2.3 PROPERTIES OF KALMAN FILTER

We will now give some physical feel for why the Kalman filter is optimum. Let us go back to our discussion in Section 1.2. Recall that for our two-state g–h tracking we have at time n two estimates of the target position. The first is y_n, based on the measurement made at time n (see Figure 1.2-3). The second is the prediction $x^*_{n,n-1}$, based on past measurements. The Kalman filter combines these two estimates to provide a filtered estimate $x^*_{n,n}$ for the position of the target at time n. The Kalman filter combines these two estimates so as to obtain an estimate that has a minimum variance, that is, the best accuracy. The estimate $x^*_{n,n}$ will have a minimum variance if it is given by [5–7]

$$x^*_{n,n} = \left[\frac{x^*_{n,n-1}}{\mathrm{VAR}(x^*_{n,n-1})} + \frac{y_n}{\mathrm{VAR}(Y_n)}\right] \frac{1}{1/\mathrm{VAR}(x^*_{n,n-1}) + 1/\mathrm{VAR}(y_n)} \qquad (2.3\text{-}1)$$

That (2.3-1) provides a good combined estimate can be seen by examining some special cases. First consider the case where y_n and $x^*_{n,n-1}$ have equal accuracy. To make this example closer to what we are familiar with, we use the example we used before; that is, we assume that y_n and $x^*_{n,n-1}$ represent two independent estimates of your weight obtained from two scales having equal accuracy (the example of Section 1.2.1). If one scale gives a weight estimate of 110 lb and the other 120 lb, what would you use for the best combined-weight estimate? You would take the average of the two weight estimates to obtain 115 lb. This is just what (2.3-1) does. If the variances of the two estimates are equal (say to σ^2), then (2.3-1) becomes

$$x^*_{n,n} = \left(\frac{x^*_{n,n-1}}{\sigma^2} + \frac{y_n}{\sigma^2}\right) \frac{1}{1/\sigma^2 + 1/\sigma^2} = \frac{x^*_{n,n-1} + y_n}{2} \qquad (2.3\text{-}2)$$

Thus in Figure 1.2-3 the combined estimate $x^*_{n,n}$ is placed exactly in the middle between the two estimates y_n and $x^*_{n,n-1}$.

Now consider the case where $x^*_{n,n-1}$ is much more accurate than the estimate y_n. For this case $\mathrm{VAR}(x^*_{n,n-1}) \ll \mathrm{VAR}(y_n)$ or equivalently $1/\mathrm{VAR}(x^*_{n,n-1}) \gg 1/\mathrm{VAR}(y_n)$. As a result, (2.3-1) can be approximated by

$$\begin{aligned} x^*_{n,n} &= \left[\frac{x^*_{n,n-1}}{\mathrm{VAR}(x^*_{n,n-1})} + 0\right] \frac{1}{1/\mathrm{VAR}(x^*_{n,n-1}) + 0} \\ &\doteq x^*_{n,n-1} \end{aligned} \qquad (2.3\text{-}3)$$

Thus the estimate $x^*_{n,n}$ is approximately equal to $x^*_{n,n-1}$, as it should be because the accuracy of $x^*_{n,n-1}$ is much better than that of y_n. For this case, in Figure 1.2-3 the combined estimate x^*_n is placed very close to the estimate $x^*_{n,n-1}$ (equal to it).

Equation (2.3-1) can be put in the form of one of the Kalman g–h tracking filters. Specifically, (2.3-1) can be rewritten as

$$x^*_{n,n} = x^*_{n,n-1} + \frac{\mathrm{VAR}(x^*_{n,n})}{\mathrm{VAR}(y_n)}(y_n - x^*_{n,n-1}) \qquad (2.3\text{-}4)$$

This in turn can be rewritten as

$$x^*_{n,n} = x^*_{n,n-1} + g_n(y_n - x^*_{n,n-1}) \qquad (2.3\text{-}5)$$

This is the same form as (1.2-7) [and also (1.2-8b)] for the g–h tracking filter. Comparing (2.3-5) with (1.2-7) gives us the expression for the constant g_n. Specifically

$$g_n = \frac{\mathrm{VAR}(x^*_{n,n})}{\mathrm{VAR}(y_n)} \qquad (2.3\text{-}6)$$

Thus we have derived one of the Kalman tracking equations, the one for updating the target position. The equation for the tracking-filter parameter h_n is given by

$$h_n = \frac{\mathrm{COV}(x^*_{n,n}\dot{x}^*_{n,n})}{\mathrm{VAR}(y_n)} \qquad (2.3\text{-}7)$$

A derivation for (2.3-7) is given for the more general case in Section 2.6.

