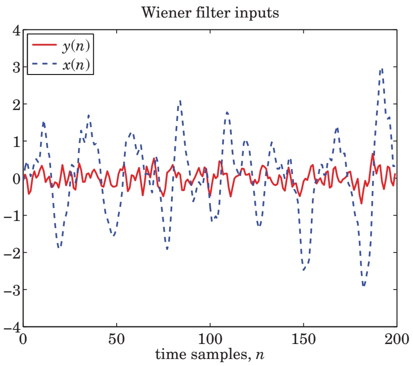
$R(0)=\sigma_y^2$. You can easily determine R_{norm} by doing a maximum entropy extension to order six, starting with the four reflection coefficients and setting $\gamma_5=\gamma_6=0$.)

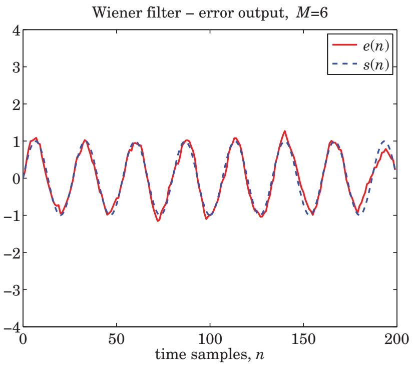
In generating y_n make sure that the transients introduced by the filter have died out. Then, generate the corresponding N samples of the signal x_n. On the same graph, plot x_n together with the desired signal s_n. On a separate graph (but using the same vertical scales as the previous one) plot the reference signal y_n versus n.

b. For $M=4$, design a Wiener filter of order-M based on the generated signal blocks $\{x_n, y_n\}$, $n=0,1,\dots,N-1$, and realize it in both the direct and lattice forms.

c. Using the *lattice* form, filter the signals x_n, y_n through the designed filter and generate the outputs $\hat{x}_n, e_n$. Explain why e_n should be an estimate of the desired signal s_n. On the same graph, plot e_n and s_n using the same vertical scales as in part (a).

d. Repeat parts (b) and (c) for filter orders $M=5,6,7,8$. Discuss the improvement obtained with increasing order. What is the smallest M that would, at least theoretically, result in $e_n=s_n$? Some example graphs are included below.





13 Kalman Filtering

13.1 State-Space Models

The Kalman filter is based on a state/measurement model of the form:

$$\boxed{\begin{aligned}\mathbf{x}_{n+1} &= A_n\mathbf{x}_n + \mathbf{w}_n\\ \mathbf{y}_n &= C_n\mathbf{x}_n + \mathbf{v}_n\end{aligned}}\qquad \begin{aligned}&\text{(state model)}\\ &\text{(measurement model)}\end{aligned}\tag{13.1.1}$$

where $\mathbf{x}_n$ is a p-dimensional state vector and $\mathbf{y}_n$, an r-dimensional vector of observations. The $p\times p$ state-transition matrix A_n and $r\times p$ measurement matrix C_n may depend on time n. The signals $\mathbf{w}_n, \mathbf{v}_n$ are assumed to be mutually-independent, zero-mean, white-noise signals with known covariance matrices Q_n and R_n:

$$\boxed{\begin{aligned}E[\mathbf{w}_n\mathbf{w}_i^T] &= Q_n\delta_{ni}\\ E[\mathbf{v}_n\mathbf{v}_i^T] &= R_n\delta_{ni}\\ E[\mathbf{w}_n\mathbf{v}_i^T] &= 0\end{aligned}}\tag{13.1.2}$$

The model is iterated starting at $n=0$. The initial state vector $\mathbf{x}_0$ is assumed to be random and independent of $\mathbf{w}_n, \mathbf{v}_n$, but with a known mean $\bar{\mathbf{x}}_0=E[\mathbf{x}_0]$ and covariance matrix $\Sigma_0=E[(\mathbf{x}_0-\bar{\mathbf{x}}_0)(\mathbf{x}_0-\bar{\mathbf{x}}_0)^T]$. We will assume, for now, that $\mathbf{x}_0, \mathbf{w}_n, \mathbf{v}_n$ are normally distributed, and therefore, their statistical description is completely determined by their means and covariances. A non-zero cross-covariance $E[\mathbf{w}_n\mathbf{v}_i^T]=S_n\delta_{ni}$ may also be assumed. A scalar version of the model was discussed in Chap. 11.

The covariance matrices Q_n, R_n have dimensions $p\times p$ and $r\times r$, but they need not have full rank (which would mean that some of the components of $\mathbf{x}_n$ or $\mathbf{y}_n$ would, in an appropriate basis, be noise-free.) For example, to allow the possibility of fewer state noise components, the model (13.1.1) is often written in the form:

$$\begin{aligned}\mathbf{x}_{n+1} &= A_n\mathbf{x}_n + G_n\mathbf{w}_n\\ \mathbf{y}_n &= C_n\mathbf{x}_n + \mathbf{v}_n\end{aligned}\qquad \begin{aligned}&\text{(state model)}\\ &\text{(measurement model)}\end{aligned}\tag{13.1.3}$$

where the new $\mathbf{w}_n$ is lower-dimensional with (full-rank) covariance Q_n. In this model, the covariances of the noise components will be $E[(G_n\mathbf{w}_n)(G_n\mathbf{w}_n)^T]= G_nQ_nG_n^T$, In addition, external deterministic inputs may be present, for example,

$$\begin{aligned}
\mathbf{x}_{n+1} &= A_n\mathbf{x}_n + B_n\mathbf{u}_n + G_n\mathbf{w}_n \qquad &&\text{(state model)}\\
\mathbf{y}_n &= C_n\mathbf{x}_n + \mathbf{v}_n &&\text{(measurement model)}
\end{aligned} \tag{13.1.4}$$

where $\mathbf{u}_n$ is the deterministic input. Such modifications do not affect much the essential form of the Kalman filter and, therefore, we will use the simpler model (13.1.1).

The iterated solution of the state equation (13.1.1) may be obtained with the help of the corresponding $p\times p$ state-transition matrix $\Phi_{n,k}$ defined as follows:

$$\boxed{\begin{aligned}
\Phi_{n,k} &= A_{n-1}\cdots A_k\,, \quad \text{for } n>k\\
\Phi_{n,n} &= I\\
\Phi_{n,k} &= \Phi_{k,n}^{-1}\,, \quad \text{for } n<k
\end{aligned}} \tag{13.1.5}$$

where I is the $p\times p$ identity matrix and the third equation is valid only if the inverse exists. In particular, we have $\Phi_{n,0} = A_{n-1}\cdots A_0$ for $n>0$, and $\Phi_{0,0} = I$. If the state matrix A_n is independent of n, then the above definitions imply that $\Phi_{n,k} = A^{n-k}$. We note also the properties:

$$\begin{aligned}
\Phi_{n,n-1} &= A_n\,, \quad n\ge 1\\
\Phi_{n+1,k} &= A_n\Phi_{n,k}\,, \quad n\ge k\\
\Phi_{n,k} &= \Phi_{n,i}\Phi_{i,k}\,, \quad n\ge i\ge k
\end{aligned} \tag{13.1.6}$$

It is easily verified that the solution of Eq. (13.1.1) is given by:

$$\mathbf{x}_n = \Phi_{n,0}\,\mathbf{x}_0 + \sum_{k=1}^{n}\Phi_{n,k}\mathbf{w}_{k-1}\,, \quad n\ge 1 \tag{13.1.7}$$

so that $\mathbf{x}_n$ depends only on $\{\mathbf{x}_0,\mathbf{w}_0,\mathbf{w}_1,\dots,\mathbf{w}_{n-1}\}$, for example,

$$\begin{aligned}
\mathbf{x}_1 &= \Phi_{1,0}\,\mathbf{x}_0 + \Phi_{1,1}\mathbf{w}_0\\
\mathbf{x}_2 &= \Phi_{2,0}\,\mathbf{x}_0 + \Phi_{2,1}\mathbf{w}_0 + \Phi_{2,2}\mathbf{w}_1\\
\mathbf{x}_3 &= \Phi_{3,0}\,\mathbf{x}_0 + \Phi_{3,1}\mathbf{w}_0 + \Phi_{3,2}\mathbf{w}_1 + \Phi_{3,3}\mathbf{w}_2\\
&\;\vdots\\
\mathbf{x}_n &= \Phi_{n,0}\,\mathbf{x}_0 + \Phi_{n,1}\mathbf{w}_0 + \Phi_{n,2}\mathbf{w}_1 + \cdots + \Phi_{n,n}\mathbf{w}_{n-1}
\end{aligned}$$

and more generally, starting at time $n=i$,

$$\mathbf{x}_n = \Phi_{n,i}\,\mathbf{x}_i + \sum_{k=i+1}^{n}\Phi_{n,k}\mathbf{w}_{k-1}\,, \quad n>i \tag{13.1.8}$$

