^- F_k^T + Q_k \\
&= F_k P_k^- F_k^T - F_k (I + P_k^- H_k^T R_k^{-1} H_k)^{-1} P_k^- H_k^T R_k^{-1} H_k P_k^- F_k^T + Q_k \\
&= F_k P_k^- F_k^T - F_k (\mathcal{I}_k^- + H_k^T R_k^{-1} H_k)^{-1} H_k^T R_k^{-1} H_k P_k^- F_k^T + Q_k \qquad (11.5)
\end{aligned}
$$

Apply the matrix inversion lemma again to the inverse on the right side of the above equation to obtain

$$
\begin{aligned}
P_{k+1}^- &= F_k P_k^- F_k^T - \\
&\quad F_k \left[P_k^- - P_k^- H_k^T (R_k + H_k P_k^- H_k^T)^{-1} H_k P_k^-\right] H_k^T R_k^{-1} H_k P_k^- F_k^T + Q_k \\
&= F_k P_k^- \left[I - H_k^T R_k^{-1} H_k P_k^- + \right. \\
&\quad \left. H_k^T (R_k + H_k P_k^- H_k^T)^{-1} H_k P_k^- H_k^T R_k^{-1} H_k P_k\right] F_k^T + Q_k \\
&= F_k P_k^- T_k F_k^T + Q_k \qquad (11.6)
\end{aligned}
$$

where T_k is defined by the above equation. Apply the matrix inversion lemma to the inverse that is in T_k to obtain

$$
\begin{aligned}
T_k &= I - H_k^T R_k^{-1} H_k P_k^- + \\
&\quad H_k^T \left[R_k^{-1} - R_k^{-1} H_k (\mathcal{I}_k^- + H_k^T R_k^{-1} H_k)^{-1} H_k^T R_k^{-1}\right] H_k P_k^- H_k^T R_k^{-1} H_k P_k \\
&= I - H_k^T R_k^{-1} H_k P_k^- + (H_k^T R_k^{-1} H_k P_k^-)^2 - \\
&\quad H_k^T R_k^{-1} H_k (\mathcal{I}_k^- + H_k^T R_k^{-1} H_k)^{-1} (H_k^T R_k^{-1} H_k P_k^-)^2 \\
&= I - H_k^T R_k^{-1} H_k P_k^- + (H_k^T R_k^{-1} H_k P_k^-)^2 - \\
&\quad H_k^T R_k^{-1} H_k P_k^- (I + H_k^T R_k^{-1} H_k P_k^-)^{-1} (H_k^T R_k^{-1} H_k P_k^-)^2 \\
&= I - H_k^T R_k^{-1} H_k P_k^- + (H_k^T R_k^{-1} H_k P_k^-)^2 - \\
&\quad (H_k^T R_k^{-1} H_k P_k^-)^3 (I + H_k^T R_k^{-1} H_k P_k^-)^{-1} \\
&= \left[(I + H_k^T R_k^{-1} H_k P_k^-) - H_k^T R_k^{-1} H_k P_k^- (I + H_k^T R_k^{-1} H_k P_k^-) + \right. \\
&\quad \left. (H_k^T R_k^{-1} H_k P_k^-)^2 (I + H_k^T R_k^{-1} H_k P_k^-) - (H_k^T R_k^{-1} H_k P_k^-)^3\right] \times \\
&\quad (I + H_k^T R_k^{-1} H_k P_k^-)^{-1} \\
&= (I + H_k^T R_k^{-1} H_k P_k^-)^{-1} \qquad (11.7)
\end{aligned}
$$

Substituting this expression for T_k into Equation (11.6) gives

$$
P_{k+1}^- = F_k P_k^- (I + H_k^T R_k^{-1} H_k P_k^-)^{-1} F_k^T + Q_k \qquad (11.8)
$$

From Equation (11.4) the Kalman gain can be written as

$$K_k = (I + P_k^- H_k^T R_k^{-1} H_k)^{-1} P_k^- H_k^T R_k^{-1} \tag{11.9}$$

We can premultiply outside the parentheses by P_k^-, and postmultiply each term inside the parenthesis by P_k^-, to obtain

$$K_k = P_k^- (P_k^- + P_k^- H_k^T R_k^{-1} H_k P_k^-)^{-1} P_k^- H_k^T R_k^{-1} \tag{11.10}$$

We can postmultiply outside the parentheses by the inverse of P_k^-, and premultiply each term inside the parentheses by the inverse of P_k^-, to obtain

$$K_k = P_k^- (I + H_k^T R_k^{-1} H_k P_k^-)^{-1} H_k^T R_k^{-1} \tag{11.11}$$

Combining this expression for K_k with Equations (11.2) and (11.8) we can summarize the Kalman filter as follows:

$$\begin{aligned} \hat{x}_{k+1}^- &= F_k \hat{x}_k^- + F_k K_k (y_k - H_k \hat{x}_k^-) \\ K_k &= P_k^- (I + H_k^T R_k^{-1} H_k P_k^-)^{-1} H_k^T R_k^{-1} \\ P_{k+1}^- &= F_k P_k^- (I + H_k^T R_k^{-1} H_k P_k^-)^{-1} F_k^T + Q_k \end{aligned} \tag{11.12}$$

11.1.2 Kalman filter limitations

The Kalman filter works well, but only under certain conditions.

- First, we need to know the mean and correlation of the noise w_k and v_k at each time instant.
- Second, we need to know the covariances Q_k and R_k of the noise processes. The Kalman filter uses Q_k and R_k as design parameters, so if we do not know Q_k and R_k then it may be difficult to successfully use a Kalman filter.
- Third, the attractiveness of the Kalman filter lies in the fact that it is the one estimator that results in the smallest possible standard deviation of the estimation error. That is, the Kalman filter is the minimum variance estimator if the noise is Gaussian, and it is the linear minimum variance estimator if the noise is not Gaussian. If we desire to minimize a different cost function (such as the worst-case estimation error) then the Kalman filter may not accomplish our objectives.
- Finally, we need to know the system model matrices F_k and H_k.

So what do we do if one of the Kalman filter assumptions is not satisfied? What should we do if we do not have any information about the noise statistics? What should we do if we want to minimize the worst-case estimation error rather than the covariance of the estimation error?

Perhaps we could just use the Kalman filter anyway, even though its assumptions are not satisfied, and hope for the best. That is a common solution to our Kalman filter quandary and it works reasonably well in many cases. However, there is yet another option that we will explore in this chapter: the H_∞ filter, also called the minimax filter. The H_∞ filter does not make any assumptions about the noise, and it minimizes the worst-case estimation error (hence the term minimax).

11.2 CONSTRAINED OPTIMIZATION

In this section we show how constrained optimization can be performed through the use of Lagrange multipliers. This background is required for the solution of the H_∞ filtering problem that is presented in Section 11.3. In Section 11.2.1 we will investigate static problems (i.e., problems in which the independent variables are constant). In Section 11.2.2 we will take a brief segue to look at problems with inequality constraints. In Section 11.2.3 we will extend our constrained optimization method to dynamic problems (i.e., problems in which the independent variables change with time).

11.2.1 Static constrained optimization

Suppose we want to minimize some scalar function $J(x, w)$ with respect to x and w. x is an n-dimensional vector, and w is an m-dimensional vector. w is the independent variable and x is the dependent variable; that is, x is somehow determined by w. Suppose our vector-valued constraint is given as $f(x, w) = 0$. Further assume that the dimension of $f(x, w)$ is the same as the dimension of x. This problem can be written as

$$\min_{x,w} J(x, w) \text{ such that } f(x, w) = 0 \tag{11.13}$$

Suppose that the constrained minimum of $J(x, w)$ occurs at $x = x^*$ and $w = w^*$. We call this the stationary point of $J(x, w)$. Now suppose that we choose values of x and w such that x is close to x^*, w is close to w^*, and $f(x, w) = 0$. Expanding $J(x, w)$ and $f(x, w)$ in a Taylor series around x^* and w^* gives

$$\begin{aligned} J(x, w) &= J(x^*, w^*) + \left.\frac{\partial J}{\partial x}\right|_{x^*,w^*} \Delta x + \left.\frac{\partial J}{\partial w}\right|_{x^*,w^*} \Delta w \\ f(x, w) &= f(x^*, w^*) + \left.\frac{\partial f}{\partial x}\right|_{x^*,w^*} \Delta x + \left.\frac{\partial f}{\partial w}\right|_{x^*,w^*} \Delta w \end{aligned} \tag{11.14}$$

where higher-order terms have been neglected (with the assumption that x is close to x^*, and w is close to w^*), $\Delta x = x - x^*$, and $\Delta w = w - w^*$. These equations can be written as

$$\begin{aligned} \Delta J(x, w) &= J(x, w) - J(x^*, w^*) \\ &= \left.\frac{\partial J}{\partial x}\right|_{x^*,w^*} \Delta x + \left.\frac{\partial J}{\partial w}\right|_{x^*,w^*} \Delta w \\ \Delta f(x, w) &= f(x, w) - f(x^*, w^*) \\ &= \left.\frac{\partial f}{\partial x}\right|_{x^*,w^*} \Delta x + \left.\frac{\partial f}{\partial w}\right|_{x^*,w^*} \Delta w \end{aligned} \tag{11.15}$$

Now note that for values of x and w that are close to x^* and w^*, we have $\Delta J(x, w) = 0$. This is because the partial derivatives on the right side of the $\Delta J(x, w)$ equation are zero at the stationary point of $J(x, w)$. We also see that $\Delta f(x, w) = 0$ at the stationary point of $J(x, w)$. This is because $f(x^*, w^*) = 0$ at the constrained stationary point of $J(x, w)$, and we chose x and w such that $f(x, w) = 0$ also. The above equations can therefore be written as

$$\begin{aligned}
\left.\frac{\partial J}{\partial x}\right|_{x^*,w^*} \Delta x + \left.\frac{\partial J}{\partial w}\right|_{x^*,w^*} \Delta w &= 0 \\
\left.\frac{\partial f}{\partial x}\right|_{x^*,w^*} \Delta x + \left.\frac{\partial f}{\partial w}\right|_{x^*,w^*} \Delta w &= 0
\end{aligned} \tag{11.16}$$

These equations are true for arbitrary x and w that are close to x^* and w^* and that satisfy the constraint $f(x, w) = 0$. Equation (11.16) can be solved for Δx as

$$\Delta x = -\left(\left.\frac{\partial f}{\partial x}\right|_{x^*,w^*}\right)^{-1} \left.\frac{\partial f}{\partial w}\right|_{x^*,w^*} \Delta w \tag{11.17}$$

This can be substituted into Equation (11.16) to obtain

$$\left.\frac{\partial J}{\partial w}\right|_{x^*,w^*} - \left.\frac{\partial J}{\partial x}\right|_{x^*,w^*} \left(\left.\frac{\partial f}{\partial x}\right|_{x^*,w^*}\right)^{-1} \left.\frac{\partial f}{\partial w}\right|_{x^*,w^*} = 0 \tag{11.18}$$

This equation, combined with the constraint $f(x, w) = 0$, gives us $(m+n)$ equations that can be solved for the vectors w and x to find the constrained stationary point of $J(x, w)$.

Now consider the augmented cost function

$$J_a = J + \lambda^T f \tag{11.19}$$

where λ is an n-element unknown constant vector called a Lagrange multiplier. Note that

$$\begin{aligned}
\frac{\partial J_a}{\partial x} &= \frac{\partial J}{\partial x} + \lambda^T \frac{\partial f}{\partial x} \\
\frac{\partial J_a}{\partial w} &= \frac{\partial J}{\partial w} + \lambda^T \frac{\partial f}{\partial w} \\
\frac{\partial J_a}{\partial \lambda} &= f
\end{aligned} \tag{11.20}$$

If we set all three of these equations equal to zero then we have

$$\begin{aligned}
\lambda^T &= \frac{-\partial J}{\partial x}\left(\frac{\partial f}{\partial x}\right)^{-1} \\
\frac{\partial J}{\partial w} - \frac{\partial J}{\partial x}\left(\frac{\partial f}{\partial x}\right)^{-1} \frac{\partial f}{\partial w} &= 0 \\
f &= 0
\end{aligned} \tag{11.21}$$

The first equation gives us the value of the Lagrange multiplier, the second equation is identical to Equation (11.18), and the third equation forces the constraint to be satisfied. We therefore see that we can solve the original constrained problem by creating an augmented cost function J_a, taking the partial derivatives with respect to x, w, and λ, setting them equal to zero, and solving for x, w, and λ. The partial derivative equations give us $(2n + m)$ equations to solve for the n-element vector x, the m-element vector w, and the n-element vector λ. We have increased the dimension of the original problem by introducing a Lagrange multiplier, but we have transformed the constrained optimization problem into an unconstrained optimization problem, which can simplify the problem considerably.

■ **EXAMPLE 11.1**

Suppose we need to find the minimum of $J(x,u) = x^2/2 + xu + u^2 + u$ with respect to x and u such that $f(x,u) = x - 3 = 0$. This simple example can be solved by simply realizing that $x = 3$ in order to satisfy the constraint. Substituting $x = 3$ into $J(x,u)$ gives $J(x,u) = 9/2 + 4u + u^2$. Setting the derivative with respect to u equal to zero and solving for u gives $u = -2$.

We can also solve this problem using the Lagrange multiplier method. We create an augmented cost function as

$$\begin{aligned} J_a &= J + \lambda^T f \\ &= x^2/2 + xu + u^2 + u + \lambda(x-3) \end{aligned} \tag{11.22}$$

The Lagrange multiplier λ has the same dimension as x (scalar in this example). The three necessary conditions for a constrained stationary point of J are obtained by setting the partial derivations of Equation (11.20) equal to 0.

$$\begin{aligned} \frac{\partial J_a}{\partial x} &= x + u + \lambda = 0 \\ \frac{\partial J_a}{\partial u} &= x + 2u + 1 = 0 \\ \frac{\partial J_a}{\partial \lambda} &= x - 3 = 0 \end{aligned} \tag{11.23}$$

Solving these three equations for x, u, and λ gives $x = 3$, $u = -2$, and $\lambda = -1$. In this example the Lagrange multiplier method seems to require more effort than simply solving the problem directly. However, in more complicated constrained optimization problems the Lagrange multiplier method is essential for finding a solution.