2.4 KALMAN FILTER IN MATRIX NOTATION

In this section we shall rework the Kalman filter in matrix notation. The Kalman filter in matrix notation looks more impressive. You can impress your friends when you give it in matrix form! Actually there are very good reasons for putting it in matrix form. First, it is often put in matrix notation in the literature, and hence it is essential to know it in this form in order to recognize it. Second, and more importantly, as shall be shown later, in the matrix notation form the Kalman filter applies to a more general case than the one-dimensional case given by (2.1-3) or (1.2-11).

First we will put the system dynamics model given by (1.1-1) into matrix notation. Then we will put the random system dynamics model of (2.1-1) into matrix notation. Equation (1.1-1) in matrix notation is

$$X_{n+1} = \Phi X_n \qquad (2.4\text{-}1)$$

where

$$X_n = \begin{bmatrix} x_n \\ \dot{x}_n \end{bmatrix} = \text{state vector} \qquad (2.4\text{-}1a)$$

and

$$\Phi = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}$$

$$= \text{state transition matrix for constant-velocity trajectory [5, 43]} \qquad (2.4\text{-}1b)$$

To show that (2.4-1) is identical to (1.1-1), we just substitute (2.4-1a) and (2.4-1b) into (2.4-1) to obtain

$$\begin{bmatrix} x_{n+1} \\ \dot{x}_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_n \\ \dot{x}_n \end{bmatrix} \qquad (2.4\text{-}1c)$$

which on carrying out the matrix multiplication yields

$$\begin{bmatrix} x_{n+1} \\ \dot{x}_{n+1} \end{bmatrix} = \begin{bmatrix} x_n + T\dot{x}_n \\ \dot{x}_n \end{bmatrix} \qquad (2.4\text{-}1d)$$

which we see is identical to (1.1-1).

As indicated in (2.4-1a), X_n is the target trajectory state vector. This state vector is represented by a column matrix. As pointed out in Section 1.4, it consists of the quantities being tracked. For the filter under consideration these quantities are the target position and velocity at time n. It is called a two-state vector because it consists of two target states: target position and target velocity. Here, Φ is the state transition matrix. This matrix transitions the state vector X_n at time n to the state vector X_{n+1} at time $n+1$ a period T later.

It is now a simple matter to give the random system dynamics model represented by (2.1-1) in matrix form. Specifically, it becomes

$$X_{n+1} = \Phi X_n + U_n \qquad (2.4\text{-}2)$$

where

$$U_n = \begin{bmatrix} 0 \\ u_n \end{bmatrix}$$

$$= \text{dynamic model driving noise vector} \qquad (2.4\text{-}2a)$$

To show that (2.4-2) is identical to (2.1-1), we now substitute (2.4-1a), (2.4-1b), and (2.4-2a) into (2.4-2) to obtain directly from (2.4-1d)

$$\begin{bmatrix} x_{n+1} \\ \dot{x}_{n+1} \end{bmatrix} = \begin{bmatrix} x_n + T\dot{x}_n \\ \dot{x}_n \end{bmatrix} + \begin{bmatrix} 0 \\ u_n \end{bmatrix} \qquad (2.4\text{-}2b)$$

which on adding the corresponding terms of the matrices on the right-hand side of (2.4-2b) yields

$$\begin{bmatrix} x_{n+1} \\ \dot{x}_{n+1} \end{bmatrix} = \begin{bmatrix} x_n + T\dot{x}_n \\ \dot{x}_n + u_n \end{bmatrix} \qquad (2.4\text{-}2c)$$

which is identical to (2.1-1), as we desired to show.