Let $\bar{\mathbf{x}}_n = E[\mathbf{x}_n]$ and $\Sigma_n = E[(\mathbf{x}_n-\bar{\mathbf{x}}_n)(\mathbf{x}_n-\bar{\mathbf{x}}_n)^T]$ be the mean and covariance matrix of the state vector $\mathbf{x}_n$. Then, it follows from Eqs. (13.1.2) and (13.1.7) and the independence of $\mathbf{x}_0$ and $\mathbf{w}_n$ that,

$$\begin{aligned}
\bar{\mathbf{x}}_n &= \Phi_{n,0}\bar{\mathbf{x}}_0\\
\Sigma_n &= \Phi_{n,0}\Sigma_0\Phi_{n,0}^T + \sum_{k=1}^{n}\Phi_{n,k}Q_{k-1}\Phi_{n,k}^T\,, \quad n\ge 1
\end{aligned} \tag{13.1.9}$$

It is straightforward to show from (13.1.9) or (13.1.1) that $\bar{\mathbf{x}}_n$ and Σ_n satisfy the recursions:

$$\begin{aligned}
\bar{\mathbf{x}}_{n+1} &= A_n\bar{\mathbf{x}}_n\\
\Sigma_{n+1} &= A_n\Sigma_nA_n^T + Q_n\,, \quad n\ge 1
\end{aligned} \tag{13.1.10}$$

Indeed, subtracting (13.1.1) and (13.1.10) and using the independence of $\mathbf{x}_n$ and $\mathbf{w}_n$, we find:

$$\begin{aligned}
&\mathbf{x}_{n+1}-\bar{\mathbf{x}}_{n+1} = A_n(\mathbf{x}_n-\bar{\mathbf{x}}_n)+\mathbf{w}_n\\
&\Sigma_{n+1} = E[(\mathbf{x}_{n+1}-\bar{\mathbf{x}}_{n+1})(\mathbf{x}_{n+1}-\bar{\mathbf{x}}_{n+1})^T]= E[(A_n(\mathbf{x}_n-\bar{\mathbf{x}}_n)+\mathbf{w}_n)(A_n(\mathbf{x}_n-\bar{\mathbf{x}}_n)+\mathbf{w}_n)^T]\\
&\quad= A_nE[(\mathbf{x}_n-\bar{\mathbf{x}}_n)(\mathbf{x}_n-\bar{\mathbf{x}}_n)^T]A_n^T + E[\mathbf{w}_n\mathbf{w}_n^T]= A_n\Sigma_nA_n^T + Q_n
\end{aligned}$$

In a similar fashion, we can obtain the statistical properties of the observations $\mathbf{y}_n$ from those of $\mathbf{x}_n$ and $\mathbf{v}_n$:

$$\begin{aligned}
\bar{\mathbf{y}}_n &= C_n\bar{\mathbf{x}}_n\\
\mathbf{y}_n-\bar{\mathbf{y}}_n &= C_n(\mathbf{x}_n-\bar{\mathbf{x}}_n)+\mathbf{v}_n\\
\Sigma_{y_ny_n} &= E[(\mathbf{y}_n-\bar{\mathbf{y}}_n)(\mathbf{y}_n-\bar{\mathbf{y}}_n)^T]= C_n\Sigma_nC_n^T + R_n\,, \quad n\ge 0
\end{aligned} \tag{13.1.11}$$

Example 13.1.1: *Local Level Model.* The local-level model discussed in Chap. 6 is already in state-space form:

$$\begin{aligned}
x_{n+1} &= x_n + w_n\\
y_n &= x_n + v_n
\end{aligned}$$

and represents a random-walk process x_n observed in noise. The noise variances are defined as $Q=\sigma_w^2$ and $R=\sigma_v^2$. □

Example 13.1.2: *Local Trend Model.* The local-trend model was discussed in Sec. 6.13. Let a_n, b_n be the local level and local slope. The model is defined by,

$$\begin{aligned}
a_{n+1} &= a_n + b_n + w_n\\
b_{n+1} &= b_n + u_n\\
y_n &= a_n + v_n
\end{aligned}$$

with mutually uncorrelated noise components w_n, u_n, v_n. The model can be written in state-space form as follows:

$$\begin{bmatrix} a_{n+1}\\ b_{n+1}\end{bmatrix} = \begin{bmatrix} 1 & 1\\ 0 & 1\end{bmatrix}\begin{bmatrix} a_n\\ b_n\end{bmatrix} + \begin{bmatrix} w_n\\ u_n\end{bmatrix}, \quad y_n = [1,0]\begin{bmatrix} a_n\\ b_n\end{bmatrix} + v_n$$

The noise covariances are:

$$\mathbf{w}_n = \begin{bmatrix} w_n\\ u_n\end{bmatrix}, \quad Q = E[\mathbf{w}_n\mathbf{w}_n^T]= \begin{bmatrix} \sigma_w^2 & 0\\ 0 & \sigma_u^2\end{bmatrix}, \quad R=\sigma_v^2$$

As we mentioned in Sec. 6.13, the steady-state version of the Kalman filter for this model is equivalent to Holt's exponential smoothing method. We demonstrate this later. □

Example 13.1.3: *Kinematic Models for Radar Tracking.* Consider the one-dimensional motion of an object moving with constant acceleration, $\ddot{x}(t)=a$. By integrating this equation, the object's position $x(t)$ and velocity $\dot{x}(t)$ are,

$$\begin{aligned} x(t) &= x(t_0)+(t-t_0)\dot{x}(t_0)+\frac{1}{2}(t-t_0)^2 a \\ \dot{x}(t) &= \dot{x}(t_0)+(t-t_0)a \end{aligned} \tag{13.1.12}$$

Suppose the motion is sampled at time intervals T, i.e., at the time instants $t_n = nT$, and let us assume that the acceleration is not necessarily constant for all t, but is constant within each interval T, that is, $a(t)=a(t_n)$, for $t_n \le t < t_{n+1}$. Then, applying Eq. (13.1.12) at $t = t_{n+1}$ and $t_0 = t_n$, and denoting $x(t_n)=x_n$, $\dot{x}(t_n)=\dot{x}_n$, and $a(t_n)=a_n$, we obtain,

$$\begin{aligned} x_{n+1} &= x_n + T\dot{x}_n + \frac{1}{2}T^2 a_n \\ \dot{x}_{n+1} &= \dot{x}_n + Ta_n \end{aligned} \tag{13.1.13}$$

To describe an object that is trying to move at constant speed but is subject to random accelerations, we may assume that a_n is a zero-mean random variable with variance σ_a^2. If the object's position x_n is observed in noise v_n, we obtain the model:

$$\begin{aligned} x_{n+1} &= x_n + T\dot{x}_n + \frac{1}{2}T^2 a_n \\ \dot{x}_{n+1} &= \dot{x}_n + Ta_n \\ y_n &= x_n + v_n \end{aligned} \tag{13.1.14}$$

which may be written in state-space form:

$$\begin{aligned} \begin{bmatrix} x_{n+1} \\ \dot{x}_{n+1} \end{bmatrix} &= \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_n \\ \dot{x}_n \end{bmatrix} + \begin{bmatrix} T^2/2 \\ T \end{bmatrix} a_n \\ y_n &= [1, 0] \begin{bmatrix} x_n \\ \dot{x}_n \end{bmatrix} + v_n \end{aligned} \tag{13.1.15}$$

with measurement noise variance $R = \sigma_v^2$, and state noise vector and covariance matrix:

$$\mathbf{w}_n = \begin{bmatrix} T^2/2 \\ T \end{bmatrix} a_n \quad \Rightarrow \quad Q = E[\mathbf{w}_n\mathbf{w}_n^T] = \begin{bmatrix} T^4/4 & T^3/2 \\ T^3/2 & T^2 \end{bmatrix} \sigma_a^2 \tag{13.1.16}$$

This model is, of course, very similar to the local-trend model of the previous example if we set $T = 1$, except now the state noise arises from a single acceleration noise a_n affecting both components of the state vector, whereas in the local-trend model, we assumed independent noises for the local level and local slope.