$\nabla\nabla\nabla$

11.2.2 Inequality constraints

Suppose that we want to minimize a scalar function that is subject to an inequality constraint:

$$\min J(x) \text{ such that } f(x) \leq 0 \tag{11.24}$$

This can be reduced to two minimization problems, neither of which contain inequality constraints. The first minimization problem is unconstrained, and the second minimization problem has an equality constraint:

1. $\min J(x)$

2. $\min J(x)$ such that $f(x) = 0$

In other words, the optimal value of x is either not on the constraint boundary [i.e., $f(x) < 0$], or it is on the constraint boundary [i.e., $f(x) = 0$]. If it is not on the constraint boundary then $f(x) < 0$ and the optimal value of x is obtained by solving the problem without the constraint. If it is on the constraint boundary then $f(x) = 0$ at the constrained minimum, and the optimal value of x is obtained by solving the problem with the equality constraint $f(x) = 0$.

The procedure for solving Equation (11.24) involves solving the unconstrained problem first. Then we check to see if the unconstrained minimum satisfies the constraint. If the unconstrained minimum satisfies the constraint, then the unconstrained minimum solves the inequality-constrained minimization problem and we are done. However, if the unconstrained minimum does not satisfy the constraint, then the minimization problem with the inequality constraint is equivalent to the minimization problem with the equality constraint. So we solve the problem with the equality constraint $f(x) = 0$ to obtain the final solution. This is illustrated for the scalar case in Figure 11.1.

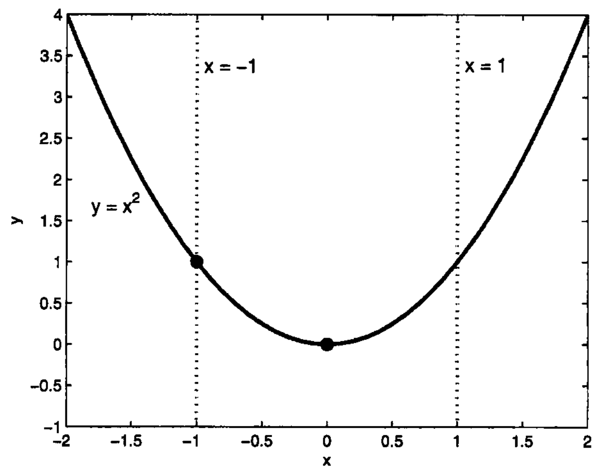

Figure 11.1 This illustrates the constrained minimization of x^2. If the constraint is $x - 1 \leq 0$, then the constrained minimum is equal to the unconstrained minimum and occurs at $x = 0$. If the constraint is $x + 1 \leq 0$, then the constrained minimum can be solved by enforcing the equality constraint $x + 1 = 0$ and occurs at $x = -1$.

When we extend this idea to more than one dimension, we obtain the following procedure, which is called the active-set method for optimization with inequality constraints [Fle81, Gil81].

1. The problem is to minimize $J(x)$ such that $f(x) \leq 0$, where $f(x)$ is an m-element constraint function and the inequality is taken one element at a time.

2. First solve the unconstrained minimization problem. If the unconstrained solution satisfies the constraint $f(x) \leq 0$ then the problem is solved. If not, continue to the next step.

3. For all possible combinations of constraints, solve the problem using those constraints as equality constraints. If the solution satisfies the remaining (unused) constraints, then the solution is feasible. Note that this step requires the solution of $(2^m - 1)$ constrained optimization problems.

4. Out of all the feasible solutions that were obtained in the previous step, the one with the smallest $J(x)$ is the solution to the constrained minimization problem.

Note that there are also other methods for solving optimization problems with inequality constraints, including primal–dual interior-point methods [Wri97].

11.2.3 Dynamic constrained optimization

In this section we extend the Lagrange multiplier method of constrained optimization to the optimization of dynamic systems. Suppose that we have a dynamic system given as

$$x_{k+1} = F_k x_k + w_k \qquad (k = 0, \cdots, N-1) \tag{11.25}$$

where x_k is an n-dimensional state vector. We want to minimize the scalar function

$$J = \psi(x_0) + \sum_{k=0}^{N-1} \mathcal{L}_k \tag{11.26}$$

where $\psi(x_0)$ is a known function of x_0, and $\mathcal{L}_k$ is a known function of x_k and w_k. This is a constrained dynamic optimization problem similar to the type that arises in optimal control [Lew86a, Ste94]. It is slightly different than typical optimal control problems because $\psi(x_k)$ in the above equation is evaluated at the initial time ($k = 0$) instead of the final time ($k = N$), but the methods of optimal control can be used with only slight modifications to solve our problem. The constraints are given in Equation (11.25). From the previous section we know that we can solve this problem by introducing a Lagrange multiplier λ, creating an augmented cost function J_a, and then setting the partial derivatives of J_a with respect to x_k, w_k, and λ equal to zero. Since we have N constraints in Equation (11.25) (each of dimension n), we have to introduce N Lagrange multipliers $\lambda_1, \cdots, \lambda_N$ (each of dimension n). The augmented cost function is therefore written as

$$J_a = \psi(x_0) + \sum_{k=0}^{N-1} \left[\mathcal{L}_k + \lambda_{k+1}^T (F_k x_k + w_k - x_{k+1})\right] \tag{11.27}$$

This can be written as

$$\begin{aligned} J_a &= \psi(x_0) + \sum_{k=0}^{N-1} \left[\mathcal{L}_k + \lambda_{k+1}^T (F_k x_k + w_k)\right] - \sum_{k=0}^{N-1} \lambda_{k+1}^T x_{k+1} \\ &= \psi(x_0) + \sum_{k=0}^{N-1} \left[\mathcal{L}_k + \lambda_{k+1}^T (F_k x_k + w_k)\right] - \sum_{k=0}^{N} \lambda_k^T x_k + \lambda_0^T x_0 \end{aligned} \tag{11.28}$$

where λ_0 is now an additional term in the Lagrange multiplier sequence. It is not in the original augmented cost function, but we will see in Section 11.3 that its value will be determined when we solve the constrained optimization problem. Now we define the Hamiltonian $\mathcal{H}_k$ as

$$\mathcal{H}_k = \mathcal{L}_k + \lambda_{k+1}^T (F_k x_k + w_k) \tag{11.29}$$

With this notation we can write the augmented cost function as follows.

$$
\begin{aligned}
J_a &= \psi(x_0) + \sum_{k=0}^{N-1} \mathcal{H}_k - \sum_{k=0}^{N} \lambda_k^T x_k + \lambda_0^T x_0 \\
&= \psi(x_0) + \sum_{k=0}^{N-1} \mathcal{H}_k - \sum_{k=0}^{N-1} \lambda_k^T x_k - \lambda_N^T x_N + \lambda_0^T x_0 \\
&= \psi(x_0) + \sum_{k=0}^{N-1} \left(\mathcal{H}_k - \lambda_k^T x_k\right) - \lambda_N^T x_N + \lambda_0^T x_0
\end{aligned} \tag{11.30}
$$

The conditions that are required for a constrained stationary point are

$$
\begin{aligned}
\frac{\partial J_a}{\partial x_k} &= 0 \quad (k = 0, \cdots, N) \\
\frac{\partial J_a}{\partial w_k} &= 0 \quad (k = 0, \cdots, N-1) \\
\frac{\partial J_a}{\partial \lambda_k} &= 0 \quad (k = 0, \cdots, N)
\end{aligned} \tag{11.31}
$$

These conditions can also be written as

$$
\begin{aligned}
\frac{\partial J_a}{\partial x_0} &= 0 \\
\frac{\partial J_a}{\partial x_N} &= 0 \\
\frac{\partial J_a}{\partial x_k} &= 0 \quad (k = 1, \cdots, N-1) \\
\frac{\partial J_a}{\partial w_k} &= 0 \quad (k = 0, \cdots, N-1) \\
\frac{\partial J_a}{\partial \lambda_k} &= 0 \quad (k = 0, \cdots, N)
\end{aligned} \tag{11.32}
$$

The fifth condition ensures that the constraint $x_{k+1} = F_k x_k + w_k$ is satisfied. Based on the expression for J_a in Equation (11.30), the first four conditions above can be written as

$$
\begin{aligned}
\lambda_0^T + \frac{\partial \psi_0}{\partial x_0} &= 0 \\
-\lambda_N^T &= 0 \\
\lambda_k^T &= \frac{\partial \mathcal{H}_k}{\partial x_k} \quad (k = 1, \cdots, N-1) \\
\frac{\partial \mathcal{H}_k}{\partial w_k} &= 0 \quad (k = 0, \cdots, N-1)
\end{aligned} \tag{11.33}
$$

This gives us the necessary conditions for a constrained stationary point of our dynamic optimization problem. These are the results that we will use to solve the H_∞ estimation problem in the next section.

11.3 A GAME THEORY APPROACH TO H$_\infty$ FILTERING

The H$_\infty$ solution that we present in this section was originally developed by Ravi Banavar [Ban92] and is further discussed in [She95, She97]. Suppose we have the standard linear discrete-time system

$$\begin{aligned} x_{k+1} &= F_k x_k + w_k \\ y_k &= H_k x_k + v_k \end{aligned} \tag{11.34}$$

where w_k and v_k are noise terms. These noise terms may be random with possibly unknown statistics, or they may be deterministic. They may have a nonzero mean. Our goal is to estimate a linear combination of the state. That is, we want to estimate z_k, which is given by

$$z_k = L_k x_k \tag{11.35}$$

where L_k is a user-defined matrix (assumed to be full rank). If we want to directly estimate x_k (as in the Kalman filter) then we set $L_k = I$. But in general we may only be interested in certain linear combinations of the state. Our estimate of z_k is denoted $\hat{z}_k$, and our estimate of the state at time 0 is denoted $\hat{x}_0$. We want to estimate z_k based on measurements up to and including time $(N-1)$. In the game theory approach to H$_\infty$ filtering we define the following cost function:

$$J_1 = \frac{\sum_{k=0}^{N-1} ||z_k - \hat{z}_k||_{S_k}^2}{||x_0 - \hat{x}_0||_{P_0^{-1}}^2 + \sum_{k=0}^{N-1} \left(||w_k||_{Q_k^{-1}}^2 + ||v_k||_{R_k^{-1}}^2 \right)} \tag{11.36}$$

Our goal as engineers is to find an estimate $\hat{z}_k$ that minimizes J_1. Nature's goal as our adversary is to find disturbances w_k and v_k, and the initial state x_0, to maximize J_1. Nature's ultimate goal is to maximize the estimation error $(z_k - \hat{z}_k)$. The way that nature maximizes $(z_k - \hat{z}_k)$ is by a clever choice of w_k, v_k, and x_0. Nature could maximize $(z_k - \hat{z}_k)$ by simply using infinite magnitudes for w_k, v_k, and x_0, but this would not make the game fair. That is why we define J_1 with $(x_0 - \hat{x}_0)$, w_k, and v_k in the denominator. If nature uses large magnitudes for w_k, v_k, and x_0 then $(z_k - \hat{z}_k)$ will be large, but J_1 may not be large because of the denominator. The form of J_1 prevents nature from using brute force to maximize $(z_k - \hat{z}_k)$. Instead, nature must try to be clever in its choice of w_k, v_k, and x_0 as it tries to maximize $(z_k - \hat{z}_k)$. Likewise, we as engineers must be clever in finding an estimation strategy to minimize $(z_k - \hat{z}_k)$.

This discussion highlights a fundamental difference in the philosophy of the Kalman filter and the H$_\infty$ filter. In Kalman filtering, nature is assumed to be indifferent. The pdf of the noise is given. We (as filter designers) know the pdf of the noise and can use that knowledge to obtain a statistically optimal state estimate. But nature cannot change the pdf to degrade our state estimate. In H$_\infty$ filtering, nature is assumed to be perverse and actively seeks to degrade our state estimate as much as possible. Intuition and experience seem to indicate that neither of these extreme viewpoints of nature is entirely correct, but reality probably lies somewhere in the middle.[1]

[1]Nevertheless, it is advisable to remember the *principle of perversity of inanimate objects* [Bar01, p. 96] – for instance, when dropping a piece of buttered toast on the floor, the probability is significantly more than 50% that the toast will land buttered-side down.

P_0, Q_k, R_k, and S_k in Equation (11.36) are symmetric positive definite matrices chosen by the engineer based on the specific problem. For example, if the user is particularly interested in obtaining an accurate estimate of the third element of z_k, then $S_k(3,3)$ should be chosen to be large relative to the other elements of S_k. If the user knows *a priori* that the second element of the w_k disturbance is small, then $Q_k(2,2)$ should be chosen to be small relative to the other elements of Q_k. In this way, we see that P_0, Q_k, and R_k are analogous to those same quantities in the Kalman filter, if those quantities are known. That is, suppose that we know that the initial estimation error, the process noise, and the measurement noise are zero-mean. Further suppose that we know their covariances. Then we should use those quantities for P_0, Q_k, and R_k in the H$_\infty$ estimation problem. In the Kalman filter, there is no analogy to the S_k matrix given in Equation (11.36). The Kalman filter minimizes the S_k-weighted sum of estimation-error variances for all positive definite S_k matrices (see Section 5.2). But in the H$_\infty$ filter, we will see that the choice of S_k affects the filter gain.