We now put the trivial measurements equation given by (1.2-17) into matrix form. It is given by

$$Y_n = MX_n + N_n \qquad (2.4\text{-}3)$$

where

$$M = [1 \quad 0] = \text{observation matrix} \qquad (2.4\text{-}3a)$$

$$N_n = [\nu_n] = \text{observation error} \qquad (2.4\text{-}3b)$$

$$Y_n = [y_n] = \text{measurement matrix} \qquad (2.4\text{-}3c)$$

Equation (2.4-3) is called the observation system equation. This is because it relates the quantities being estimated to the parameter being observed, which, as pointed out in Section 1.5, are not necessarily the same. In this example, the parameters x_n and $\dot{x}_n$ (target range and velocity) are being estimated (tracked) while only target range is observed. In the way of another example, one could track a target in rectangular coordinates (x, y, z) and make measurements on the target in spherical coordinates (R, θ, ϕ). In this case the observation matrix M would transform from the rectangular coordinates being used by the tracking filter to the spherical coordinates in which the radar makes its measurements.

To show that (2.4-3) is given by (1.2-17), we substitute (2.4-3a) to (2.4-3c), into (2.4-3) to obtain

$$[y_n] = [1 \quad 0] \begin{bmatrix} x_n \\ \dot{x}_n \end{bmatrix} + [\nu_n] \qquad (2.4\text{-}3d)$$

which on carrying out the multiplication becomes

$$[y_n] = [x_n] + [\nu_n] \qquad (2.4\text{-}3e)$$

Finally, carrying out the addition yields

$$[y_n] = [x_n + \nu_n] \qquad (2.4\text{-}3f)$$

which is identical to (1.2-17).

Rather than put the g–h tracking equations as given by (1.2-11) in matrix form, we will put (1.2-8) and (1.2-10) into matrix form. These were the equations that were combined to obtain (1.2-11). Putting (1.2-10) into matrix form yields

$$X^*_{n+1,n} = \Phi X^*_{n,n} \tag{2.4-4a}$$

where

$$X^*_{n,n} = \begin{bmatrix} x^*_{n,n} \\ \dot{x}^*_{n,n} \end{bmatrix} \tag{2.4-4b}$$

$$X^*_{n+1,n} = \begin{bmatrix} x^*_{n+1,n} \\ \dot{x}^*_{n+1,n} \end{bmatrix} \tag{2.4-4c}$$

This is called the prediction equation because it predicts the position and velocity of the target at time $n+1$ based on the position and velocity of the target at time n, the predicted position and velocity being given by the state vector of (2.4-4c). Putting (1.2-8) into matrix form yields

$$X^*_{n,n} = X^*_{n,n-1} + H_n(Y_n - MX^*_{n,n-1}) \tag{2.4-4d}$$

Equation (2.4-4d) is called the Kalman filtering equation because it provides the updated estimate of the present position and velocity of the target.

The matrix H_n is a matrix giving the tracking-filter constants g_n and h_n. It is given by

$$H_n = \begin{bmatrix} g_n \\ \dfrac{h_n}{T} \end{bmatrix} \tag{2.4-5}$$

for the two-state g–h or Kalman filter equations of (1.2-10). This form does not however tell us how to obtain g_n and h_n. The following form (which we shall derive shortly) does:

$$H_n = S^*_{n,n-1} M^T \left[R_n + MS^*_{n,n-1} M^T \right]^{-1} \tag{2.4-4e}$$

where

$$S^*_{n,n-1} = \Phi S^*_{n-1,n-1} \Phi^T + Q_n \quad \text{(predictor equation)} \tag{2.4-4f}$$

and

$$Q_n = \text{COV}[U_n] = E[U_n U_n^T] \quad \text{(dynamic model noise covariance)} \tag{2.4-4g}$$

$$S^*_{n,n-1} = \text{COV}(X^*_{n,n-1}) = E[X^*_{n,n-1} X^{*T}_{n,n-1}] \tag{2.4-4h}$$

$$R_n = \text{COV}(N_n) = E[N_n N_n^T] \quad \text{(observation noise covariance)} \tag{2.4-4i}$$

$$\begin{aligned} S^*_{n-1,n-1} &= \text{COV}(X^*_{n-1,n-1}) \\ &= [\text{I} - H_{n-1}M] S_{n-1,n-2} \quad \text{(corrector equation)} \end{aligned} \tag{2.4-4j}$$

As was the case for (1.4-1), covariances in (2.4-4g) and (2.4-4i) apply as long as the entries of the column matrices U_n and N_n have zero mean. Otherwise U_n and N_n have to be replaced by $U_n - E[U_n]$ and $N_n - E[N_n]$, respectively. These equations at first look formidable, but as we shall see, they are not that bad. We shall go through them step by step.