We will see later that the steady-state Kalman filter for the model defined by Eqs. (13.1.15) and (13.1.16) is equivalent to an optimum version of the popular α–β radar tracking filter [868,869]. An alternative model, which leads to a somewhat different α–β tracking model [870,874], has state noise added only to the velocity component:

$$\begin{aligned} \begin{bmatrix} x_{n+1} \\ \dot{x}_{n+1} \end{bmatrix} &= \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_n \\ \dot{x}_n \end{bmatrix} + \begin{bmatrix} 0 \\ w_n \end{bmatrix} \\ y_n &= [1, 0] \begin{bmatrix} x_n \\ \dot{x}_n \end{bmatrix} + v_n \end{aligned} \tag{13.1.17}$$

with $R = \sigma_v^2$ and

$$\mathbf{w}_n = \begin{bmatrix} 0 \\ w_n \end{bmatrix}, \quad Q = E[\mathbf{w}_n\mathbf{w}_n^T] = \begin{bmatrix} 0 & 0 \\ 0 & \sigma_w^2 \end{bmatrix}$$

The models (13.1.15) and (13.1.17) are appropriate for uniformly moving objects subject to random accelerations. In order to describe a maneuvering, accelerating, object, we may start with the model (13.1.15) and make the acceleration a_n part of the state vector and assume that it deviates from a constant acceleration by an additive white noise term, i.e., replace a_n by $a_n + w_n$. Denoting a_n by $\ddot{x}_n$, we obtain the model [874,875]:

$$\begin{aligned} x_{n+1} &= x_n + T\dot{x}_n + \frac{1}{2}T^2(\ddot{x}_n + w_n) \\ \dot{x}_{n+1} &= \dot{x}_n + T(\ddot{x}_n + w_n) \\ \ddot{x}_{n+1} &= \ddot{x}_n + w_n \\ y_n &= x_n + v_n \end{aligned} \tag{13.1.18}$$

which may be written in the matrix form:

$$\begin{aligned} \begin{bmatrix} x_{n+1} \\ \dot{x}_{n+1} \\ \ddot{x}_{n+1} \end{bmatrix} &= \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_n \\ \dot{x}_n \\ \ddot{x}_n \end{bmatrix} + \begin{bmatrix} T^2/2 \\ T \\ 1 \end{bmatrix} w_n \\ y_n &= [1, 0, 0] \begin{bmatrix} x_n \\ \dot{x}_n \\ \ddot{x}_n \end{bmatrix} + v_n \end{aligned} \tag{13.1.19}$$

This leads to the so-called α–β–γ tracking filter. An alternative model may be derived by starting with a linearly increasing acceleration $\ddot{x}(t)=a(t)=a(t_0)+(t-t_0)\dot{a}(t_0)$, whose integration gives:

$$\begin{aligned} x(t) &= x(t_0)+(t-t_0)u(t_0)+\frac{1}{2}(t-t_0)^2 a(t_0)+\frac{1}{6}(t-t_0)^3\dot{a}(t_0) \\ \dot{x}(t) &= \dot{x}(t_0)+(t-t_0)a(t_0)+\frac{1}{2}(t-t_0)^2\dot{a}(t_0) \\ a(t) &= a(t_0)+(t-t_0)\dot{a}(t_0) \end{aligned} \tag{13.1.20}$$

Its sampled version is obtained by treating the acceleration rate $\dot{a}_n$ as a zero-mean white-noise term with variance $\sigma_{\dot{a}}^2$, resulting in the state model [876]:

$$\begin{aligned} \begin{bmatrix} x_{n+1} \\ \dot{x}_{n+1} \\ \ddot{x}_{n+1} \end{bmatrix} &= \begin{bmatrix} 1 & T & T^2/2 \\ 0 & 1 & T \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_n \\ \dot{x}_n \\ \ddot{x}_n \end{bmatrix} + \begin{bmatrix} T^3/3 \\ T^2/2 \\ T \end{bmatrix} \dot{a}_n \\ y_n &= [1, 0, 0] \begin{bmatrix} x_n \\ \dot{x}_n \\ \ddot{x}_n \end{bmatrix} + v_n \end{aligned} \tag{13.1.21}$$

Later on we will look at the Kalman filters for such kinematic models and discuss their connection to the α–β and α–β–γ tracking filters. □

13.2 Kalman Filter

The Kalman filter is a time-recursive procedure for estimating the state vector $\mathbf{x}_n$ from the observations signal $\mathbf{y}_n$. Let $Y_n = \{\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_n\}$ be the linear span of the observations up to the current time instant n. The Kalman filter estimate of $\mathbf{x}_n$ is defined as the optimum linear combination of these observations that minimizes the mean-square estimation error, and as we already know, it is given by the projection of $\mathbf{x}_n$ onto the observation subspace Y_n. For the gaussian case, this projection happens to be the conditional mean $\hat{\mathbf{x}}_{n/n} = E[\mathbf{x}_n|Y_n]$. Let us define also the predicted estimate $\hat{\mathbf{x}}_{n/n-1}$ based on the observations $Y_{n-1} = \{\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_{n-1}\}$. Again, for the gaussian case we have $\hat{\mathbf{x}}_{n/n-1} = E[\mathbf{x}_n|Y_{n-1}]$. To cover both the gaussian and nongaussian, but linear, cases, we will use the following notation for the estimates, estimation errors, and mean-square error covariances:

$$\boxed{\begin{aligned} \hat{\mathbf{x}}_{n/n-1} &= \mathrm{Proj}[\mathbf{x}_n|Y_{n-1}] \\ \mathbf{e}_{n/n-1} &= \mathbf{x}_n - \hat{\mathbf{x}}_{n/n-1} \\ P_{n/n-1} &= E[\mathbf{e}_{n/n-1}\mathbf{e}_{n/n-1}^T] \end{aligned}} \quad \text{and} \quad \boxed{\begin{aligned} \hat{\mathbf{x}}_{n/n} &= \mathrm{Proj}[\mathbf{x}_n|Y_n] \\ \mathbf{e}_{n/n} &= \mathbf{x}_n - \hat{\mathbf{x}}_{n/n} \\ P_{n/n} &= E[\mathbf{e}_{n/n}\mathbf{e}_{n/n}^T] \end{aligned}} \tag{13.2.1}$$

We will first state the estimation algorithm, and then prove it. The Kalman filtering algorithm for the model (13.1.1)–(13.1.2) is as follows:

$$\boxed{\begin{array}{ll}
\multicolumn{2}{l}{\text{Initialize in time by: } \hat{\mathbf{x}}_{0/-1} = \bar{\mathbf{x}}_0,\ P_{0/-1} = \Sigma_0} \\
\multicolumn{2}{l}{\text{At time } n,\ \hat{\mathbf{x}}_{n/n-1},\ P_{n/n-1},\ \mathbf{y}_n \text{ are available,}} \\
\quad D_n = C_n P_{n/n-1} C_n^T + R_n & \text{innovations covariance} \\
\quad G_n = P_{n/n-1} C_n^T D_n^{-1} & \text{Kalman gain for filtering} \\
\quad K_n = A_n G_n = A_n P_{n/n-1} C_n^T D_n^{-1} & \text{Kalman gain for prediction} \\
\quad \hat{\mathbf{y}}_{n/n-1} = C_n \hat{\mathbf{x}}_{n/n-1} & \text{predicted measurement} \\
\quad \boldsymbol{\varepsilon}_n = \mathbf{y}_n - \hat{\mathbf{y}}_{n/n-1} = \mathbf{y}_n - C_n \hat{\mathbf{x}}_{n/n-1} & \text{innovations sequence} \\
\multicolumn{2}{l}{\text{Measurement update / correction:}} \\
\quad \hat{\mathbf{x}}_{n/n} = \hat{\mathbf{x}}_{n/n-1} + G_n \boldsymbol{\varepsilon}_n & \text{filtered estimate} \\
\quad P_{n/n} = P_{n/n-1} - G_n D_n G_n^T & \text{estimation error} \\
\multicolumn{2}{l}{\text{Time update / prediction:}} \\
\quad \hat{\mathbf{x}}_{n+1/n} = A_n \hat{\mathbf{x}}_{n/n} = A_n \hat{\mathbf{x}}_{n/n-1} + K_n \boldsymbol{\varepsilon}_n & \text{predicted estimate} \\
\quad P_{n+1/n} = A_n P_{n/n} A_n^T + Q_n & \text{prediction error} \\
\multicolumn{2}{l}{\text{Go to time } n+1}
\end{array}} \tag{13.2.2}$$