The direct minimization of J_1 is not tractable, so instead we choose a performance bound and seek an estimation strategy that satisfies the threshold. That is, we will try to find an estimate $\hat{z}_k$ that results in

$$J_1 < \frac{1}{\theta} \tag{11.37}$$

where θ is our user-specified performance bound. Rearranging this equation results in

$$\begin{aligned} J &= \frac{-1}{\theta}||x_0 - \hat{x}_0||^2_{P_0^{-1}} + \sum_{k=0}^{N-1}\left[||z_k - \hat{z}_k||^2_{S_k} - \frac{1}{\theta}\left(||w_k||^2_{Q_k^{-1}} + ||v_k||^2_{R_k^{-1}}\right)\right] \\ &< 1 \end{aligned} \tag{11.38}$$

where J is defined by the above equation. The minimax problem becomes

$$J^* = \min_{\hat{z}_k} \max_{w_k, v_k, x_0} J \tag{11.39}$$

Since $z_k = L_k x_k$, we naturally choose $\hat{z}_k = L_k \hat{x}_k$ and try to find the $\hat{x}_k$ that minimizes J. This gives us the problem

$$J^* = \min_{\hat{x}_k} \max_{w_k, v_k, x_0} J \tag{11.40}$$

Nature is choosing x_0, w_k, and v_k to maximize J. But x_0, w_k, and v_k completely determine y_k, so we can replace the v_k in the minimax problem with y_k. We therefore have

$$J^* = \min_{\hat{x}_k} \max_{w_k, y_k, x_0} J \tag{11.41}$$

Since $y_k = H_k x_k + v_k$, we see that $v_k = y_k - H_k x_k$ and

$$||v_k||^2_{R_k^{-1}} = ||y_k - H_k x_k||^2_{R_k^{-1}} \tag{11.42}$$

Since $z_k = L_k x_k$ and $\hat{z}_k = L_k \hat{x}_k$, we see that

$$\begin{aligned} ||z_k - \hat{z}_k||^2_{S_k} &= (z_k - \hat{z}_k)^T S_k (z_k - \hat{z}_k) \\ &= (x_k - \hat{x}_k)^T L_k^T S_k L_k (x_k - \hat{x}_k) \\ &= ||x_k - \hat{x}_k||^2_{\bar{S}_k} \end{aligned} \tag{11.43}$$

where $\bar{S}_k$ is defined as

$$\bar{S}_k = L_k^T S_k L_k \tag{11.44}$$

We substitute these results in Equation (11.38) to obtain

$$\begin{aligned} J &= \frac{-1}{\theta}||x_0 - \hat{x}_0||^2_{P_0^{-1}} + \sum_{k=0}^{N-1}\left[||x_k - \hat{x}_k||^2_{\bar{S}_k} - \frac{1}{\theta}\left(||w_k||^2_{Q_k^{-1}} + ||y_k - H_k x_k||^2_{R_k^{-1}}\right)\right] \\ &= \psi(x_0) + \sum_{k=0}^{N-1}\mathcal{L}_k \end{aligned} \tag{11.45}$$

where $\psi(x_0)$ and $\mathcal{L}_k$ are defined by the above equation. To solve the minimax problem, we will first find a stationary point of J with respect to x_0 and w_k, and then we will find a stationary point of J with respect to $\hat{x}_k$ and y_k.

11.3.1 Stationarity with respect to x_0 and w_k

The problem in this section is to maximize $J = \psi(x_0) + \sum_{k=0}^{N-1}\mathcal{L}_k$ (subject to the constraint $x_{k+1} = F_k x_k + w_k$) with respect to x_0 and w_k. This is the dynamic constrained optimization problem that we solved in Section 11.2.3. The Hamiltonian for this problem is defined as

$$\mathcal{H}_k = \mathcal{L}_k + \frac{2\lambda_{k+1}^T}{\theta}(F_k x_k + w_k) \tag{11.46}$$

where $2\lambda_{k+1}/\theta$ is the time-varying Lagrange multiplier that must be computed $(k = 0, \cdots, N-1)$. Note that we have defined the Lagrange multiplier as $2\lambda_{k+1}/\theta$ instead of λ_{k+1}. This does not change the solution to the problem, it simply scales the Lagrange multiplier (in hindsight) by a constant to make the ensuing math more straightforward. From Equation (11.33) we know that the constrained stationary point of J (with respect to x_0 and w_k) is solved by the following four equations:

$$\begin{aligned} \frac{2\lambda_0^T}{\theta} + \frac{\partial\psi_0}{\partial x_0} &= 0 \\ \frac{2\lambda_N^T}{\theta} &= 0 \\ \frac{\partial\mathcal{H}_k}{\partial w_k} &= 0 \\ \frac{2\lambda_k^T}{\theta} &= \frac{\partial\mathcal{H}_k}{\partial x_k} \end{aligned} \tag{11.47}$$

From the first expression in the above equation we obtain

$$\begin{aligned} \frac{2\lambda_0}{\theta} - \frac{2}{\theta}P_0^{-1}(x_0 - \hat{x}_0) &= 0 \\ P_0\lambda_0 - x_0 + \hat{x}_0 &= 0 \\ x_0 &= \hat{x}_0 + P_0\lambda_0 \end{aligned} \tag{11.48}$$

From the second expression in Equation (11.47) we obtain

$$\lambda_N = 0 \tag{11.49}$$

From the third expression in Equation (11.47) we obtain

$$\begin{aligned} -\frac{2}{\theta}Q_k^{-1}w_k + \frac{2}{\theta}\lambda_{k+1} &= 0 \\ w_k &= Q_k\lambda_{k+1} \end{aligned} \tag{11.50}$$

This can be substituted into the process dynamics equation to obtain

$$x_{k+1} = F_k x_k + Q_k\lambda_{k+1} \tag{11.51}$$

From the fourth expression in Equation (11.47) we obtain

$$\begin{aligned} \frac{2\lambda_k}{\theta} &= 2\bar{S}_k(x_k - \hat{x}_k) + \frac{2}{\theta}H_k^T R_k^{-1}(y_k - H_k x_k) + \frac{2}{\theta}F_k^T\lambda_{k+1} \\ \lambda_k &= F_k^T\lambda_{k+1} + \theta\bar{S}_k(x_k - \hat{x}_k) + H_k^T R_k^{-1}(y_k - H_k x_k) \end{aligned} \tag{11.52}$$

At this point we have to make an assumption in order to proceed any further. From Equation (11.48) we know that $x_0 = \hat{x}_0 + P_0\lambda_0$, so we will assume that

$$x_k = \mu_k + P_k\lambda_k \tag{11.53}$$

for all k, where μ_k and P_k are some functions to be determined, with P_0 given, and the initial condition $\mu_0 = \hat{x}_0$. That is, we assume that x_k is an affine function of λ_k. This assumption may or may not turn out to be valid. We will proceed as if the assumption were true, and if our results turn out to be correct then we will know that our assumption was indeed valid. Substituting Equation (11.53) into Equation (11.51) gives

$$\mu_{k+1} + P_{k+1}\lambda_{k+1} = F_k\mu_k + F_kP_k\lambda_k + Q_k\lambda_{k+1} \tag{11.54}$$

Substituting Equation (11.53) into Equation (11.52) gives

$$\lambda_k = F_k^T\lambda_{k+1} + \theta\bar{S}_k(\mu_k + P_k\lambda_k - \hat{x}_k) + H_k^T R_k^{-1}\left[y_k - H_k(\mu_k + P_k\lambda_k)\right] \tag{11.55}$$

Rearranging this equation gives

$$\begin{aligned} &\lambda_k - \theta\bar{S}_kP_k\lambda_k + H_k^T R_k^{-1}H_kP_k\lambda_k = \\ &\qquad F_k^T\lambda_{k+1} + \theta\bar{S}_k(\mu_k - \hat{x}_k) + H_k^T R_k^{-1}(y_k - H_k\mu_k) \end{aligned} \tag{11.56}$$

This can be solved for λ_k as

$$\begin{aligned} \lambda_k &= \left[I - \theta\bar{S}_kP_k + H_k^T R_k^{-1}H_kP_k\right]^{-1} \times \\ &\qquad \left[F_k^T\lambda_{k+1} + \theta\bar{S}_k(\mu_k - \hat{x}_k) + H_k^T R_k^{-1}(y_k - H_k\mu_k)\right] \end{aligned} \tag{11.57}$$

Substituting this expression for λ_k into Equation (11.54) gives

$$\begin{aligned} &\mu_{k+1} + P_{k+1}\lambda_{k+1} = F_k\mu_k + F_kP_k\left[I - \theta\bar{S}_kP_k + H_k^T R_k^{-1}H_kP_k\right]^{-1} \times \\ &\qquad \left[F_k^T\lambda_{k+1} + \theta\bar{S}_k(\mu_k - \hat{x}_k) + H_k^T R_k^{-1}(y_k - H_k\mu_k)\right] + Q_k\lambda_{k+1} \end{aligned} \tag{11.58}$$

This equation can be rearranged as follows:

$$\begin{aligned} &\mu_{k+1} - F_k\mu_k - F_kP_k\left[I - \theta\bar{S}_kP_k + H_k^T R_k^{-1}H_kP_k\right]^{-1} \times \\ &\left[\theta\bar{S}_k(\mu_k - \hat{x}_k) + H_k^T R_k^{-1}(y_k - H_k\mu_k)\right] = \\ &\left[-P_{k+1} + F_kP_k\left[I - \theta\bar{S}_kP_k + H_k^T R_k^{-1}H_kP_k\right]^{-1}F_k^T + Q_k\right]\lambda_{k+1} \end{aligned} \tag{11.59}$$

This equation is satisfied if both sides are zero. Setting the left side of the above equation equal to zero gives

$$\begin{aligned}\mu_{k+1} &= F_k\mu_k + F_kP_k\left[I - \theta\bar{S}_kP_k + H_k^TR_k^{-1}H_kP_k\right]^{-1} \times \\ &\quad \left[\theta\bar{S}_k(\mu_k - \hat{x}_k) + H_k^TR_k^{-1}(y_k - H_k\mu_k)\right] \end{aligned} \tag{11.60}$$

with the initial condition

$$\mu_0 = \hat{x}_0 \tag{11.61}$$

Setting the right side of Equation (11.59) equal to zero gives

$$\begin{aligned}P_{k+1} &= F_kP_k\left[I - \theta\bar{S}_kP_k + H_k^TR_k^{-1}H_kP_k\right]^{-1}F_k^T + Q_k \\ &= F_k\tilde{P}_kF_k^T + Q_k \end{aligned} \tag{11.62}$$

where $\tilde{P}_k$ is defined by the above equation. That is,

$$\begin{aligned}\tilde{P}_k &= P_k\left[I - \theta\bar{S}_kP_k + H_k^TR_k^{-1}H_kP_k\right]^{-1} \\ &= \left[P_k^{-1} - \theta\bar{S}_k + H_k^TR_k^{-1}H_k\right]^{-1} \end{aligned} \tag{11.63}$$

From the above equation we see that if P_k, $\bar{S}_k$, and R_k are symmetric, then $\tilde{P}_k$ will be symmetric. We see from Equation (11.62) that if Q_k is also symmetric, then P_{k+1} will be symmetric. So if P_0, Q_k, R_k, and S_k are symmetric for all k, then $\tilde{P}_k$ and P_k will be symmetric for all k. The values of x_0 and w_k that provide a stationary point of J can be summarized as follows:

$$\begin{aligned}x_0 &= \hat{x}_0 + P_0\lambda_0 \\ w_k &= Q_k\lambda_{k+1} \\ \lambda_N &= 0 \\ \lambda_k &= \left[I - \theta\bar{S}_kP_k + H_k^TR_k^{-1}H_kP_k\right]^{-1} \times \\ &\quad \left[F_k^T\lambda_{k+1} + \theta\bar{S}_k(\mu_k - \hat{x}_k) + H_k^TR_k^{-1}(y_k - H_k\mu_k)\right] \\ P_{k+1} &= F_kP_k\left[I - \theta\bar{S}_kP_k + H_k^TR_k^{-1}H_kP_k\right]^{-1}F_k^T + Q_k \\ \mu_0 &= \hat{x}_0 \\ \mu_{k+1} &= F_k\mu_k + F_kP_k\left[I - \theta\bar{S}_kP_k + H_k^TR_k^{-1}H_kP_k\right]^{-1} \times \\ &\quad \left[\theta\bar{S}_k(\mu_k - \hat{x}_k) + H_k^TR_k^{-1}(y_k - H_k\mu_k)\right] \end{aligned} \tag{11.64}$$

The fact that we were able to find a stationary point of J shows that we were correct in our assumption that x_k was an affine function of λ_k. In the following section, given these values of x_0 and w_k, we will find the values of $\hat{x}_k$ and y_k that provide a stationary point of J.