Physically, the matrix $S^*_{n,n-1}$ is an estimate of our accuracy in prediciting the target position and velocity at time n based on the measurements made at time $n-1$ and before. Here, $S^*_{n,n-1}$ is the covariance matrix of the state vector $X^*_{n,n-1}$. To get a better feel for $S^*_{n,n-1}$, let us write it out for our two-state $X^*_{n,n-1}$. From (1.4-1) and (2.4-4c) it follows that

$$\begin{aligned} \text{COV}\, X^*_{n,n-1} &= \overline{X^*_{n,n-1} X^{*T}_{n,n-1}} \\ &= \overline{\begin{bmatrix} x^*_{n,n-1} \\ \dot{x}^*_{n,n-1} \end{bmatrix} \begin{bmatrix} x^*_{n,n-1} & \dot{x}^*_{n,n-1} \end{bmatrix}} = \begin{bmatrix} \overline{x^*_{n,n-1} x^*_{n,n-1}} & \overline{x^*_{n,n-1} \dot{x}^*_{n,n-1}} \\ \overline{\dot{x}^*_{n,n-1} x^*_{n,n-1}} & \overline{\dot{x}^*_{n,n-1} \dot{x}^*_{n,n-1}} \end{bmatrix} \\ &= \begin{bmatrix} \overline{x^{*2}_{n,n-1}} & \overline{x^*_{n,n-1} \dot{x}^*_{n,n-1}} \\ \overline{\dot{x}^*_{n,n-1} x^*_{n,n-1}} & \overline{\dot{x}^{*2}_{n,n-1}} \end{bmatrix} \\ &= \begin{bmatrix} s^*_{00} & s^*_{01} \\ s^*_{10} & s^*_{11} \end{bmatrix} = S^*_{n,n-1} \end{aligned} \tag{2.4-4k}$$

where for convenience $E[Z]$ has been replaced by $\bar{Z}$, that is, $E[\cdot]$ is replaced by the overbar. Again, the assumption is made that mean of $X^*_{n,n-1}$ has been substracted out in the above.

The matrix R_n gives the accuracy of the radar measurements. It is the covariance matrix of the measurement error matrix N_n given by (2.4-4i). For our two-state filter with the measurement equation given by (2.4-3) to (2.4-3c),

$$\begin{aligned} R_n &= \text{COV}[N_n] = \overline{[\nu_n][\nu_n]^T} = \overline{[\nu_n][\nu_n]} \\ &= \overline{[\nu_n^2]} = \overline{[\nu_n^2]} \\ &= [\sigma_\nu^2] = [\sigma_x^2] \end{aligned} \tag{2.4-4l}$$

where it is assumed as in Section 1.2.4.4 that σ_ν and σ_x are the rms of ν_n independent of n. Thus σ_ν^2 and σ_x^2 are the variance of ν_n, the assumption being that the mean of ν_n is zero; see (1.2-18).

The matrix Q_n, which gives the magnitude of the target trajectory uncertainty or the equivalent maneuvering capability, is the covariance matrix of the dynamic model driving noise vector, that is, the random-velocity component of the target trajectory given by (2.4-2a); see also (2.1-1). To get a better feel for Q_n, let us evaluate it for our two-state Kalman filter, that is, for U_n given by (2.4-2a). Here

$$Q_n = \text{COV}\, U_n = \overline{U_n U_n^T} = \overline{\begin{bmatrix} 0 \\ u_n \end{bmatrix} \begin{bmatrix} 0 & u_n \end{bmatrix}}$$
$$= \overline{\begin{bmatrix} 0 \cdot 0 & 0 \cdot u_n \\ u_n \cdot 0 & u_n \cdot u_n \end{bmatrix}} = \begin{bmatrix} 0 & 0 \\ 0 & \overline{u_n^2} \end{bmatrix} \tag{2.4-4m}$$