The quantity D_n represents the innovations covariance matrix, that is, $D_n = E[\boldsymbol{\varepsilon}_n \boldsymbol{\varepsilon}_n^T]$. The innovations sequence $\{\boldsymbol{\varepsilon}_0, \boldsymbol{\varepsilon}_1, \dots, \boldsymbol{\varepsilon}_n\}$, constructed recursively by the algorithm, represents the Gram-Schmidt orthogonalization of the observations $\{\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_n\}$ in the sense that the innovations form an orthogonal basis for the observation subspace Y_n, that is, Y_n is the linear span of either set:

$$Y_n = \{\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_n\} = \{\boldsymbol{\varepsilon}_0, \boldsymbol{\varepsilon}_1, \dots, \boldsymbol{\varepsilon}_n\}$$

The orthogonality property of $\boldsymbol{\varepsilon}_n$ is expressed by:

$$\boxed{E[\boldsymbol{\varepsilon}_n \boldsymbol{\varepsilon}_i^T] = D_n \delta_{ni}} \tag{13.2.3}$$

There are some alternative ways of writing the above equations. For example, the equation for $P_{n/n}$ may be written in the equivalent ways:

$$\boxed{\begin{array}{l}
1.\ P_{n/n} = P_{n/n-1} - G_n D_n G_n^T = (I - G_n C_n) P_{n/n-1} \\
2.\ P_{n/n} = P_{n/n-1} - P_{n/n-1} C_n^T D_n^{-1} C_n P_{n/n-1} = \text{standard form} \\
3.\ P_{n/n} = (I - G_n C_n) P_{n/n-1} (I - G_n C_n)^T + G_n R_n G_n^T = \text{Joseph form} \\
4.\ P_{n/n} = \left[P_{n/n-1}^{-1} + C_n^T R_n^{-1} C_n\right]^{-1} = \text{information form}
\end{array}} \tag{13.2.4}$$

with $D_n = C_n P_{n/n-1} C_n^T + R_n$ and $G_n = P_{n/n-1} C_n^T D_n^{-1}$. Similarly, we can write the Kalman gain G_n in its information form:

$$\boxed{G_n = P_{n/n-1} C_n^T D_n^{-1} = P_{n/n} C_n^T R_n^{-1}} \tag{13.2.5}$$

It follows from the information forms that the filtered estimate may be re-expressed as:

$$\begin{aligned}
\hat{\mathbf{x}}_{n/n} &= \hat{\mathbf{x}}_{n/n-1} + G_n \boldsymbol{\varepsilon}_n = \hat{\mathbf{x}}_{n/n-1} + P_{n/n} C_n^T R_n^{-1} (\mathbf{y}_n - C_n \hat{\mathbf{x}}_{n/n-1}) \\
&= P_{n/n} [P_{n/n}^{-1} - C_n^T R_n^{-1} C_n] \hat{\mathbf{x}}_{n/n-1} + P_{n/n} C_n^T R_n^{-1} \mathbf{y}_n \\
&= P_{n/n} P_{n/n-1}^{-1} \hat{\mathbf{x}}_{n/n-1} + P_{n/n} C_n^T R_n^{-1} \mathbf{y}_n
\end{aligned}$$

from which we obtain the information form of the updating equation:

$$\boxed{P_{n/n}^{-1} \hat{\mathbf{x}}_{n/n} = P_{n/n-1}^{-1} \hat{\mathbf{x}}_{n/n-1} + C_n^T R_n^{-1} \mathbf{y}_n} \tag{13.2.6}$$

In the relations that involve R_n^{-1}, one must assume that R_n has full rank. The difference equation for the predicted estimate may be written directly in terms of the current observation $\mathbf{y}_n$ and the *closed-loop* state matrix $F_n = A_n - K_n C_n$, as follows:

$$\begin{aligned}
\hat{\mathbf{x}}_{n+1/n} &= A_n \hat{\mathbf{x}}_{n/n-1} + K_n \boldsymbol{\varepsilon}_n = A_n \hat{\mathbf{x}}_{n/n-1} + K_n (\mathbf{y}_n - C_n \hat{\mathbf{x}}_{n/n-1}) \\
&= (A_n - K_n C_n) \hat{\mathbf{x}}_{n/n-1} + K_n \mathbf{y}_n
\end{aligned}$$

that is,

$$\boxed{\hat{\mathbf{x}}_{n+1/n} = (A_n - K_n C_n) \hat{\mathbf{x}}_{n/n-1} + K_n \mathbf{y}_n} \tag{13.2.7}$$

A block diagram realization is depicted in Fig. 13.2.1. The error covariance update equations,

$$\begin{aligned}
P_{n/n} &= P_{n/n-1} - P_{n/n-1} C_n^T D_n^{-1} C_n P_{n/n-1} \\
P_{n+1/n} &= A_n P_{n/n} A_n^T + Q_n
\end{aligned}$$

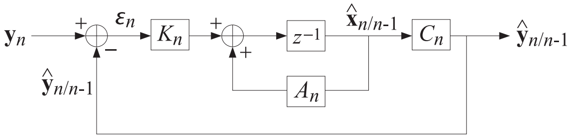

$\mathbf{y}_n$
+
ε_n
K_n
+
z^{-1}
$\hat{\mathbf{x}}_{n/n\text{-}1}$
C_n
$\hat{\mathbf{y}}_{n/n\text{-}1}$
−
+
A_n
$\hat{\mathbf{y}}_{n/n\text{-}1}$

Fig. 13.2.1 Kalman filter realization.

may be combined into a single equation known as the discrete-time *Riccati difference* equation, to be initialized at $P_{0/-1} = \Sigma_0$:

$$\boxed{P_{n+1/n} = A_n\Big[P_{n/n-1} - P_{n/n-1}C_n^T(C_nP_{n/n-1}C_n^T + R_n)^{-1}C_nP_{n/n-1}\Big]A_n^T + Q_n} \tag{13.2.8}$$

which can also be written in the "information" forms (if R_n^{-1} exists):

$$\boxed{\begin{aligned} P_{n+1/n} &= A_n[P_{n/n-1}^{-1} + C_n^TR_n^{-1}C_n)]^{-1}A_n^T + Q_n \\ P_{n+1/n} &= A_n[I + P_{n/n-1}C_n^TR_n^{-1}C_n)]^{-1}P_{n/n-1}A_n^T + Q_n \end{aligned}} \tag{13.2.9}$$

and in the Joseph-like forms:

$$\boxed{\begin{aligned} P_{n+1/n} &= A_nP_{n/n-1}A_n^T + Q_n - K_nD_nK_n^T\,, \quad K_n = A_nP_{n/n-1}C_n^TD_n^{-1} \\ P_{n+1/n} &= (A_n - K_nC_n)P_{n/n-1}(A_n - K_nC_n)^T + K_nR_nK_n^T + Q_n \end{aligned}} \tag{13.2.10}$$

Similarly, the closed-loop transition matrix can be written as,

$$\boxed{F_n = A_n - K_nC_n = A_n[I + P_{n/n-1}C_n^TR_n^{-1}C_n]^{-1}} \tag{13.2.11}$$