11.3.2 Stationarity with respect to $\hat{x}$ and y

The problem in this section is to find a stationary point (with respect to $\hat{x}_k$ and y_k) of $J = \psi(x_k)|_{k=0} + \sum_{k=0}^{N-1}\mathcal{L}_k$ (subject to the constraint $x_{k+1} = F_kx_k + w_k$). This problem is solved given the fact that x_0 and w_k have already been set to their maximizing values as described in Section 11.3.1. From Equation (11.53), and the initial condition of μ_k in Equation (11.61), we see that

$$\begin{aligned} \lambda_k &= P_k^{-1}(x_k - \mu_k) \\ \lambda_0 &= P_0^{-1}(x_0 - \hat{x}_0) \end{aligned} \tag{11.65}$$

We therefore obtain

$$\begin{aligned} ||\lambda_0||_{P_0}^2 &= \lambda_0^T P_0 \lambda_0 \\ &= (x_0 - \hat{x}_0)^T P_0^{-T} P_0 P_0^{-1}(x_0 - \hat{x}_0) \\ &= (x_0 - \hat{x}_0)^T P_0^{-1}(x_0 - \hat{x}_0) \\ &= ||x_0 - \hat{x}_0||_{P_0^{-1}}^2 \end{aligned} \tag{11.66}$$

Therefore, Equation (11.45) becomes

$$J = \frac{-1}{\theta}||\lambda_0||_{P_0}^2 + \sum_{k=0}^{N-1}\left[||x_k - \hat{x}_k||_{\bar{S}_k}^2 - \frac{1}{\theta}\left(||w_k||_{Q_k^{-1}}^2 + ||y_k - H_k x_k||_{R_k^{-1}}^2\right)\right] \tag{11.67}$$

Substituting for x_k from Equation (11.53) in this expression gives

$$\begin{aligned} J = &\frac{-1}{\theta}||\lambda_0||_{P_0}^2 + \\ &\sum_{k=0}^{N-1}\left[||\mu_k + P_k\lambda_k - \hat{x}_k||_{\bar{S}_k}^2 - \frac{1}{\theta}\left(||w_k||_{Q_k^{-1}}^2 + ||y_k - H_k(\mu_k + P_k\lambda_k)||_{R_k^{-1}}^2\right)\right] \end{aligned} \tag{11.68}$$

Consider the term $w_k^T Q_k^{-1} w_k$ in the above equation. Substituting for w_k from Equation (11.50) in this term gives

$$\begin{aligned} w_k^T Q_k^{-1} w_k &= \lambda_{k+1}^T Q_k^T Q_k^{-1} Q_k \lambda_{k+1} \\ &= \lambda_{k+1}^T Q_k \lambda_{k+1} \end{aligned} \tag{11.69}$$

where we have used the fact that Q_k is symmetric. Equation (11.68) can therefore be written as

$$\begin{aligned} J = &\frac{-1}{\theta}||\lambda_0||_{P_0}^2 + \\ &\sum_{k=0}^{N-1}\left[||\mu_k + P_k\lambda_k - \hat{x}_k||_{\bar{S}_k}^2 - \frac{1}{\theta}||y_k - H_k(\mu_k + P_k\lambda_k)||_{R_k^{-1}}^2\right] - \frac{1}{\theta}\sum_{k=0}^{N-1}||\lambda_{k+1}||_{Q_k}^2 \end{aligned} \tag{11.70}$$

Now we take a slight digression to notice that

$$\sum_{k=0}^{N}\lambda_k^T P_k \lambda_k - \sum_{k=0}^{N-1}\lambda_k^T P_k \lambda_k = 0 \tag{11.71}$$

The reason that this equation is correct is because from Equation (11.49) we know that $\lambda_N = 0$. Therefore, the last term in the first summation above is equal to zero and the two summations are equal. The above equation can be written as

$$\begin{aligned}
0 &= \lambda_0^T P_0 \lambda_0 + \sum_{k=1}^{N} \lambda_k^T P_k \lambda_k - \sum_{k=0}^{N-1} \lambda_k^T P_k \lambda_k \\
&= \lambda_0^T P_0 \lambda_0 + \sum_{k=0}^{N-1} \lambda_{k+1}^T P_{k+1} \lambda_{k+1} - \sum_{k=0}^{N-1} \lambda_k^T P_k \lambda_k \\
&= \frac{-1}{\theta} ||\lambda_0||_{P_0}^2 - \frac{1}{\theta} \sum_{k=0}^{N-1} (\lambda_{k+1}^T P_{k+1} \lambda_{k+1} - \lambda_k^T P_k \lambda_k) \qquad (11.72)
\end{aligned}$$

We can subtract this zero term to the cost function of Equation (11.70) to obtain

$$\begin{aligned}
J &= \sum_{k=0}^{N-1} \Big[||\mu_k + P_k\lambda_k - \hat{x}_k||_{\bar{S}_k}^2 - \\
&\quad \frac{1}{\theta}||\lambda_{k+1}||_{Q_k}^2 + \frac{1}{\theta}(\lambda_{k+1}^T P_{k+1}\lambda_{k+1} - \lambda_k^T P_k \lambda_k) - \frac{1}{\theta}||y_k - H_k(\mu_k + P_k\lambda_k)||_{R_k^{-1}}^2 \Big] \\
&= \sum_{k=0}^{N-1} \Big[(\mu_k - \hat{x}_k)^T \bar{S}_k (\mu_k - \hat{x}_k) + 2(\mu_k - \hat{x}_k)^T \bar{S}_k P_k \lambda_k + \lambda_k^T P_k \bar{S}_k P_k \lambda_k + \\
&\quad \frac{1}{\theta}\lambda_{k+1}^T (P_{k+1} - Q_k)\lambda_{k+1} - \frac{1}{\theta}\lambda_k^T P_k \lambda_k - \frac{1}{\theta}(y_k - H_k\mu_k)^T R_k^{-1}(y_k - H_k\mu_k) + \\
&\quad \frac{2}{\theta}(y_k - H_k\mu_k)^T R_k^{-1} H_k P_k \lambda_k - \frac{1}{\theta}\lambda_k^T P_k H_k^T R_k^{-1} H_k P_k \lambda_k \Big] \qquad (11.73)
\end{aligned}$$

Now we consider the term $\lambda_{k+1}^T(P_{k+1} - Q_k)\lambda_{k+1}$ in the above expression. Substituting for P_{k+1} from Equation (11.62) in this term gives

$$\begin{aligned}
\lambda_{k+1}^T(P_{k+1} - Q_k)\lambda_{k+1} &= \lambda_{k+1}^T(Q_k + F_k\tilde{P}_k F_k^T - Q_k)\lambda_{k+1} \\
&= \lambda_{k+1}^T F_k \tilde{P}_k F_k^T \lambda_{k+1} \qquad (11.74)
\end{aligned}$$

But from Equation (11.55) we see that

$$F_k^T \lambda_{k+1} = \lambda_k - \theta\bar{S}_k(\mu_k + P_k\lambda_k - \hat{x}_k) - H_k^T R_k^{-1}[y_k - H_k(\mu_k + P_k\lambda_k)] \qquad (11.75)$$

Substituting this expression for $F_k^T\lambda_{k+1}$ into Equation (11.74) gives

$$\begin{aligned}
&\lambda_{k+1}^T(P_{k+1} - Q_k)\lambda_{k+1} \\
&= \left\{\lambda_k - \theta\bar{S}_k(\mu_k + P_k\lambda_k - \hat{x}_k) - H_k^T R_k^{-1}[y_k - H_k(\mu_k + P_k\lambda_k)]\right\}^T \\
&\quad \tilde{P}_k \left\{\lambda_k - \theta\bar{S}_k(\mu_k + P_k\lambda_k - \hat{x}_k) - H_k^T R_k^{-1}[y_k - H_k(\mu_k + P_k\lambda_k)]\right\} \\
&= \left\{\lambda_k^T(I - \theta P_k\bar{S}_k + P_k H_k^T R_k^{-1} H_k) - \theta(\mu_k - \hat{x}_k)^T\bar{S}_k - \right. \\
&\quad \left. (y_k - H_k\mu_k)^T R_k^{-1} H_k\right\} \tilde{P}_k \left\{\lambda_k^T(I - \theta P_k\bar{S}_k + P_k H_k^T R_k^{-1} H_k) - \right. \\
&\quad \left. \theta(\mu_k - \hat{x}_k)^T\bar{S}_k - (y_k - H_k\mu_k)^T R_k^{-1} H_k\right\}^T \qquad (11.76)
\end{aligned}$$

Now note from Equation (11.63) that $(I - \theta P_k\bar{S}_k + P_k H_k^T R_k^{-1} H_k) = P_k\tilde{P}_k^{-1}$. Making this substitution in the above equation gives the following.

$$
\begin{aligned}
&\lambda_{k+1}^T(P_{k+1}-Q_k)\lambda_{k+1} \\
&= \left\{\lambda_k^T P_k \tilde{P}_k^{-1} - \theta(\mu_k-\hat{x}_k)^T\bar{S}_k - (y_k - H_k\mu_k)^T R_k^{-1}H_k\right\} \\
&\quad \tilde{P}_k \left\{\lambda_k^T P_k \tilde{P}_k^{-1} - \theta(\mu_k-\hat{x}_k)^T\bar{S}_k - (y_k - H_k\mu_k)^T R_k^{-1}H_k\right\}^T \\
&= \lambda_k^T P_k \tilde{P}_k^{-1} P_k \lambda_k - \theta(\mu_k-\hat{x}_k)^T \bar{S}_k P_k \lambda_k - (y_k - H_k\mu_k)^T R_k^{-1} H_k P_k \lambda_k - \\
&\quad \theta \lambda_k P_k \bar{S}_k(\mu_k - \hat{x}_k) + \theta^2(\mu_k-\hat{x}_k)^T\bar{S}_k\tilde{P}_k\bar{S}_k(\mu_k-\hat{x}_k) + \\
&\quad \theta(y_k - H_k\mu_k)^T R_k^{-1} H_k \tilde{P}_k \bar{S}_k(\mu_k-\hat{x}_k) - \lambda_k^T P_k H_k^T R_k^{-1}(y_k - H_k\mu_k) + \\
&\quad \theta(\mu_k-\hat{x}_k)^T\bar{S}_k\tilde{P}_k H_k^T R_k^{-1}(y_k - H_k\mu_k) + \\
&\quad (y_k - H_k\mu_k)^T R_k^{-1} H_k \tilde{P}_k H_k^T R_k^{-1}(y_k - H_k\mu_k)
\end{aligned}
\tag{11.77}
$$

Notice that the above expression is a scalar. That means that each term on the right side is a scalar, which means that each term is equal to its transpose. For example, consider the second term on the right side. Since it is a scalar, we see that $\theta(\mu_k-\hat{x}_k)^T\bar{S}_k P_k\lambda_k = \theta\lambda_k^T P_k \bar{S}_k(\mu_k - \hat{x}_k)$. (We have used the fact that P_k and $\bar{S}_k$ are symmetric, and θ is a scalar.) Equation (11.77) can therefore be written as

$$
\begin{aligned}
&\lambda_{k+1}^T(P_{k+1}-Q_k)\lambda_{k+1} \\
&= \lambda_k^T P_k \tilde{P}_k^{-1} P_k \lambda_k - 2\theta(\mu_k-\hat{x}_k)^T \bar{S}_k P_k \lambda_k - \\
&\quad 2(y_k - H_k\mu_k)^T R_k^{-1} H_k P_k \lambda_k + \theta^2(\mu_k-\hat{x}_k)^T\bar{S}_k\tilde{P}_k\bar{S}_k(\mu_k-\hat{x}_k) + \\
&\quad 2\theta(\mu_k-\hat{x}_k)^T\bar{S}_k\tilde{P}_k H_k^T R_k^{-1}(y_k - H_k\mu_k) + \\
&\quad (y_k - H_k\mu_k)^T R_k^{-1} H_k \tilde{P}_k H_k^T R_k^{-1}(y_k - H_k\mu_k)
\end{aligned}
\tag{11.78}
$$

Now note from Equation (11.63) that

$$
\begin{aligned}
\tilde{P}_k^{-1} &= \left[I - \theta\bar{S}_k P_k + H_k^T R_k^{-1} H_k P_k\right] P_k^{-1} \\
&= P_k^{-1}\left[P_k^{-1} - \theta\bar{S}_k + H_k^T R_k^{-1} H_k\right] P_k^{-1} \\
&= P_k^{-1}\left[I - P_k\theta\bar{S}_k + P_k H_k^T R_k^{-1} H_k\right]
\end{aligned}
\tag{11.79}
$$

We therefore see that

$$
\begin{aligned}
\lambda_k^T P_k \tilde{P}_k^{-1} P_k \lambda_k &= \lambda_k^T\left[I - \theta P_k\bar{S}_k + P_k H_k^T R_k^{-1} H_k\right] P_k\lambda_k \\
&= \lambda_k^T P_k \lambda_k - \theta\lambda_k^T P_k \bar{S}_k P_k \lambda_k + \lambda_k^T P_k H_k^T R_k^{-1} H_k P_k \lambda_k
\end{aligned}
\tag{11.80}
$$

Substituting this into Equation (11.78) gives

$$
\begin{aligned}
&\lambda_{k+1}^T(P_{k+1}-Q_k)\lambda_{k+1} \\
&= \lambda_k^T P_k \lambda_k - \theta\lambda_k^T P_k \bar{S}_k P_k \lambda_k + \lambda_k^T P_k H_k^T R_k^{-1} H_k P_k \lambda_k - \\
&\quad 2\theta(\mu_k-\hat{x}_k)^T\bar{S}_k P_k\lambda_k - 2(y_k - H_k\mu_k)^T R_k^{-1} H_k P_k \lambda_k + \\
&\quad \theta^2(\mu_k-\hat{x}_k)^T\bar{S}_k\tilde{P}_k\bar{S}_k(\mu_k-\hat{x}_k) + 2\theta(\mu_k-\hat{x}_k)^T\bar{S}_k\tilde{P}_k H_k^T R_k^{-1}(y_k - H_k\mu_k) + \\
&\quad (y_k - H_k\mu_k)^T R_k^{-1} H_k \tilde{P}_k H_k^T R_k^{-1}(y_k - H_k\mu_k)
\end{aligned}
\tag{11.81}
$$

Substituting this equation for $\lambda_{k+1}^T(P_{k+1}-Q_k)\lambda_{k+1}$ into Equation (11.73) gives the following.

$$
\begin{aligned}
J &= \sum_{k=0}^{N-1} \Big[(\mu_k - \hat{x}_k)^T \bar{S}_k (\mu_k - \hat{x}_k) - \frac{1}{\theta}(y_k - H_k\mu_k)^T R_k^{-1}(y_k - H_k\mu_k) + \\
&\quad \theta(\mu_k - \hat{x})_k^T \bar{S}_k \tilde{P}_k \bar{S}_k (\mu_k - \hat{x}_k) + 2(\mu_k - \hat{x}_k)^T \bar{S}_k \tilde{P}_k H_k^T R_k^{-1}(y_k - H_k\mu_k) + \\
&\quad \frac{1}{\theta}(y_k - H_k\mu_k)^T R_k^{-1} H_k \tilde{P}_k H_k^T R_k^{-1}(y_k - H_k\mu_k) \Big] \\
&= \sum_{k=0}^{N-1} \Big[(\mu_k - \hat{x}_k)^T (\bar{S}_k + \theta \bar{S}_k \tilde{P}_k \bar{S}_k)(\mu_k - \hat{x}_k) + \\
&\quad 2(\mu_k - \hat{x}_k)^T \bar{S}_k \tilde{P}_k H_k^T R_k^{-1}(y_k - H_k\mu_k) + \\
&\quad \frac{1}{\theta}(y_k - H_k\mu_k)^T (R_k^{-1} H_k \tilde{P}_k H_k^T R_k^{-1} - R_k^{-1})(y_k - H_k\mu_k) \Big] \qquad (11.82)
\end{aligned}
$$