Equation (2.4-4f) allows us to obtain the prediction covariance matrix $S^*_{n,n-1}$ from the covariance matrix of the filtered estimate of the target state vector at time $n-1$ given by $S^*_{n-1,n-1}$. The filtered estimate covariance matrix $S^*_{n-1,n-1}$ is in turn obtained from the previous prediction covariance matrix $S^*_{n-1,n-2}$ using (2.4-4j). Equations (2.4-4e), (2.4-4f), and (2.4-4j) allow us to obtain the filter weights H_n at successive observation intervals. For the two-state g–h filter discussed earlier, the observation matrix is given by (2.4-3a) and the filter coefficient matrix H_n is given by (2.4-5). The covariance matrix for the initial a priori estimates of the target position and velocity given by $S^*_{0,-1}$ allows initiation of the tracking equations given by (2.4-4d). First (2.4-4e) is used to calculate H_0 (assuming that $n = 0$ is the time for the first filter observation). For convenience the above Kalman filter equations are summarized in Table 2.4-1.

TABLE 2.4-1. Kalman Equation

Predictor equation:

$$X^*_{n+1,n} = \Phi X^*_{n,n} \tag{2.4-4a}$$

Filtering equation:

$$X^*_{n,n} = X^*_{n,n-1} + H_n(Y_n - MX^*_{n,n-1}) \tag{2.4-4d}$$

Weight equation:

$$H_n = S^*_{n,n-1} M^T [R_n + MS^*_{n,n-1} M^T]^{-1} \tag{2.4-4e}$$

Predictor covariance matrix equation:

$$S^*_{n,n-1} = \text{COV}(X^*_{n,n-1}) \tag{2.4-4h}$$
$$S^*_{n,n-1} = \Phi S^*_{n-1,n-1} \Phi^T + Q_n \tag{2.4-4f}$$

Covariance of random system dynamics model noise vector U^a:

$$Q_n = \text{COV}(U_n) = E[U_n U_n^T] \tag{2.4-4g}$$

Covariance of measurement vector $Y_n = X_n + N_n^a$:

$$R_n = \text{COV}(Y_n) = \text{COV}(N_n) = E[N_n N_n^T] \tag{2.4-4i}$$

Corrector equation (covariance of smoothed estimate):

$$S^*_{n-1,n-1} = \text{COV}(X^*_{n-1,n-1}) = (I - H_{n-1}M) S^*_{n-1,n-2} \tag{2.4-4j}$$

[a] If $E[U] = E[N_n] = 0$.

The beauty of the matrix form of the Kalman tracking-filter equations as given by (2.4-4) is, although presented here for our one-dimensional (range only), two-state (position and velocity) case, that the matrix form applies in general. That is, it applies for tracking in any number of dimensions for the measurement and state space and for general dynamics models. All that is necessary is the proper specification of the state vector, observation matrix, transition matrix, dynamics model, and measurement covariance matrix. For example, the equations apply when one is tracking a ballistic target in the atmosphere in three dimensions using rectangular coordinates (x, y, z) with a ten-state vector given by

$$X^*_{n,n-1} = \begin{bmatrix} x^*_{n,n-1} \\ \dot{x}^*_{n,n-1} \\ \ddot{x}^*_{n,n-1} \\ y^*_{n,n-1} \\ \dot{y}^*_{n,n-1} \\ \ddot{y}^*_{n,n-1} \\ z^*_{n,n-1} \\ \dot{z}^*_{n,n-1} \\ \ddot{z}^*_{n,n-1} \\ \beta^*_{n,n-1} \end{bmatrix} \tag{2.4-6}$$

where β is the atmospheric drag on the target. One can assume that the sensor measures R, θ, ϕ, and the target Doppler $\dot{R}$ so that Y_n is given by

$$Y_n = \begin{bmatrix} R_n \\ \dot{R}_n \\ \theta_n \\ \phi_n \end{bmatrix} \tag{2.4-7}$$

In general the vector Y_n would be given by

$$Y_n = \begin{bmatrix} y_{1n} \\ y_{2n} \\ \vdots \\ y_{mn} \end{bmatrix} \tag{2.4-8}$$

where y_{in} is the ith target parameter measured by the sensor at time n.