We note also that since, $\hat{\mathbf{x}}_{n+1/n} = A_n\hat{\mathbf{x}}_{n/n}$ and $\hat{\mathbf{x}}_{n/n-1} = A_{n-1}\hat{\mathbf{x}}_{n-1/n-1}$, the difference equations for the predicted and filtered estimates would read as follows in terms of the Kalman gains K_n and G_n, respectively,

$$\boxed{\begin{array}{ll} \hat{\mathbf{y}}_{n/n-1} = C_n\hat{\mathbf{x}}_{n/n-1} = C_nA_{n-1}\hat{\mathbf{x}}_{n-1/n-1} & \text{predicted measurement} \\ \boldsymbol{\varepsilon}_n = \mathbf{y}_n - \hat{\mathbf{y}}_{n/n-1} & \text{innovations sequence} \\ \hat{\mathbf{x}}_{n+1/n} = A_n\hat{\mathbf{x}}_{n/n-1} + K_n\boldsymbol{\varepsilon}_n & \text{predicted estimate} \\ \hat{\mathbf{x}}_{n/n} = A_{n-1}\hat{\mathbf{x}}_{n-1/n-1} + G_n\boldsymbol{\varepsilon}_n & \text{filtered estimate} \end{array}} \tag{13.2.12}$$

13.3 Derivation

To derive Eq. (13.2.2), we recall from Chap. 1 that the optimum linear estimate of a zero-mean random vector $\mathbf{x}$ based on a zero-mean random vector $\mathbf{y}$, and the corresponding

orthogonality condition and error covariance, are given by:

$$\begin{aligned} \hat{\mathbf{x}} &= R_{xy}R_{yy}^{-1}\mathbf{y} = E[\mathbf{x}\mathbf{y}^T]E[\mathbf{y}\mathbf{y}^T]^{-1}\mathbf{y} \\ \mathbf{e} &= \mathbf{x} - \hat{\mathbf{x}}\,, \quad R_{ey} = E[\mathbf{e}\mathbf{y}^T] = 0 \\ R_{ee} &= E[\mathbf{e}\mathbf{e}^T] = R_{xx} - R_{xy}R_{yy}^{-1}R_{yx} \end{aligned} \tag{13.3.1}$$

When the vectors have non-zero means, say $\bar{\mathbf{x}}, \bar{\mathbf{y}}$, the same results apply with the replacement $\mathbf{x} \to \mathbf{x} - \bar{\mathbf{x}}$, $\mathbf{y} \to \mathbf{y} - \bar{\mathbf{y}}$, and $\hat{\mathbf{x}} \to \hat{\mathbf{x}} - \bar{\mathbf{x}}$ in (13.3.1). Under this replacement, the correlation matrices are replaced by the corresponding covariances, e.g.,

$$R_{xy} = E[\mathbf{x}\mathbf{y}^T] \to E[(\mathbf{x} - \bar{\mathbf{x}})(\mathbf{y} - \bar{\mathbf{y}})^T] = \Sigma_{xy}\,, \quad \text{etc.}$$

Hence, the optimum estimate is now:

$$\begin{aligned} \hat{\mathbf{x}} &= \bar{\mathbf{x}} + \Sigma_{xy}\Sigma_{yy}^{-1}(\mathbf{y} - \bar{\mathbf{y}}) \\ \mathbf{e} &= \mathbf{x} - \hat{\mathbf{x}}\,, \quad R_{ey} = \Sigma_{ey} = E[\mathbf{e}\mathbf{y}^T] = 0 \\ \Sigma_{ee} &= \Sigma_{xx} - \Sigma_{xy}\Sigma_{yy}^{-1}\Sigma_{yx} \end{aligned} \tag{13.3.2}$$

We note that the estimate is unbiased, that is, $E[\hat{\mathbf{x}}] = \bar{\mathbf{x}}$, and therefore, the estimation error has zero mean, $E[\mathbf{e}] = 0$, its covariance matrix will be $\Sigma_{ee} = R_{ee} = E[\mathbf{e}\mathbf{e}^T]$, and the orthogonality condition may be written as $E[\mathbf{e}(\mathbf{y} - \bar{\mathbf{y}})^T] = E[\mathbf{e}\mathbf{y}^T] = 0$.

Let us apply now this result to the state model (13.1.1) at $n = 0$. This will also clarify the manner in which the algorithm is to be initialized. As part of the model, we assume that the initial state $\mathbf{x}_0$ has a known mean $\bar{\mathbf{x}}_0$ and covariance matrix Σ_0. According to (13.3.2), the optimum estimate $\hat{\mathbf{x}}_{0/0}$ of $\mathbf{x}_0$ based on the observation $\mathbf{y}_0$ will be given by:

$$\begin{aligned} \hat{\mathbf{x}}_{0/0} &= \bar{\mathbf{x}}_0 + \Sigma_{x_0y_0}\Sigma_{y_0y_0}^{-1}(\mathbf{y}_0 - \bar{\mathbf{y}}_0) \\ \mathbf{e}_{0/0} &= \mathbf{x}_0 - \hat{\mathbf{x}}_{0/0}\,, \quad E[\mathbf{e}_{0/0}(\mathbf{y}_0 - \bar{\mathbf{y}}_0)^T] = 0 \\ \Sigma_{e_0e_0} &= E[\mathbf{e}_{0/0}\mathbf{e}_{0/0}^T] = \Sigma_{x_0x_0} - \Sigma_{x_0y_0}\Sigma_{y_0y_0}^{-1}\Sigma_{y_0x_0} \end{aligned} \tag{13.3.3}$$

with $E[\hat{\mathbf{x}}_{0/0}] = \bar{\mathbf{x}}_0$ and $E[\mathbf{e}_{0/0}] = 0$. Let us define $\boldsymbol{\varepsilon}_0 = \mathbf{y}_0 - \bar{\mathbf{y}}_0$ and $G_0 = \Sigma_{x_0y_0}\Sigma_{y_0y_0}^{-1}$. From the measurement model $\mathbf{y}_0 = C_0\mathbf{x}_0 + \mathbf{v}_0$, we have $\bar{\mathbf{y}}_0 = C_0\bar{\mathbf{x}}_0$, which gives:

$$\boldsymbol{\varepsilon}_0 = \mathbf{y}_0 - \bar{\mathbf{y}}_0 = C_0(\mathbf{x}_0 - \bar{\mathbf{x}}_0) + \mathbf{v}_0$$

Clearly, $E[\boldsymbol{\varepsilon}_0] = 0$. Since $\mathbf{v}_0$ is uncorrelated with $\mathbf{x}_0$, we will have:

$$E[\boldsymbol{\varepsilon}_0\boldsymbol{\varepsilon}_0^T] = C_0E[(\mathbf{x}_0 - \bar{\mathbf{x}}_0)(\mathbf{x}_0 - \bar{\mathbf{x}}_0)^T]C_0^T + E[\mathbf{v}_0\mathbf{v}_0^T]\,, \quad \text{or,}$$

$$D_0 = E[\boldsymbol{\varepsilon}_0\boldsymbol{\varepsilon}_0^T] = \Sigma_{y_0y_0} = C_0\Sigma_0C_0^T + R_0$$

where $\Sigma_0 = E[(\mathbf{x}_0 - \bar{\mathbf{x}}_0)(\mathbf{x}_0 - \bar{\mathbf{x}}_0)^T] = \Sigma_{x_0x_0}$. Similarly, we find:

$$\Sigma_{x_0y_0} = E[(\mathbf{x}_0 - \bar{\mathbf{x}}_0)\boldsymbol{\varepsilon}_0^T] = E[(\mathbf{x}_0 - \bar{\mathbf{x}}_0)(\mathbf{x}_0 - \bar{\mathbf{x}}_0)^T]C_0^T = \Sigma_0C_0^T$$

Thus, the Kalman gain becomes,

$$G_0 = \Sigma_{x_0y_0}\Sigma_{y_0y_0}^{-1} = \Sigma_0 C_0^T D_0^{-1}$$

With the definitions $\hat{\mathbf{x}}_{0/-1} = \bar{\mathbf{x}}_0$ and $P_{0/-1} = \Sigma_0$, we may rewrite (13.3.3) as,

$$\hat{\mathbf{x}}_{0/0} = \hat{\mathbf{x}}_{0/-1} + G_0\boldsymbol{\varepsilon}_0\,, \quad G_0 = P_{0/-1}C_0^T D_0^{-1} \tag{13.3.4}$$