Now recall our original objective: we are trying to find the stationary point of J with respect to $\hat{x}_k$ and y_k. If we take the partial derivative of the above expression for J with respect to $\hat{x}_k$ and y_k and set them equal to 0 we obtain

$$
\begin{aligned}
\frac{\partial J}{\partial \hat{x}_k} &= 2(\bar{S}_k + \theta \bar{S}_k \tilde{P}_k \bar{S}_k)(\hat{x}_k - \mu_k) + 2\bar{S}_k \tilde{P}_k H_k^T R_k^{-1}(H_k\mu_k - y_k) \\
&= 0 \\
\frac{\partial J}{\partial y_k} &= \frac{2}{\theta}(R_k^{-1} H_k \tilde{P}_k H_k^T R_k^{-1} - R_k^{-1})(y_k - H_k\mu_k) + 2R_k^{-1} H_k \tilde{P}_k \bar{S}_k (\mu_k - \hat{x}_k) \\
&= 0 \qquad (11.83)
\end{aligned}
$$

These equations are clearly satisfied for the following values of $\hat{x}_k$ and y_k:

$$
\begin{aligned}
\hat{x}_k &= \mu_k \\
y_k &= H_k\mu_k \qquad (11.84)
\end{aligned}
$$

These are the extremizing values of $\hat{x}_k$ and y_k. However, we still are not sure if these extremizing values give a local minimum or maximum of J. Recall that the second derivative of J tells us what kind of stationary point we have. If the second derivative is positive definite, then our stationary point is a minimum. If the second derivative is negative definite, then our stationary point is a maximum. If the second derivative has both positive and negative eigenvalues, then our stationary point is a saddle point. The second derivative of J with respect to $\hat{x}_k$ can be computed as

$$
\frac{\partial^2 J}{\partial \hat{x}_k^2} = 2(\bar{S}_k + \theta \bar{S}_k \tilde{P}_k \bar{S}_k) \qquad (11.85)
$$

Our $\hat{x}_k$ will therefore be a minimizing value of J if $(\bar{S}_k + \theta \bar{S}_k \tilde{P}_k \bar{S}_k)$ is positive definite. The value of S_k chosen for use in Equation (11.36) should always be positive definite, which means that $\bar{S}_k$ defined in Equation (11.44) will be positive definite. This means that our $\hat{x}_k$ will be a minimizing value of J if $\tilde{P}_k$ is positive definite.

So, from the definition of $\tilde{P}_k$ in Equation (11.63), the condition required for $\hat{x}_k$ to minimize J is that $\left(P_k^{-1} - \theta \bar{S}_k + H_k^T R_k^{-1} H_k\right)^{-1}$ be positive definite. This is equivalent to requiring that $\left(P_k^{-1} - \theta\bar{S}_k + H_k^T R_k^{-1} H_k\right)$ be positive definite. The individual terms in this expression are always positive definite [note in particular from Equation (11.62) that P_k will be positive definite if $\tilde{P}_k$ is positive definite]. So the condition for $\hat{x}_k$ to minimize J is that $\theta\bar{S}_k$ be "small enough" so that $\left(P_k^{-1} - \theta\bar{S}_k + H_k^T R_k^{-1} H_k\right)$ is positive definite. Requiring that $\theta\bar{S}_k$ be small can be accomplished three different ways.

1. $\theta\bar{S}_k$ will be small if θ is small. This means that the performance requirement specified in Equation (11.37) is not too stringent. As long as our performance requirement is not too stringent then the problem will have a solution. If, however, the performance requirement is too stringent (i.e., θ is large) then the problem will not have a solution.

2. $\theta\bar{S}_k$ will be small if L_k is small. This statement is based on the relationship between $\bar{S}_k$ and L_k as shown in Equation (11.44). From Equation (11.36) we see that the numerator of the cost function is given as $(x_k - \hat{x}_k)^T L_k^T S_k L_k (x_k - \hat{x}_k)$. So if L_k is small we see that the numerator of the cost function will be small, which means that it will be easier to minimize the cost function. If, however, L_k is too large, then the problem will not have a solution.

3. $\theta\bar{S}_k$ will be small if S_k is small. This statement is based on the relationship between $\bar{S}_k$ and S_k as shown in Equation (11.44). From Equation (11.36) we see that the numerator of the cost function is given as $(x_k - \hat{x}_k)^T L_k^T S_k L_k (x_k - \hat{x}_k)$. So if S_k is small we see that the numerator of the cost function will be small, which means that it will be easier to minimize the cost function. If, however, S_k is too large, then the problem will not have a solution.

Note from Equation (11.62) that the positive definiteness of $\tilde{P}_k$ implies the positive definiteness of P_{k+1}. Therefore, if P_0 is positive definite (per our original problem statement), and $\tilde{P}_k$ is positive definite for all k, then P_k will also be positive definite for all k.

It is also academically interesting (though of questionable utility) to note the conditions under which the y_k that we found in Equation (11.84) will be a maximizing value of J. (Recall that y_k is chosen by nature, our adversary, to maximize the cost function.) The second derivative of J with respect to y_k can be computed as

$$\begin{aligned} \frac{\partial^2 J}{\partial y_k^2} &= \frac{2}{\theta}(R_k^{-1} H_k \tilde{P}_k H_k^T R_k^{-1} - R_k^{-1}) \\ &= \frac{2}{\theta} R_k^{-1} (H_k \tilde{P}_k H_k^T - R_k) R_k^{-1} \end{aligned} \tag{11.86}$$

R_k and R_k^{-1}, specified by the user as part of the problem statement in Equation (11.36), should always be positive definite. So the second derivative above will be negative definite (which means that y_k will be a maximizing value of J) if $(R_k - H_k \tilde{P}_k H_k^T)$ is positive definite. This requirement can be satisfied in two ways.

1. $(R_k - H_k \tilde{P}_k H_k^T)$ will be positive definite if R_k is large enough. A large value of R_k means that the denominator of the cost function of Equation (11.36) will be small, which means that the cost function will be large. A large cost function value is easier to maximize and will therefore tend to have a

maximizing value for y_k. Also note that the designer typically chooses R_k to be proportional to the magnitude of the measurement noise. If the user knows that the measurement noise is large, then R_k will be large, which again will result in a problem with a maximizing value for y_k. In other words, nature will be better able to maximize the cost function if the measurement noise is large.

2. $(R_k - H_k \tilde{P}_k H_k^T)$ will be positive definite if H_k is small enough. If H_k becomes smaller, that means that the measurement noise becomes larger relative to the size of the measurements, as seen in Equation (11.34). In other words, a small value of H_k means a smaller signal-to-noise ratio for the measurements. A small signal-to-noise ratio gives nature a better opportunity to find a maximizing value of y_k.

Of course, we are not really interested in finding a maximizing value of y_k. Our goal was to find the minimizing value of x_k. The H$_\infty$ filter algorithm can be summarized as follows.

The discrete-time H$_\infty$ filter

1. The system equations are given as

$$\begin{aligned} x_{k+1} &= F_k x_k + w_k \\ y_k &= H_k x_k + v_k \\ z_k &= L_k x_k \end{aligned} \tag{11.87}$$

where w_k and v_k are noise terms, and our goal is to estimate z_k.

2. The cost function is given as

$$J_1 = \frac{\sum_{k=0}^{N-1} ||z_k - \hat{z}_k||_{S_k}^2}{||x_0 - \hat{x}_0||_{P_0^{-1}}^2 + \sum_{k=0}^{N-1} \left(||w_k||_{Q_k^{-1}}^2 + ||v_k||_{R_k^{-1}}^2 \right)} \tag{11.88}$$

where P_0, Q_k, R_k, and S_k are symmetric, positive definite matrices chosen by the engineer based on the specific problem.

3. The cost function can be made to be less than $1/\theta$ (a user-specified bound) with the following estimation strategy, which is derived from Equations (11.44), (11.60), (11.62), and (11.84):

$$\begin{aligned} \bar{S}_k &= L_k^T S_k L_k \\ K_k &= P_k \left[I - \theta \bar{S}_k P_k + H_k^T R_k^{-1} H_k P_k \right]^{-1} H_k^T R_k^{-1} \\ \hat{x}_{k+1} &= F_k \hat{x}_k + F_k K_k (y_k - H_k \hat{x}_k) \\ P_{k+1} &= F_k P_k \left[I - \theta \bar{S}_k P_k + H_k^T R_k^{-1} H_k P_k \right]^{-1} F_k^T + Q_k \end{aligned} \tag{11.89}$$

4. The following condition must hold at each time step k in order for the above estimator to be a solution to the problem:

$$P_k^{-1} - \theta \bar{S}_k + H_k^T R_k^{-1} H_k > 0 \tag{11.90}$$

11.3.3 A comparison of the Kalman and H_∞ filters

Comparing the Kalman filter in Equation (11.12) and the H_∞ filter in Equation (11.89) reveals some fascinating connections. For instance, in the H_∞ filter, Q_k, R_k, and P_0 are design parameters chosen by the user based on *a priori* knowledge of the magnitude of the process disturbance w_k, the measurement disturbance v_k, and the initial estimation error $(x_0 - \hat{x}_0)$. In the Kalman filter, w_k, v_k, and $(x_0 - \hat{x}_0)$ are zero-mean, and Q_k, R_k, and P_0 are their respective covariances.

Now suppose we use $L_k = S_k = I$ in the H_∞ filter. That is, we are interested in estimating the entire state, and we want to weight all of the estimation errors equally in the cost function. If we use $\theta = 0$ then the H_∞ filter reduces to the Kalman filter (assuming Q_k, R_k, and P_0 are chosen as above). This provides an interesting interpretation of the Kalman filter; that is, the Kalman filter is the minimax filter in the case that the performance bound in Equation (11.36) is set equal to ∞. We see that although the Kalman filter minimizes the variance of the estimation error (as discussed in Section 5.2), it does not provide any guarantee as far as limiting the worst-case estimation error. That is, it does not guarantee any bound for the cost function of Equation (11.36).

The Kalman and H_∞ filter equations have an interesting difference. If we want to estimate a linear combination of states using the Kalman filter, the estimator is the same regardless of the linear combination that we want to estimate. That is, if we want to estimate $L_k x_k$ using the Kalman filter, the answer is the same regardless of the L_k matrix that we choose. However, in the H_∞ approach, the resulting filter depends strongly on L_k and the particular linear combination of states that we want to estimate.

Note that the H_∞ filter of Equation (11.89) is identical to the Kalman filter except for subtraction of the term $\theta \bar{S}_k P_k$ in the K_k and P_{k+1} equations. Recall from Section 5.5 that the Kalman filter can be made more robust to unmodeled noise and unmodeled dynamics by artificially increasing Q_k in the Kalman filter equations. This results in a larger covariance P_k, which in turn results in a larger gain K_k. From Equation (11.89) we can see that subtracting $\theta \bar{S}_k P_k$ on the right side of the P_{k+1} equation tends to make P_{k+1} larger (since the subtraction is inside a matrix inverse operation). Similarly, subtracting $\theta \bar{S}_k P_k$ on the right side of the K_k equation tends to make K_k larger. Increasing Q_k in the Kalman filter is conceptually the same as increasing P_k and K_k. Therefore, the H_∞ filter equations make intuitive sense when compared with the Kalman filter equations. The H_∞ filter is a worst-case filter in the sense that it assumes that w_k, v_k, and x_0 will be chosen by nature to maximize the cost function. The H_∞ filter is therefore robust by design. Comparing the H_∞ filter with the Kalman filter, we can see that the H_∞ filter is simply a robust version of the Kalman filter. When we robustified the Kalman filter in Section 5.5 to add tolerance to unmodeled noise and dynamics, we did not derive an optimal way to increase Q_k. However, H_∞ filter theory shows us the optimal way to robustify the Kalman filter.

11.3.4 Steady-state H_∞ filtering

If the underlying system and the design parameters are time-invariant, then it may be possible to obtain a steady-state solution to the H_∞ filtering problem. Suppose that our system is given as

$$\begin{aligned} x_{k+1} &= Fx_k + w_k \\ y_k &= Hx_k + v_k \\ z_k &= Lx_k \end{aligned} \tag{11.91}$$

where w_k and v_k are noise terms. Our goal is to estimate z_k such that

$$\lim_{N \to \infty} \frac{\sum_{k=0}^{N-1} ||z_k - \hat{z}_k||_S^2}{\sum_{k=0}^{N-1} \left(||w_k||_{Q^{-1}}^2 + ||v_k||_{R^{-1}}^2 \right)} < \frac{1}{\theta} \tag{11.92}$$

where Q, R, and S are symmetric positive definite matrices chosen by the engineer based on the specific problem. The steady-state filter of Equation (11.89) becomes

$$\begin{aligned} \bar{S} &= L^T S L \\ K &= P\left[I - \theta \bar{S} P + H^T R^{-1} H P\right]^{-1} H^T R^{-1} \\ \hat{x}_{k+1} &= F\hat{x}_k + FK_k(y_k - H\hat{x}_k) \\ P &= FP\left[I - \theta \bar{S} P + H^T R^{-1} H P\right]^{-1} F^T + Q \end{aligned} \tag{11.93}$$

The following condition must hold in order for the above estimator to be a solution to the problem:

$$P^{-1} - \theta \bar{S} + H^T R^{-1} H > 0 \tag{11.94}$$

If θ, L, R, or S is too large, or if H is too small, then the H_∞ estimator will not have a solution. Note that the expression for P in Equation (11.93) can be written as

$$P = F\left[P^{-1} - \theta \bar{S} + H^T R^{-1} H\right]^{-1} F^T + Q \tag{11.95}$$

Applying the matrix inversion lemma to the inverse in the above expression gives

$$\begin{aligned} P &= F\left\{P - P\left[(H^T R^{-1} H - \theta \bar{S})^{-1} + P\right]^{-1} P\right\} F^T + Q \\ &= FPF^T - FP\left[(H^T R^{-1} H - \theta \bar{S})^{-1} + P\right]^{-1} PF^T + Q \end{aligned} \tag{11.96}$$

This is a discrete-time algebraic Riccati equation that can be solved with control system software.[2] If control system software is not available, then the algebraic Riccati equation can be solved by numerically iterating the discrete-time Riccati equation of Equation (11.89) until it converges to a steady-state value. The steady-state filter is much easier to implement in a system in which real-time computational effort or code size is a serious consideration. The disadvantage of the steady-state filter is that (theoretically) it does not perform as well as the time-varying filter. However, the reduced performance that is seen in the steady-state filter is often a small fraction of the optimal performance, whereas the computational savings can be significant.