The atmosheric ballistic coefficient β is given by

$$\beta = \frac{m}{C_D A} \tag{2.4-9}$$

where m is the target mass, C_D is the atmospheric dimensionless drag coefficient dependent on the body shape, and A is the cross-sectional area of the target perpendicular to the direction of motion. [See (16.3-18), (16.3-19), (16.3-27) and (16.3-28) of Section 16.3 for the relation between drag constant and target atmospheric deceleration.]

For the g–h Kalman filter whose dynamics model is given by (2.1-1) or (2.4-2), the matrix Q is given by (2.4-4m), which becomes

$$Q = \begin{bmatrix} 0 & 0 \\ 0 & \sigma_u^2 \end{bmatrix} \tag{2.4-10}$$

if it is assumed that the mean of u_n is zero and its variance is σ_u^2 independent of n. For the equivalent g–h–k Kalman filter to our two-state g–h Kalman filter having the dynamic model of (2.4-2), the three-state dynamics model is given by (1.3-3) with (1.3-3a) replaced by

$$\ddot{x}^*_{n+1,n} = \ddot{x}^*_{n,n} + w_n \tag{2.4-11}$$

where w_n equals a random change in acceleration from time n to $n+1$. We assume w_n is independent from n to $n+1$ for all n and that it has a variance σ_w^2. Physically w_n represents a random-acceleration jump occurring just prior to the $n+1$ observation. For this case

$$Q = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sigma_w^2 \end{bmatrix} \tag{2.4-12}$$

The variance of the target acceleration dynamics σ_w^2 (also called σ_a^2) can be specified using an equation similar to that used for specifying the target velocity dynamics for the Kalman g–h filter. Specifically

$$\sigma_w = \frac{T\ddot{x}_{\max}}{C} \tag{2.4-13}$$

where C is a constant and $\ddot{x}_{\max}$ is the maximum $\ddot{x}$. For the steady-state g–h–k Kalman filter for which Q is given by (2.4-12) g,h, and k are related by (1.3-10a) to (1.3-10c) [11, 14, 15] and σ_a^2, σ_x^2, and T are related to g and k by [14]

$$\frac{T^4\sigma_a^2}{4\sigma_x^2} = \frac{k^2}{1-g} \tag{2.4-14}$$

For the general g–h–k Kalman filter (2.4-5) becomes [14]

$$H_n = \begin{bmatrix} g_n \\ \dfrac{h_n}{T} \\ \dfrac{2k_n}{T^2} \end{bmatrix} \tag{2.4-15}$$

This is a slightly underdamped filter, just as is the steady-state g–h Kalman filter that is the Benedict–Bordner filter. Its total error $E_{TN} = 3\sigma_{n+1,n} + b^*$ is less than that for the critically damped g–h–k filter, and its transient response is about as good as that of the critical damped filter [11]. In the literature, this steady-state Kalman filter has been called the optimum g–h–k filter [11].

If we set $\sigma_u^2 = 0$ in (2.4-10), that is, remove the random maneuvering part of the Kalman dynamics, then

$$Q = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \tag{2.4-16}$$

and we get the growing-memory filter of Section 1.2.10, the filter used for track initiation of the constant g–h filters.

2.5 DERIVATION OF MINIMUM-VARIANCE EQUATION

In Section 2.3 we used the minimum-variance equation (2.3-1) to derive the two-state Kalman filter range-filtering equation. We will now give two derivations of the minimum-variance equation.

2.5.1 First Derivation

The first derivation parallels that of reference 7. For simplicity, designate the two independent estimates $x^*_{n,n-1}$ and y_n by respectively x_1^* and x_2^*. Designate $x^*_{n,n}$, the optimum combined estimate, by x_c^*. We desire to find an optimum linear estimate for x_c^*. We can designate this linear estimate as

$$x_c^* = k_1 x_1^* + k_2 x_2^* \tag{2.5-1}$$