The corresponding error covariance matrix will be given by (13.3.3):

$$\Sigma_{e_0e_0} = \Sigma_{x_0x_0} - \Sigma_{x_0y_0}\Sigma_{y_0y_0}^{-1}\Sigma_{y_0x_0} = \Sigma_0 - \Sigma_0 C_0^T D_0^{-1} C_0 \Sigma_0\,, \quad \text{or,}$$

$$P_{0/0} = P_{0/-1} - P_{0/-1}C_0^T D_0^{-1} C_0 P_{0/-1} = P_{0/-1} - G_0 D_0 G_0^T \tag{13.3.5}$$

Because $\mathbf{e}_{0/0}$ has zero mean, we may write the orthogonality condition in Eq. (13.3.3) as,

$$E[\mathbf{e}_{0/0}\mathbf{y}_0^T] = E[\mathbf{e}_{0/0}(\mathbf{y}_0 - \bar{\mathbf{y}}_0)^T] = E[\mathbf{e}_{0/0}\boldsymbol{\varepsilon}_0^T] = 0$$

which states that the estimation error is orthogonal to the observation $\mathbf{y}_0$, or equivalently, to the innovations vector $\boldsymbol{\varepsilon}_0$. To complete the $n = 0$ step of the algorithm, we must now determine the prediction of $\mathbf{x}_1$ based on $\mathbf{y}_0$ or $\boldsymbol{\varepsilon}_0$. We may apply (13.3.2) again,

$$\begin{aligned}
&\hat{\mathbf{x}}_{1/0} = \bar{\mathbf{x}}_1 + \Sigma_{x_1y_0}\Sigma_{y_0y_0}^{-1}(\mathbf{y}_0 - \bar{\mathbf{y}}_0) = \bar{\mathbf{x}}_1 + E[\mathbf{x}_1\boldsymbol{\varepsilon}_0]E[\boldsymbol{\varepsilon}_0\boldsymbol{\varepsilon}_0^T]^{-1}\boldsymbol{\varepsilon}_0 \\
&\mathbf{e}_{1/0} = \mathbf{x}_1 - \hat{\mathbf{x}}_{1/0}\,, \quad E[\mathbf{e}_{1/0}(\mathbf{y}_0 - \bar{\mathbf{y}}_0)^T] = E[\mathbf{e}_{1/0}\boldsymbol{\varepsilon}_0^T] = 0 \\
&\Sigma_{e_1e_1} = E[\mathbf{e}_{1/0}\mathbf{e}_{1/0}^T] = \Sigma_{x_1x_1} - \Sigma_{x_1y_0}\Sigma_{y_0y_0}^{-1}\Sigma_{y_0x_1}
\end{aligned} \tag{13.3.6}$$

From the state equation $\mathbf{x}_1 = A_0\mathbf{x}_0 + \mathbf{w}_0$ and the independence of $\mathbf{w}_0$ and $\mathbf{y}_0$, we find,

$$\begin{aligned}
\Sigma_{x_1y_0} &= \Sigma_{(A_0x_0+w_0)y_0} = A_0\Sigma_{x_0y_0} \\
K_0 &\equiv \Sigma_{x_1y_0}\Sigma_{y_0y_0}^{-1} = A_0\Sigma_{x_0y_0}\Sigma_{y_0y_0}^{-1} = A_0 G_0 \\
\Sigma_{x_1x_1} &= \Sigma_{(A_0x_0+w_0)(A_0x_0+w_0)} = A_0\Sigma_{x_0x_0}A_0^T + Q_0 \\
P_{1/0} &= \Sigma_{e_1e_1} = \Sigma_{x_1x_1} - \Sigma_{x_1y_0}\Sigma_{y_0y_0}^{-1}\Sigma_{y_0x_1} \\
&= A_0\Sigma_{x_0x_0}A_0^T + Q_0 - A_0\Sigma_{x_0y_0}\Sigma_{y_0y_0}^{-1}\Sigma_{y_0x_0}A_0^T \\
&= A_0[\Sigma_{x_0x_0} - \Sigma_{x_0y_0}\Sigma_{y_0y_0}^{-1}\Sigma_{y_0x_0}]A_0^T + Q_0 = A_0 P_{0/0} A_0^T + Q_0
\end{aligned}$$

Since, $\bar{\mathbf{x}}_1 = A_0\bar{\mathbf{x}}_0 = A_0\hat{\mathbf{x}}_{0/-1}$, we may rewrite the predicted estimate and its error as,

$$\begin{aligned}
\hat{\mathbf{x}}_{1/0} &= A_0\hat{\mathbf{x}}_{0/-1} + K_0\boldsymbol{\varepsilon}_0 = A_0[\hat{\mathbf{x}}_{0/-1} + G_0\boldsymbol{\varepsilon}_0] = A_0\hat{\mathbf{x}}_{0/0} \\
P_{1/0} &= A_0 P_{0/0} A_0^T + Q_0
\end{aligned} \tag{13.3.7}$$

This completes all the steps at $n = 0$. We collect the results together:

$$\boxed{\begin{aligned}
&\hat{\mathbf{x}}_{0/-1} = \bar{\mathbf{x}}_0\,, \quad P_{0/-1} = \Sigma_0 \\
&D_0 = C_0 P_{0/-1} C_0^T + R_0 \\
&G_0 = P_{0/-1}C_0^T D_0^{-1} \\
&K_0 = A_0 G_0 = A_0 P_{0/-1} C_0^T D_0^{-1} \\
&\hat{\mathbf{y}}_{0/-1} = C_0\hat{\mathbf{x}}_{0/-1} \\
&\boldsymbol{\varepsilon}_0 = \mathbf{y}_0 - \hat{\mathbf{y}}_{0/-1} = \mathbf{y}_0 - C_0\hat{\mathbf{x}}_{0/-1} \\
&\hat{\mathbf{x}}_{0/0} = \hat{\mathbf{x}}_{0/-1} + G_0\boldsymbol{\varepsilon}_0 \\
&P_{0/0} = P_{0/-1} - G_0 D_0 G_0^T \\
&\hat{\mathbf{x}}_{1/0} = A_0\hat{\mathbf{x}}_{0/0} = A_0\hat{\mathbf{x}}_{0/-1} + K_0\boldsymbol{\varepsilon}_0 \\
&P_{1/0} = A_0 P_{0/0} A_0^T + Q_0
\end{aligned}}$$

Moving on to $n = 1$, we construct the next innovations vector $\boldsymbol{\varepsilon}_1$ by:

$$\boldsymbol{\varepsilon}_1 = \mathbf{y}_1 - \hat{\mathbf{y}}_{1/0} = \mathbf{y}_1 - C_1\hat{\mathbf{x}}_{1/0} \tag{13.3.8}$$

Since $\mathbf{y}_1 = C_1\mathbf{x}_1 + \mathbf{v}_1$, it follows that,

$$\boldsymbol{\varepsilon}_1 = C_1(\mathbf{x}_1 - \hat{\mathbf{x}}_{1/0}) + \mathbf{v}_1 = C_1\mathbf{e}_{1/0} + \mathbf{v}_1 \tag{13.3.9}$$

Because $\boldsymbol{\varepsilon}_0$ is orthogonal to $\mathbf{e}_{1/0}$ and $\mathbf{v}_1$ is independent of $\mathbf{y}_0$, we have:

$$E[\boldsymbol{\varepsilon}_1\boldsymbol{\varepsilon}_0^T] = 0 \tag{13.3.10}$$

We also have $E[\boldsymbol{\varepsilon}_1] = 0$ and the covariance matrix:

$$D_1 = E[\boldsymbol{\varepsilon}_1\boldsymbol{\varepsilon}_1^T] = C_1 P_{1/0} C_1^T + R_1 \tag{13.3.11}$$