[2]For example, in MATLAB's Control System Toolbox we can use the command DARE($F^T, I, Q, (H^T R^{-1} H - \theta \bar{S})^{-1}$.

EXAMPLE 11.2

Suppose we are trying to estimate a randomly varying scalar on the basis of noisy measurements. We have the scalar system

$$\begin{aligned} x_{k+1} &= x_k + w_k \\ y_k &= x_k + v_k \\ z_k &= x_k \end{aligned} \tag{11.97}$$

This system could describe our attempt to estimate a noisy voltage. The voltage is essentially constant, but it is subject to random fluctuations, hence the noise term w_k in the process equation. Our measurement of the voltage is also subject to noise or instrument bias, hence the noise term v_k in the measurement equation. We see in this example that $F = H = L = 1$. Further suppose that $Q = R = S = 1$ in the cost function of Equation (11.88). Then the discrete-time Riccati equation associated with the H_∞ filter equations becomes

$$\begin{aligned} P_{k+1} &= F_k P_k \left[I - \theta \bar{S}_k P_k + H_k^T R_k^{-1} H_k P_k\right]^{-1} F_k^T + Q_k \\ &= P_k \left[1 - \theta P_k + P_k\right]^{-1} + 1 \end{aligned} \tag{11.98}$$

This can be solved numerically or analytically as a function of time for a given θ to give P_k, and then the H_∞ gain can be obtained as

$$\begin{aligned} K_k &= P_k \left[I - \theta \bar{S}_k P_k + H_k^T R_k^{-1} H_k P_k\right]^{-1} H_k^T R_k^{-1} \\ &= P_k \left[1 - \theta P_k + P_k\right]^{-1} \end{aligned} \tag{11.99}$$

We can set $P_{k+1} = P_k$ in Equation (11.98) to obtain the steady-state solution for P_k. This gives

$$\begin{aligned} P &= P(1 - \theta P + P)^{-1} + 1 \\ P(1 - \theta P + P) &= P + (1 - \theta P + P) \\ (1-\theta)P^2 + (\theta - 1)P - 1 &= 0 \\ P &= \frac{1 - \theta \pm \sqrt{(\theta - 1)(\theta - 5)}}{2(1-\theta)} \end{aligned} \tag{11.100}$$

As we discussed earlier, in order for this value of P to be a solution to the H_∞ estimation problem, P must be positive definite. The first solution for P is positive if $\theta < 1$, and both solutions for P are positive if $\theta \geq 5$. Another condition for the solution of the H_∞ estimation problem is that

$$\begin{aligned} P^{-1} - \theta \bar{S} + H^T R^{-1} H &> 0 \\ P^{-1} - \theta + 1 &> 0 \end{aligned} \tag{11.101}$$

If $\theta < 1$ then the first solution for P satisfies this bound. However, if $\theta \geq 5$, then neither solution for P satisfies this bound. Combining this data shows that the H_∞ estimator problem has a solution for $\theta < 1$. Every H_∞ estimator problem will have a solution for θ less than some upper bound because of the nature of the cost function.

For a general estimator gain K the estimate can be written as

$$\begin{aligned}\hat{x}_{k+1} &= F\hat{x}_k + FK(y_k - H_k\hat{x}_k) \\ &= (1-K)\hat{x}_k + Ky_k \end{aligned} \tag{11.102}$$

If we choose $\theta = 1/2$, then we obtain $P = 2$ and $K = 1$. As seen from the above equation, this results in $\hat{x}_{k+1} = y_k$. In other words, the estimator ignores the previous estimate and simply sets the estimate equal to the previous measurement. As θ increases toward 1, P increases above 2 and approaches ∞, and the estimator gain K increases greater than 1 and also approaches ∞. In this case, the estimator will actually place a negative weight on the previous estimate and compensate by placing additional weight on the measurement. If θ increases too much (gets too close to 1) then the estimator gain K will be greater than 2 and the H_∞ estimator will be unstable. It is always a good idea to check the stability of your H_∞ filter. If the filter is unstable then you should probably decrease θ to obtain a stable filter. As θ decreases below 1/2, P decreases below 2 and the gain K decreases below 1. In this case, the estimator balances the relative weight placed on the previous estimate and the measurement.

A Kalman filter to estimate x_k is equivalent to an H_∞ filter with $\theta = 0$. In this case, we obtain the positive definite solution of the steady-state Riccati equation as $P = (1 + \sqrt{5})/2$. This gives a steady-state estimator gain $K = (1 + \sqrt{5})/(3 + \sqrt{5}) = (\sqrt{5} - 1)/2 \approx 0.62$. The Kalman filter gain is smaller than the H_∞ filter gain for $\theta > 0$, which means that the Kalman filter relies less on measurements and more on the system model. The Kalman filter gives an optimal estimate if the model and the noise statistics are known, but it may undervalue the measurements if there are errors in the system model or the assumed noise statistics.

Figure 11.2 shows the true state x_k and the estimate $\hat{x}_k$ when the steady-state Kalman and H_∞ filters are used to estimate the state. The H_∞ filter was designed with $\theta = 1/3$, which gave a filter gain $K = (3 + 3\sqrt{7})/(8 + 2\sqrt{7}) \approx 0.82$. The disturbances w_k and v_k were both normally distributed zero-mean white noise sequences with standard deviations equal to 10. The performance of the two filters is very similar. The RMS estimation error of the Kalman filter is 3.6 and the RMS estimation error of the H_∞ filter is 4.1. As expected, the Kalman filter performs better than the H_∞ filter. However, suppose that the process noise has a mean of 10. Figure 11.3 shows the performance of the filters for this situation. In this case the H_∞ filter performs better. The RMS estimation error of the Kalman filter is 15.6 and the RMS estimation error of the H_∞ filter is 12.0.

If we choose $\theta = 1/10$ then we obtain $P = 5/3$ and $K = 2/3$. As θ gets smaller, the H_∞ estimator gain gets closer and closer to the Kalman filter gain.

$\nabla\nabla\nabla$

11.3.5 The transfer function bound of the H_∞ filter

In this section, we show that the steady-state H_∞ filter derived in the previous section bounds the transfer function from the noise to the estimation error, if Q,

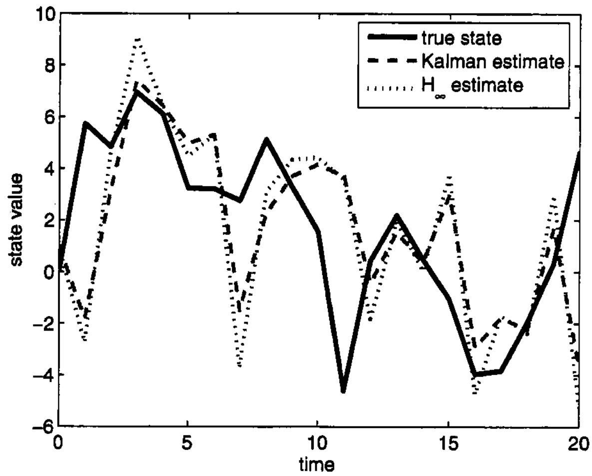

Figure 11.2 Example 11.2 results. Kalman and H_∞ filter peformance when the noise statistics are known. The Kalman gain is 0.62 and the H_∞ gain is 0.82. The Kalman filter performs about 12% better than the H_∞ filter.

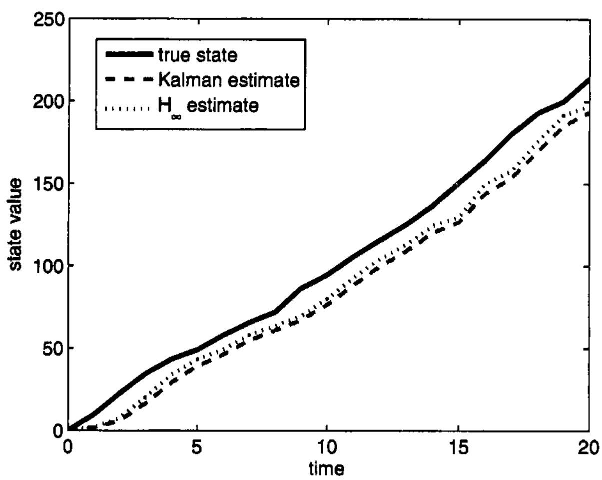

Figure 11.3 Example 11.2 results. Kalman and H_∞ filter peformance when the process noise is biased. The Kalman gain is 0.62 and the H_∞ gain is 0.82. The H_∞ filter performs about 23% better than the Kalman filter.

R, and S are all identity matrices. Recall that the two-norm of a column vector x is defined as

$$||x||_2^2 = x^T x \tag{11.103}$$

Now suppose we have a time-varying vector $x_0, x_1, x_2, \cdots$. The signal two-norm of x is defined as

$$||x||_2^2 = \sum_{k=0}^{\infty} ||x_k||_2^2 \tag{11.104}$$

That is, the square of the signal two-norm is the sum of all of the squares of the vector two-norms that are taken at each time step.[3] Now suppose that we have a system with input u and output x, and the transfer function is $G(z)$. If the input u is comprised entirely of signals at the frequency ω and the sample time of the system is T, then we define the phase of u as $\phi = T\omega$. In this case the maximum gain from u to x is determined as

$$\sup_{u \neq 0} \frac{||x||_2}{||u||_2} = \sigma_1 \left[G\left(e^{j\phi}\right) \right] \tag{11.105}$$

where $\sigma_1(G)$ is the largest singular value of the matrix G. If u can be comprised of an arbitrary mix of frequencies, then the maximum gain from u to x is determined as follows:

$$\begin{aligned} \sup_{\phi} \frac{||x||_2}{||u||_2} &= \sup_{\phi} \sigma_1 \left[G\left(e^{j\phi}\right) \right] \\ &= ||G||_\infty \end{aligned} \tag{11.106}$$

The above equation defines $||G||_\infty$, which is the infinity-norm of the system that has the transfer function $G(z)$.[4]

Now consider Equation (11.92), the cost function that is bounded by the steady-state H$_\infty$ filter:

$$J = \lim_{N \to \infty} \frac{\sum_{k=0}^{N-1} ||z_k - \hat{z}_k||_S^2}{\sum_{k=0}^{N-1} \left(||w_k||_{Q^{-1}}^2 + ||v_k||_{R^{-1}}^2 \right)} \tag{11.107}$$

If Q, R, and S are all equal to identity matrices, then

$$J = \lim_{N \to \infty} \frac{\sum_{k=0}^{N-1} ||z_k - \hat{z}_k||_2^2}{\sum_{k=0}^{N-1} \left(||w_k||_2^2 + ||v_k||_2^2 \right)} \tag{11.108}$$

Since the H$_\infty$ filter makes this scalar less than $1/\theta$ for all w_k and v_k, we can write

$$\begin{aligned} ||G_{\tilde{z}e}||_\infty^2 &= \sup_{\phi} \frac{||z - \hat{z}||_2^2}{||w||_2^2 + ||v||_2^2} \\ &\leq \frac{1}{\theta} \end{aligned} \tag{11.109}$$

where we have defined $\tilde{z} = z - \hat{z}$, $e^T = \begin{bmatrix} w^T & v^T \end{bmatrix}^T$, and $G_{\tilde{z}e}$ is the system that has e as its input and $\tilde{z}$ as its output. We see that the steady-state H$_\infty$ filter bounds the infinity-norm (i.e., the maximum gain) from the combined disturbances w and v to the estimation error $\tilde{z}$, if Q, R, and S are all identity matrices. Further information about the computation of infinity-norms and related issues can be found in [Bur99].

[3]Note that this definition means that many signals have unbounded signal two-norms. The signal two-norm can also be defined as the sum from $k = 0$ to a finite limit $k = N$.

[4]Note that the infinity-norm of a matrix has a definition that is different than the infinity-norm of a system. In general, the expression $||G||_\infty$ could refer either to the matrix infinity-norm or the system infinity-norm. The meaning needs to be inferred from the context unless it is explicitly stated.

■ EXAMPLE 11.3

Consider the system and filter discussed in Example 11.2:

$$\begin{aligned} x_{k+1} &= x_k + w_k \\ y_k &= x_k + v_k \\ \hat{x}_{k+1} &= (1-K)\hat{x}_k + K y_k \end{aligned} \tag{11.110}$$

The estimation error can be computed as

$$\begin{aligned} \tilde{x}_{k+1} &= x_{k+1} - \hat{x}_{k+1} \\ &= (1-K)\tilde{x}_k + w_k - K v_k \end{aligned} \tag{11.111}$$

Taking the z-transform of this equation gives

$$\begin{aligned} z\tilde{X}(z) &= (1-K)\tilde{X}(z) + W(z) - KV(z) \\ \tilde{X}(z) &= \frac{1}{z-1+K}\begin{bmatrix} 1 & -K \end{bmatrix}\begin{bmatrix} W(z) \\ V(z) \end{bmatrix} \\ &= G(z)\begin{bmatrix} W(z) \\ V(z) \end{bmatrix} \end{aligned} \tag{11.112}$$

$G(z)$, the transfer function from w_k and v_k to $\tilde{x}_k$, is a 2×1 matrix. This matrix has one singular value, which is computed as

$$\begin{aligned} \sigma^2(G) &= \lambda_{\max}\left[G(e^{j\phi})G^H(e^{j\phi})\right] \\ &= \frac{1+K^2}{(e^{j\phi}-1+K)(e^{-j\phi}-1+K)} \\ &= \frac{1+K^2}{K^2+2(K-1)(\cos\phi-1)} \end{aligned} \tag{11.113}$$

The supremum of this expression occurs at $\phi = 0$ when $K \leq 1$, so

$$\begin{aligned} \|G\|_\infty^2 &= \sup_\phi \sigma^2\left[G\left(e^{j\phi}\right)\right] \\ &= \frac{1+K^2}{K^2} \end{aligned} \tag{11.114}$$

Recall from Example 11.2 that $\theta = 1/2$ resulted in $K = 1$. In this case, the above expression indicates that $\|G\|_\infty^2 = 2 \leq 1/\theta = 2$. In this case, the infinity-norm bound specified by θ is exact. Also recall from Example 11.2 that $\theta = 1/10$ resulted in $K = 2/3$. In this case, the above expression indicates that $\|G\|_\infty^2 = 13/4 \leq 1/\theta = 10$. In this case, the infinity-norm bound specified by θ is quite conservative.

Note that as K increases, the infinity-norm from the noise to the estimation error decreases. However, the estimator also is unstable for $K > 1$. So even though large K reduces the infinity-norm of the estimator, it gives poor results. In other words, just because the effect of the noise on the estimation error is small does not necessarily prove that the estimator is good. For example, we could set the estimate $\hat{x}_k = \infty$ for all k. In that case, the noise will have zero effect on the estimation error because the estimation error will be infinite regardless of the noise value. However, the estimate will obviously be poor. This example shows the importance of balancing H_∞ performance with other performance criteria.

$\nabla\nabla\nabla$

11.4 THE CONTINUOUS-TIME H_∞ FILTER

The methods of the earlier sections can also be used to derive a continuous-time H_∞ filter, as shown in [Rhe89, Ban91, Ban92]. In this section we consider the continuous-time system

$$\begin{aligned} \dot{x} &= Ax + Bu + w \\ y &= Cx + v \\ z &= Lx \end{aligned} \tag{11.115}$$

where L is a user-defined matrix and z is the vector that we want to estimate. Our estimate of z is denoted $\hat{z}$, and our estimate of the state at time 0 is denoted $\hat{x}(0)$. The vectors w and v are disturbances with unknown statistics; they may not even be zero-mean. In the game theory approach to H_∞ filtering we define the following cost function:

$$J_1 = \frac{\int_0^T ||z - \hat{z}||_S^2 \, dt}{||x(0) - \hat{x}(0)||_{P_0^{-1}}^2 + \int_0^T \left(||w||_{Q^{-1}}^2 + ||v||_{R^{-1}}^2 \right) dt} \tag{11.116}$$

P_0, Q, R, and S are positive definite matrices chosen by the engineer based on the specific problem. Our goal is to find an estimator such that

$$J_1 < \frac{1}{\theta} \tag{11.117}$$

The estimator that solves this problem is given by

$$\begin{aligned} P(0) &= P_0 \\ \dot{P} &= AP + PA^T + Q - KCP + \theta P L^T S L P \\ K &= PC^T R^{-1} \\ \dot{\hat{x}} &= A\hat{x} + Bu + K(y - C\hat{x}) \\ \hat{z} &= L\hat{x} \end{aligned} \tag{11.118}$$

These equations are identical to the continuous-time Kalman filter equations (see Section 8.2) except for the θ term in the $\dot{P}$ equation. The inclusion of the θ term in the $\dot{P}$ equation tends to increase P, which tends to increase the gain K, which tends to make the estimator more responsive to measurements than the Kalman filter. This is a way of robustifying the filter to uncertainty in the system model. The estimator given above solves the H_∞ estimation problem if and only if $P(t)$ remains positive definite for all $t \in [0, T]$. As with the discrete-time filter, we can also obtain a steady-state continuous-time H_∞ filter. To do this we let $\dot{P} = 0$ so that the differential Riccati equation above reduces to an algebraic Riccati equation.

■ EXAMPLE 11.4

Consider the scalar continuous-time system

$$\begin{aligned} \dot{x} &= x + w \\ y &= x + v \\ z &= x \end{aligned} \tag{11.119}$$

We see that $A = C = L = 1$. Further suppose that $Q = R = S = 1$ in the cost function of Equation (11.116). Then the differential Riccati equation for the H$_\infty$ filter is

$$\begin{aligned} \dot{P} &= AP + PA^T + Q - PC^TR^{-1}CP + \theta PL^TSLP \\ &= 2P + 1 + (\theta - 1)P^2 \end{aligned} \tag{11.120}$$

This can be solved numerically or analytically as a function of time for a given θ to give P, and then the H$_\infty$ gain $K = PC^TR^{-1} = P$ can be obtained. We can also set $\dot{P} = 0$ in Equation (11.120) to obtain the steady-state solution for P. This gives

$$(\theta - 1)P^2 + 2P + 1 = 0 \tag{11.121}$$

As mentioned above, the solution to this quadratic equation must be positive definite in order for it to solve the H$_\infty$ estimation problem. For this scalar equation, positive definite simply means positive. The equation has a positive solution for $\theta < 1$, in which case the steady-state solution is given by

$$P = \frac{-1 - \sqrt{2 - \theta}}{\theta - 1} \tag{11.122}$$

Suppose we choose $\theta = 7/16$. In this case, the analytic solution for the time-varying P can be obtained from Equation (11.120) as

$$\begin{aligned} P(t) &= \frac{4 + 160ce^{5t/2}}{-9 + 40ce^{5t/2}} \\ c &= \frac{9P(0) + 4}{40P(0) - 160} \end{aligned} \tag{11.123}$$

From this analytic expression for $P(t)$ we can see that

$$\lim_{t \to \infty} P(t) = 4 \tag{11.124}$$

Alternatively, we can substitute $\theta = 7/16$ in Equation (11.122) to obtain $P = 4$. Figure 11.4 shows P as a function of time when $P(0) = 1$. Note that in this example, since $C = R = 1$, the H$_\infty$ gain K is equal to P.

Figure 11.5 shows the state estimation errors for the time-varying H$_\infty$ filter and the steady-state H$_\infty$ filter. In these simulations, the disturbances w and v were both normally distributed white noise sequences with standard deviations equal to 10. w had a mean of zero, and v had a mean of 10. Both simulations were run with identical disturbance time histories. It can be seen that the performance of the two filters is very similar. There are some differences between the two plots at small values of time before the

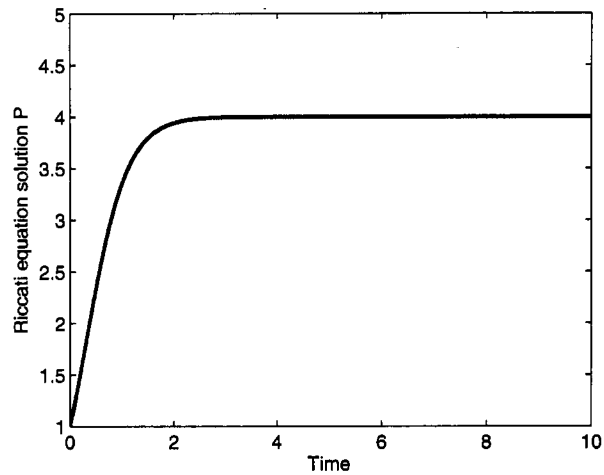

Figure 11.4 Example 11.4 H_∞ Riccati equation solution as a function of time.

time-varying Riccati solution has converged to steady state (note that the time-varying filter performs better during the initial transient). But after the Riccati solution gets close to steady state (after about $t = 1$) the performance of the two filters is nearly identical. This illustrates the possibility of saving a lot of computational effort by using a steady-state filter while giving up only an incremental amount of performance.

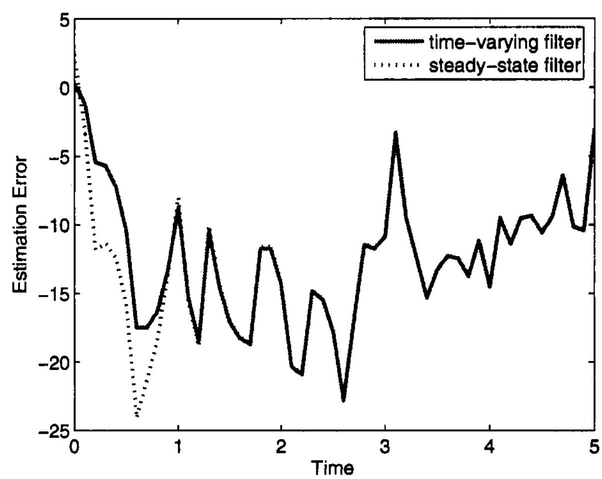

Figure 11.5 Example 11.4 time-varying and steady-state H_∞ filter performance when the measurement noise is zero-mean.

If we use the performance bound $\theta = 0$ in this example then we obtain the Kalman filter. The steady-state Riccati equation solution from Equation (11.120) is $(1+\sqrt{2})$ when $\theta = 0$, so the steady-state Kalman gain $K \approx 2.4$, which is less than the steady-state H_∞ gain $K = 4$ that we obtained for $\theta = 7/16$. From Equation (11.118) we see that this will make the Kalman filter less responsive to measurements than the H_∞ filter, but the Kalman filter should provide optimal RMS error performance. Indeed, if we run the time-varying Kalman filter ($\theta = 0$) then the two-norm of the estimation error turns out to be 26.5. If we run the time-varying H_∞ filter ($\theta = 7/16$) then the two-norm of the estimation error increases to 30.0.

However, the Kalman filter assumes that the system model is known exactly, the process and measurement noises are zero-mean and uncorrelated, and the noise statistics are known exactly. If we change the simulation so the measurement noise has a mean of 10 then the H_∞ filter works better than the Kalman filter. Figure 11.6 shows the estimation error of the two filters in this case. The two-norm of the estimation error is 112.8 for the Kalman filter but only 94.2 for the H_∞ filter.

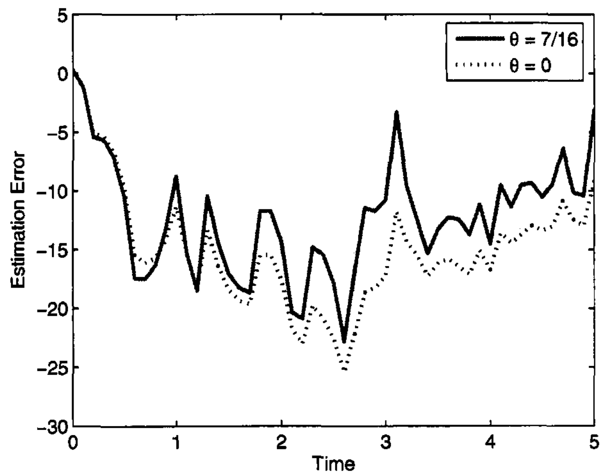

Figure 11.6 Example 11.4 time-varying Kalman and H_∞ filter performance when the measurement noise is not zero-mean.

$\nabla\nabla\nabla$

As with the discrete-time steady-state filter, if Q, R, and S are all identity matrices, the continuous-time steady-state filter bounds the maximum gain from the noise to the estimation error:

$$\begin{aligned} \|G_{\tilde{z}e}\|_\infty^2 &= \sup_\omega \frac{\|z-\hat{z}\|_2^2}{\|w\|_2^2 + \|v\|_2^2} \\ &\leq \frac{1}{\theta} \end{aligned} \qquad (11.125)$$

where ω is the frequency of the noise, and we have defined $\tilde{z} = z - \hat{z}$, $e^T = \begin{bmatrix} w^T & v^T \end{bmatrix}^T$, and $G_{\tilde{z}e}$ is the system that has e as its input and $\tilde{z}$ as its output. The continuous-time infinity-norm of the system $G_{\tilde{z}e}$ is defined as follows:

$$\begin{aligned} \|G_{\tilde{z}e}\|_\infty &= \sup_\omega \frac{\|\tilde{z}\|_2}{\|e\|_2} \\ &= \sup_\omega \sigma_1 \left[G_{\tilde{z}e}(j\omega)\right] \end{aligned} \tag{11.126}$$

where $G_{\tilde{z}e}(s)$ is the transfer function from e to $\tilde{z}$.

11.5 TRANSFER FUNCTION APPROACHES

It should be emphasized that other formulations to H_∞ filtering have been proposed. For instance, Isaac Yaesh and Uri Shaked [Yae91] consider the following time-invariant system:

$$\begin{aligned} x_{k+1} &= Fx_k + w_k \\ x_0 &= 0 \\ y_k &= Hx_k + v_k \\ z_k &= Lx_k \end{aligned} \tag{11.127}$$

where w_k and v_k are uncorrelated process and measurement noise, y_k is the measurement, and z_k is the vector to be estimated. Define the estimation error as

$$\tilde{z}_k = z_k - \hat{z}_k \tag{11.128}$$

Define an augmented disturbance vector as

$$e_k = \begin{bmatrix} w_k \\ v_k \end{bmatrix} \tag{11.129}$$

The goal is to find a steady-state estimator such that the infinity-norm of the transfer function from the augmented disturbance vector e to the estimation error $\tilde{z}$ is less than some user specified bound:

$$\|G_{\tilde{z}e}\|_\infty^2 < \frac{1}{\theta} \tag{11.130}$$

The steady-state *a priori* filter that solves this problem is given as

$$\begin{aligned} P &= I + FPF^T - FPH^T(I + HPH^T)^{-1}HPF^T + \\ &\quad\; PL(I/\theta + LPL^T)^{-1}LP \\ K &= FPH^T(I + HPH^T)^{-1} \\ \hat{x}_{k+1} &= F\hat{x}_k + K(y_k - H\hat{x}_k) \end{aligned} \tag{11.131}$$

These equations solve the H_∞ estimation problem if and only if P is positive definite.

The steady-state *a posteriori* filter that solves this problem is given as

$$\begin{aligned} \Sigma^{-1} &= \tilde{P}^{-1} - \theta L^T L + H^T H \\ \tilde{P} &= F\tilde{P}(H^T H\tilde{P} - \theta L^T L\tilde{P} + I)^{-1}F^T + I \\ \tilde{K} &= (I + \theta L^T L)^{-1}\Sigma H^T \\ &= \tilde{P}(I + H^T H\tilde{P})^{-1}H^T \\ \hat{x}_{k+1} &= F\hat{x}_k + \tilde{K}(y_{k+1} - HF\hat{x}_k) \end{aligned} \tag{11.132}$$

Again, these equations solve the H_∞ estimation problem if and only if $\tilde{P}$ is positive definite.