Thus, the zero-mean vectors $\{\boldsymbol{\varepsilon}_0, \boldsymbol{\varepsilon}_1\}$ form an orthogonal basis for the subspace $Y_1 = \{\mathbf{y}_0, \mathbf{y}_1\}$. The optimum estimate of $\mathbf{x}_1$ based on Y_1 is obtained by Eq. (13.3.2), but with $\mathbf{y}$ replaced by the extended basis vector $\begin{bmatrix}\boldsymbol{\varepsilon}_0 \\ \boldsymbol{\varepsilon}_1\end{bmatrix}$ whose covariance matrix is diagonal. It follows that,

$$\begin{aligned}
\hat{\mathbf{x}}_{1/1} &= \text{Proj}[\mathbf{x}_1|Y_1] = \text{Proj}[\mathbf{x}_1|\boldsymbol{\varepsilon}_0, \boldsymbol{\varepsilon}_1] = \bar{\mathbf{x}}_1 + E\Big[\mathbf{x}_1[\boldsymbol{\varepsilon}_0^T, \boldsymbol{\varepsilon}_1^T]\Big]\begin{bmatrix} E[\boldsymbol{\varepsilon}_0\boldsymbol{\varepsilon}_0^T] & 0 \\ 0 & E[\boldsymbol{\varepsilon}_1\boldsymbol{\varepsilon}_1^T]\end{bmatrix}^{-1}\begin{bmatrix}\boldsymbol{\varepsilon}_0 \\ \boldsymbol{\varepsilon}_1\end{bmatrix} \\
&= \bar{\mathbf{x}}_1 + E[\mathbf{x}_1\boldsymbol{\varepsilon}_0]E[\boldsymbol{\varepsilon}_0\boldsymbol{\varepsilon}_0^T]^{-1}\boldsymbol{\varepsilon}_0 + E[\mathbf{x}_1\boldsymbol{\varepsilon}_1]E[\boldsymbol{\varepsilon}_1\boldsymbol{\varepsilon}_1^T]^{-1}\boldsymbol{\varepsilon}_1
\end{aligned} \tag{13.3.12}$$

The first two terms are recognized from Eq. (13.3.6) to be the predicted estimate $\hat{\mathbf{x}}_{1/0}$. Therefore, we have,

$$\hat{\mathbf{x}}_{1/1} = \hat{\mathbf{x}}_{1/0} + E[\mathbf{x}_1\boldsymbol{\varepsilon}_1]E[\boldsymbol{\varepsilon}_1\boldsymbol{\varepsilon}_1^T]^{-1}\boldsymbol{\varepsilon}_1 = \hat{\mathbf{x}}_{1/0} + G_1\boldsymbol{\varepsilon}_1 \tag{13.3.13}$$

Since $\boldsymbol{\varepsilon}_1 \perp \boldsymbol{\varepsilon}_0$, we have $E[\hat{\mathbf{x}}_{1/0}\boldsymbol{\varepsilon}_1^T] = 0$, and using (13.3.9) we may write:

$$E[\mathbf{x}_1\boldsymbol{\varepsilon}_1] = E[(\mathbf{x}_1 - \hat{\mathbf{x}}_{1/0})\boldsymbol{\varepsilon}_1^T] = E[\mathbf{e}_{1/0}\boldsymbol{\varepsilon}_1^T] = E[\mathbf{e}_{1/0}(\mathbf{e}_{1/0}^T C_1^T + \mathbf{v}_1^T)] = P_{1/0}C_1^T$$

Thus, the Kalman gain for the filtered estimate is:

$$G_1 = E[\mathbf{x}_1\boldsymbol{\varepsilon}_1]E[\boldsymbol{\varepsilon}_1\boldsymbol{\varepsilon}_1^T]^{-1} = P_{1/0}C_1^T D_1^{-1}$$

The corresponding error covariance matrix can be obtained from Eq. (13.3.2), but perhaps a faster way is to argue as follows. Using Eq. (13.3.13), we have

$$\mathbf{e}_{1/1} = \mathbf{x}_1 - \hat{\mathbf{x}}_{1/1} = \mathbf{x}_1 - \hat{\mathbf{x}}_{1/0} - G_1\boldsymbol{\varepsilon}_1 = \mathbf{e}_{1/0} - G_1\boldsymbol{\varepsilon}_1\,, \quad \text{or,}$$

$$\mathbf{e}_{1/0} = \mathbf{e}_{1/1} + G_1\boldsymbol{\varepsilon}_1 \tag{13.3.14}$$

The orthogonality conditions for the estimate $\hat{\mathbf{x}}_{1/1}$ are $E[\mathbf{e}_{1/1}\boldsymbol{\varepsilon}_0^T] = E[\mathbf{e}_{1/1}\boldsymbol{\varepsilon}_1^T] = 0$. Thus, the two terms on the right-hand-side of (13.3.14) are orthogonal and we obtain the covariances:

$$E[\mathbf{e}_{1/0}\mathbf{e}_{1/0}^T] = E[\mathbf{e}_{1/1}\mathbf{e}_{1/1}^T] + G_1E[\boldsymbol{\varepsilon}_1\boldsymbol{\varepsilon}_1^T]G_1^T\,, \quad \text{or,}$$

$$P_{1/0} = P_{1/1} + G_1D_1G_1^T\,, \quad \text{or,}$$

$$P_{1/1} = P_{1/0} - G_1D_1G_1^T = P_{1/0} - P_{1/0}C_1^TD_1^{-1}C_1P_{1/0} \tag{13.3.15}$$

To complete the $n = 1$ steps, we must predict $\mathbf{x}_2$ from $Y_1 = \{\mathbf{y}_0, \mathbf{y}_1\} = \{\boldsymbol{\varepsilon}_0, \boldsymbol{\varepsilon}_1\}$. From the state equation $\mathbf{x}_2 = A_1\mathbf{x}_1 + \mathbf{w}_1$, we have:

$$\begin{aligned}
&\hat{\mathbf{x}}_{2/1} = \text{Proj}[\mathbf{x}_2|Y_1] = \text{Proj}[A_1\mathbf{x}_1 + \mathbf{w}_1|Y_1] = A_1\hat{\mathbf{x}}_{1/1} = A_1(\hat{\mathbf{x}}_{1/0} + G_1\boldsymbol{\varepsilon}_1) = A_1\hat{\mathbf{x}}_{1/0} + K_1\boldsymbol{\varepsilon}_1\\
&\mathbf{e}_{2/1} = \mathbf{x}_2 - \hat{\mathbf{x}}_{2/1} = A_1(\mathbf{x}_1 - \hat{\mathbf{x}}_{1/1}) + \mathbf{w}_1 = A_1\mathbf{e}_{1/1} + \mathbf{w}_1\\
&P_{2/1} = E[\mathbf{e}_{2/1}\mathbf{e}_{2/1}^T] = A_1E[\mathbf{e}_{1/1}\mathbf{e}_{1/1}^T]A_1^T + Q_1 = A_1P_{1/1}A_1^T + Q_1
\end{aligned}$$

where we defined $K_1 = A_1G_1$ and used the fact that $\mathbf{w}_1$ is independent of $\mathbf{x}_1$ and $\hat{\mathbf{x}}_{1/1}$, since the latter depends only on $\bar{\mathbf{x}}_0, \mathbf{y}_0, \mathbf{y}_1$. We collect the results for $n = 1$ together:

$$\boxed{\begin{aligned}
D_1 &= C_1P_{1/0}C_1^T + R_1\\
G_1 &= P_{1/0}C_1^TD_1^{-1}\\
K_1 &= A_1G_1 = A_1P_{1/0}C_1^TD_1^{-1}\\
\hat{\mathbf{y}}_{1/0} &= C_1\hat{\mathbf{x}}_{1/0}\\
\boldsymbol{\varepsilon}_1 &= \mathbf{y}_1 - \hat{\mathbf{y}}_{1/0} = \mathbf{y}_1 - C_1\hat{\mathbf{x}}_{1/0}\\
\hat{\mathbf{x}}_{1/1} &= \hat{\mathbf{x}}_{1/0} + G_1\boldsymbol{\varepsilon}_1\\
P_{1/1} &= P_{1/0} - G_1D_1G_1^T\\
\hat{\mathbf{x}}_{2/1} &= A_1\hat{\mathbf{x}}_{1/1} = A_1\hat{\mathbf{x}}_{1/0} + K_1\boldsymbol{\varepsilon}_1\\
P_{2/1} &= A_1P_{1/1}A_1^T + Q_1
\end{aligned}}$$