Interestingly, the P matrix in the *a priori* filter of Equation (11.131) is related to the $\tilde{P}$ matrix in the *a posteriori* filter of Equation (11.132) by the following equation:

$$P^{-1} = \tilde{P}^{-1} - \theta L^T L \tag{11.133}$$

In general, the Riccati equations in these filters can be difficult to solve. However, the solution can be obtained by the eigenvector method shown in [Yae91]. (This is similar to the Hamiltonian approach to steady-state Kalman filtering described in Section 7.3.3.) Define the $2n \times 2n$ matrix

$$\mathcal{H} = \begin{bmatrix} F^T + H^T H F^{-1} & \theta F^T L^T L - H^T H F^{-1}(I - \theta L^T L) \\ -F^{-1} & F^{-1}(I - \theta L^T L) \end{bmatrix} \tag{11.134}$$

Note that F^{-1} should always exist if it comes from a real system, because F comes from a matrix exponential that is always invertible (see Sections 1.2 and 1.4). Compute the n eigenvectors of $\mathcal{H}$ that correspond to the eigenvalues outside the unit circle. Denote those eigenvectors as ξ_i ($i = 1, \ldots, n$). Form the $2n \times n$ matrix

$$\begin{bmatrix} \xi_1 & \cdots & \xi_n \end{bmatrix} = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \tag{11.135}$$

where X_1 and X_2 are $n \times n$ matrices. The P matrix used in the *a priori* H_∞ filter can be computed as

$$P = X_2 X_1^{-1} \tag{11.136}$$

For the *a posteriori* filter, define the $2n \times 2n$ matrix

$$\tilde{\mathcal{H}} = \begin{bmatrix} F^{-T} & F^{-T}(H^T H - \theta L^T L) \\ F^{-T} & F + F^{-T}(H^T H - \theta L^T L) \end{bmatrix} \tag{11.137}$$

Compute the n eigenvectors of $\tilde{\mathcal{H}}$ that correspond to the eigenvalues outside the unit circle. Denote those eigenvectors as $\tilde{\xi}_i$ ($i = 1, \ldots, n$). Form the $2n \times n$ matrix

$$\begin{bmatrix} \tilde{\xi}_1 & \cdots & \tilde{\xi}_n \end{bmatrix} = \begin{bmatrix} \tilde{X}_1 \\ \tilde{X}_2 \end{bmatrix} \tag{11.138}$$

where $\tilde{X}_1$ and $\tilde{X}_2$ are $n \times n$ matrices. The $\tilde{P}$ matrix used in the *a posteriori* H_∞ filter can be computed as

$$\tilde{P} = \tilde{X}_2 \tilde{X}_1^{-1} \tag{11.139}$$

The eigenvector method for the Riccati equation solutions works because $\mathcal{H}$ and $\tilde{\mathcal{H}}$ are symplectic matrices (see Section 7.3.3 and Problem 11.9). This assumes that F is nonsingular and that $\mathcal{H}$ and $\tilde{\mathcal{H}}$ do not have any eigenvalues on the unit circle. If these assumptions are violated, then the problem becomes more complicated [Yae91]. A method similar to this for continuous-time systems is developed in [Nag91].

It is important to be aware that the P and $\tilde{P}$ solutions given by Equations (11.136) and (11.139) only give one solution each to Equations (11.131) and (11.132). Equations (11.136) and (11.139) may give solutions to Equations (11.131) and (11.132) that are not positive definite and therefore do not satisfy the H_∞ filtering problem. However, that does not prove that the H_∞ filtering solution does not exist (see Problem 11.13).

■ **EXAMPLE 11.5**

We will revisit Example 11.2, but assume that the initial state is 0:

$$\begin{aligned} x_{k+1} &= x_k + w_k \\ x_0 &= 0 \\ y_k &= x_k + v_k \\ z_k &= x_k \end{aligned} \tag{11.140}$$

From Equation (11.131) we can find the *a priori* steady-state filter that bounds the infinity-norm of the transfer function from e to $\tilde{z}$ by $1/\sqrt{\theta}$. (Recall that $e_k = \begin{bmatrix} w_k & v_k \end{bmatrix}^T$.) The algebraic Riccati equation associated with this problem is given by

$$\begin{aligned} P &= 1 + P - P(1+P)^{-1}P + P(1/\theta + P)^{-1}P \\ &= 1 + P - \frac{P^2}{1+P} + \frac{P^2}{1/\theta + P} \end{aligned} \tag{11.141}$$

Solving the above for P we obtain

$$P = \frac{-\theta - 1 \pm \sqrt{\theta^2 - 6\theta + 5}}{2(2\theta - 1)} \tag{11.142}$$

In order for the solution of this equation to solve the H_∞ filtering problem, we must have $P > 0$. The only solution for which $P > 0$ is when $0 \le \theta < 1/2$ and when we use the negative sign in the above solution.[5] If we choose $\theta = 1/10$ then $P = 2$. The gain of the *a priori* filter is then computed from Equation (11.131) as

$$\begin{aligned} K &= P(1+P)^{-1} \\ &= 2/3 \end{aligned} \tag{11.143}$$

Note that the P value that is obtained for $\theta = 1/10$ does not match Example 11.2, but K does match. The H_∞ filter equation is computed from Equation (11.131) as

$$\begin{aligned} \hat{x}_{k+1} &= \hat{x}_k + K(y_k - \hat{x}_k) \\ &= \hat{x}_k + (2/3)(y_k - \hat{x}_k) \end{aligned} \tag{11.144}$$

∇∇∇

11.6 SUMMARY

In this chapter, we have presented a couple of different approaches to H_∞ estimation, also called minimax estimation. H_∞ filtering minimizes the worst-case

[5]Note that Example 11.2 showed that this problem has a solution for $0 \le \theta < 1$, which indicates that the game theory approach to H_∞ filtering may be more general than the transfer function approach.

estimation error and is thus more robust than Kalman filtering, which minimizes the RMS estimation error. H_∞ filtering has sometimes been criticized for being too pessimistic in its assumption about the noise processes that impinge on the system and measurement equations. After all, H_∞ estimation assumes that the noise is worst case, thus attributing a degree of perversity to the noise that intuitively seems unrealistic. This has led to mixed Kalman/H_∞ estimation techniques, which we will discuss in Chapter 12.

Research in H_∞ estimation began in the 1980s. During that decade, some work was directed toward the design of minimax state estimators for systems corrupted by random noise whose covariances were within known bounds [Poo81, Dar84, Ver84]. This was a first step toward H_∞ filtering, although it still assumed that the noise was characterized by statistical measurements. The earliest work that could pass for what we now call H_∞ filtering was probably published by Mike Grimble [Gri88]. However, unlike the presentation in this chapter, he used a frequency domain approach. He designed a state estimator such that the frequency response from the noise to the estimation error had a user-defined upper bound.

Some early tutorials on H_∞ filtering can be found in [Gri91b, Sha92]. A polynomial systems approach to H_∞ filtering is presented in [Gri90]. Nonlinear H_∞ filtering is discussed in [Rei99], where a stable state estimator with a bounded infinity-norm is derived. System identification using H_∞ methods is discussed in [Sto94, Tse94, Bai95, Did95, Pan96].

The effectiveness of the H_∞ filter can be highly sensitive to the weighting functions [e.g., S_k, P_0, Q_k, and R_k in Equation (11.36), and θ in the performance bound]. This sometimes makes H_∞ filter design more sensitive than Kalman filter design (which is ironic, considering the higher degree of robustness in H_∞ filtering). The advantages of H_∞ estimation over Kalman filtering can be summarized as follows.

1. H_∞ filtering provides a rigorous method for dealing with systems that have model uncertainty.

2. H_∞ filtering provides a natural way to limit the frequency response of the estimator.

The disadvantages of H_∞ filtering compared to Kalman filtering can be summarized as follows.

1. The filter performance is more sensitive to the design parameters.

2. The theory underlying H_∞ filtering is more abstract and complicated.

The types of applications where H_∞ filtering may be preferred over Kalman filtering could include the following.

1. Systems in which stability margins must be guaranteed, or worst-case estimation performance is a primary consideration (rather than RMS estimation performance) [Sim96].

2. Systems in which the model changes unpredictably, and identification and gain scheduling are too complex or time-consuming.

3. Systems in which the model is not well known.

Work by Babak Hassibi, Ali Sayed, and Thomas Kailath involves the solution of state estimation problems within the context of Krein spaces (as opposed to the usual Hilbert space approach). This provides a general framework for both Kalman and H_∞ filtering (along with other types of filtering), and is discussed in some of their papers [Has96a, Has96b] and books [Has99, Kai00].

PROBLEMS

Written exercises

11.1 Show that $(I + A)^{-1}A = A(I + A)^{-1}$.

11.2 Consider a scalar system with $F = H = 1$ and with process noise and measurement noise variances Q and R. Suppose a state estimator of the form

$$\hat{x}_{k+1}^- = \hat{x}_k^- + K(y_k - \hat{x}_k^-)$$

is used to estimate the state, where K is a general estimator gain.

a) Find the optimal gain K if $R = 2Q$. Call this gain K_0. What is the resulting steady-state *a priori* estimation-error variance?

b) Suppose that $R = 0$. What is the optimal steady-state *a priori* estimation-error variance? What is the (suboptimal) steady-state *a priori* estimation-error variance if K_0 is used in the estimator? Repeat for $R = Q$ and $R = 5Q$.

11.3 Consider a scalar system with $F = H = 1$ and with process noise and measurement noise variances Q and $R = 2Q$. A Kalman filter is designed to estimate the state, but (unknown to the engineer) the process noise has a mean of $\bar{w}$.

a) What is the steady-state value of the mean of the *a priori* estimation error?

b) Introduce a new state-vector element that is equal to $\bar{w}$. Augment the new state-vector element to the original system so that a Kalman filter can be used to estimate both the original state element and the new state element. Find an analytical solution to the steady-state *a priori* estimation-error covariance for the augmented system.

11.4 Suppose that a Kalman filter is designed to estimate the state of a scalar system. The assumed system is given as

$$\begin{aligned} x_{k+1} &= Fx_k + w_k \\ y_k &= Hx_k + v_k \end{aligned}$$

where $w_k \sim (0, Q)$ and $v_k \sim (0, R)$ are uncorrelated zero-mean white noise processes. The actual system matrix is $\tilde{F} = F + \Delta F$.

a) Under what conditions is the mean of the steady-state value of the *a priori* state estimation error equal to zero?

b) What is the steady-state value of the *a priori* estimation-error variance P? How much larger is P because of the modeling error ΔF?

11.5 Find the stationary point of $(x_1^2 + x_1x_2 + x_2x_3)$ subject to the constraint $(x_1 + x_2 = 4)$ [Moo00].

11.6 Maximize $(14x - x^2 + 6y - y^2 + 7)$ subject to the constraints $(x + y \leq 2)$ and $(x + 2y \leq 3)$ [Lue84].

11.7 Consider the system

$$\begin{aligned} x_k &= \frac{1}{2}x_{k-1} + w_{k-1} \\ y_k &= x_k + v_k \end{aligned}$$

Note that this is the system model for the radiation system described in Problem 5.1.

a) Find the steady-state value of P_k for the H_∞ filter, using a variable θ and $L = R = Q = S = 1$.

b) Find the bound on θ such that the steady-state H_∞ filter exists.

11.8 Suppose that you use a continuous-time H_∞ filter to estimate a constant on the basis of noisy measurements. The measurement noise is zero-mean and white with a covariance of R. Find the H_∞ estimator gain as a function of P_0, R, θ, and time. What is the limit of the estimator gain as $t \to \infty$? What is the maximum value of θ such that the H_∞ estimation problem has a solution? How does the value of θ influence the estimator gain?

11.9 Prove that $\mathcal{H}$ and $\tilde{\mathcal{H}}$ in Equations (11.134) and (11.137) are symplectic.

11.10 Prove that the solution of the *a posteriori* H_∞ Riccati equation given in Equation (11.132) with $\theta = 0$ is equivalent to the solution of the steady-state *a priori* Kalman filter Riccati equation with $R = I$ and $Q = I$.

11.11 Prove that Σ in Equation (11.132) with $\theta = 0$ is equivalent to the solution of the steady-state *a posteriori* Kalman filter Riccati equation with $R = I$ and $Q = I$.

11.12 Find the *a posteriori* steady-state H_∞ filter for Example 11.5 when $\theta = 1/10$. Verify that the *a priori* and *a posteriori* Riccati equation solutions satisfy Equation (11.133).

11.13 Find all possible solutions P to the *a priori* H_∞ filtering problem for Example 11.5 when $\theta = 0$. Next use Equation (11.139) to find the P solution. Repeat for $\theta = 1/10$. [Note that Equation (11.139) gives a negative solution for P and therefore cannot be used.]

Computer exercises

11.14 Generate the time-varying solution to P_k for Problem 11.7 with $P_0 = 1$. What is the largest value of θ for which Equation (11.90) will be satisfied for all k up to and including $k = 20$? Answer to the nearest 0.01. Repeat for $k = 10$, $k = 5$, and $k = 1$.

11.15 Consider the vehicle navigation problem described in Example 7.12. Design a Kalman filter and an H_∞ filter to estimate the states of the system. Use the following parameters.

$$\begin{aligned}
T &= 3 \\
u_k &= 1 \\
Q &= \text{diag}(4, 4, 1, 1) \\
R &= \text{diag}(900, 900) \\
\text{heading angle} &= 0.9\pi \\
x(0) &= \hat{x}(0) = \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix}^T
\end{aligned}$$

Simulate the system and the filters for 300 seconds. In the H_∞ filter use $S = L = I$ and $\theta = 0.0005$.

a) Plot the position estimation errors for the Kalman and H_∞ filters. What are the RMS position estimation errors for the two filters?

b) Now suppose that unknown to the filter designer, $u_k = 2$. Plot the position estimation errors for the Kalman and H_∞ filters. What are the RMS position estimation errors for the two filters?

c) What are the closed loop estimator eigenvalues for the Kalman and H_∞ filters? Do their relative magnitudes agree with your intuition?

d) Use MATLAB's DARE function to find the largest θ for which a steady-state solution exists to the H_∞ DARE. Answer to the nearest 0.0001. How well does the H_∞ filter work for this value of θ? What are the closed-loop eigenvalues of the H_∞ filter for this value of θ?