At the nth time instant, we assume that we have already constructed the orthogonalized basis of zero-mean innovations up to time $n-1$, i.e.,

$$\begin{aligned}
Y_{n-1} &= \{\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_{n-1}\} = \{\boldsymbol{\varepsilon}_0, \boldsymbol{\varepsilon}_1, \dots, \boldsymbol{\varepsilon}_{n-1}\}\\
E[\boldsymbol{\varepsilon}_i\boldsymbol{\varepsilon}_j^T] &= D_i\delta_{ij}\,, \quad 0 \le i,j \le n-1
\end{aligned}$$

Then, the optimal prediction of $\mathbf{x}_n$ based on Y_{n-1} will be:

$$\hat{\mathbf{x}}_{n/n-1} = \text{Proj}[\mathbf{x}_n|Y_{n-1}] = \bar{\mathbf{x}}_n + \sum_{i=0}^{n-1} E[\mathbf{x}_n\boldsymbol{\varepsilon}_i^T]E[\boldsymbol{\varepsilon}_i\boldsymbol{\varepsilon}_i^T]^{-1}\boldsymbol{\varepsilon}_i \tag{13.3.16}$$

Defining $\boldsymbol{\varepsilon}_n = \mathbf{y}_n - \hat{\mathbf{y}}_{n/n-1} = \mathbf{y}_n - C_n\hat{\mathbf{x}}_{n/n-1}$, we obtain,

$$\boldsymbol{\varepsilon}_n = \mathbf{y}_n - C_n\hat{\mathbf{x}}_{n/n-1} = C_n(\mathbf{x}_n - \hat{\mathbf{x}}_{n/n-1}) + \mathbf{v}_n = C_n\mathbf{e}_{n/n-1} + \mathbf{v}_n \tag{13.3.17}$$

Therefore, $E[\boldsymbol{\varepsilon}_n] = 0$, and since $\mathbf{v}_n \perp \mathbf{e}_{n/n-1}$ (because $\mathbf{e}_{n/n-1}$ depends only on $\mathbf{x}_0, \dots, \mathbf{x}_n$ and $\mathbf{y}_0, \dots, \mathbf{y}_{n-1}$), we have:

$$D_n = E[\boldsymbol{\varepsilon}_n\boldsymbol{\varepsilon}_n^T] = C_nP_{n/n-1}C_n^T + R_n \tag{13.3.18}$$

From the optimality of $\hat{\mathbf{x}}_{n/n-1}$, we have the orthogonality property $\mathbf{e}_{n/n-1} \perp \boldsymbol{\varepsilon}_i$, or, $E[\mathbf{e}_{n/n-1}\boldsymbol{\varepsilon}_i^T] = 0$, for $i = 0, 1, \dots, n-1$, and since also $\mathbf{v}_n \perp \boldsymbol{\varepsilon}_i$, we conclude that the constructed $\boldsymbol{\varepsilon}_n$ will be orthogonal to all the previous $\boldsymbol{\varepsilon}_i$, i.e.,

$$E[\boldsymbol{\varepsilon}_n\boldsymbol{\varepsilon}_i^T] = 0\,, \quad i = 0, 1, \dots, n-1$$

Thus, we may enlarge the orthogonalized basis of the observation subspace to time n:

$$\begin{aligned}
Y_n &= \{\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_{n-1}, \mathbf{y}_n\} = \{\boldsymbol{\varepsilon}_0, \boldsymbol{\varepsilon}_1, \dots, \boldsymbol{\varepsilon}_{n-1}, \boldsymbol{\varepsilon}_n\}\\
E[\boldsymbol{\varepsilon}_i\boldsymbol{\varepsilon}_j^T] &= D_i\delta_{ij}\,, \quad 0 \le i,j \le n
\end{aligned}$$

We note also that the definition $\hat{\mathbf{y}}_{n/n-1} = C_n\hat{\mathbf{x}}_{n/n-1}$ is equivalent to the conventional Gram-Schmidt construction process defined in Chap. 1, that is, starting with,

$$\begin{aligned}
\hat{\mathbf{y}}_{n/n-1} &= \text{Proj}[\mathbf{y}_n|Y_{n-1}] = \bar{\mathbf{y}}_n + \sum_{i=0}^{n-1} E[\mathbf{y}_n\boldsymbol{\varepsilon}_i^T]E[\boldsymbol{\varepsilon}_i\boldsymbol{\varepsilon}_i^T]^{-1}\boldsymbol{\varepsilon}_i\\
\boldsymbol{\varepsilon}_n &= \mathbf{y}_n - \hat{\mathbf{y}}_{n/n-1}
\end{aligned}$$

then, we may justify the relationship, $\hat{\mathbf{y}}_{n/n-1} = C_n\hat{\mathbf{x}}_{n/n-1}$. Indeed, since, $\mathbf{y}_n = C_n\mathbf{x}_n + \mathbf{v}_n$, we have, $\bar{\mathbf{y}}_n = C_n\bar{\mathbf{x}}_n$, and using Eq. (13.3.16), we obtain:

$$\begin{aligned}
\hat{\mathbf{y}}_{n/n-1} &= \bar{\mathbf{y}}_n + \sum_{i=0}^{n-1} E[\mathbf{y}_n\boldsymbol{\varepsilon}_i^T]E[\boldsymbol{\varepsilon}_i\boldsymbol{\varepsilon}_i^T]^{-1}\boldsymbol{\varepsilon}_i\\
&= C_n\bar{\mathbf{x}}_n + \sum_{i=0}^{n-1} E[(C_n\mathbf{x}_n + \mathbf{v}_n)\boldsymbol{\varepsilon}_i^T]E[\boldsymbol{\varepsilon}_i\boldsymbol{\varepsilon}_i^T]^{-1}\boldsymbol{\varepsilon}_i\\
&= C_n\left[\bar{\mathbf{x}}_n + \sum_{i=0}^{n-1} E[\mathbf{x}_n\boldsymbol{\varepsilon}_i^T]E[\boldsymbol{\varepsilon}_i\boldsymbol{\varepsilon}_i^T]^{-1}\boldsymbol{\varepsilon}_i\right] = C_n\hat{\mathbf{x}}_{n/n-1}
\end{aligned}$$

Next, we consider the updated estimate of $\mathbf{x}_n$ based on Y_n and given in terms of the innovations sequence:

$$\hat{\mathbf{x}}_{n/n} = \text{Proj}[\mathbf{x}_n|Y_n] = \bar{\mathbf{x}}_n + \sum_{i=0}^{n} E[\mathbf{x}_n\boldsymbol{\varepsilon}_i^T]E[\boldsymbol{\varepsilon}_i\boldsymbol{\varepsilon}_i^T]^{-1}\boldsymbol{\varepsilon}_i \tag{13.3.19}$$

It follows that, for $n \ge 1$,

$$\hat{\mathbf{x}}_{n/n} = \hat{\mathbf{x}}_{n/n-1} + E[\mathbf{x}_n\boldsymbol{\varepsilon}_n^T]E[\boldsymbol{\varepsilon}_n\boldsymbol{\varepsilon}_n^T]^{-1}\boldsymbol{\varepsilon}_n = \hat{\mathbf{x}}_{n/n-1} + G_n\boldsymbol{\varepsilon}_n \tag{13.3.20}$$

Because $\boldsymbol{\varepsilon}_n \perp \boldsymbol{\varepsilon}_i$, $i = 0, 1, \dots, n-1$, we have $E[\hat{\mathbf{x}}_{n/n-1}\boldsymbol{\varepsilon}_n^T] = 0$, which implies,

$$\begin{aligned} E[\mathbf{x}_n\boldsymbol{\varepsilon}_n^T] &= E[(\mathbf{x}_n - \hat{\mathbf{x}}_{n/n-1})\boldsymbol{\varepsilon}_n^T] = E[\mathbf{e}_{n/n-1}\boldsymbol{\varepsilon}_n^T] = E[\mathbf{e}_{n/n-1}(C_n\mathbf{e}_{n/n-1} + \mathbf{v}_n)^T] \\ &= E[\mathbf{e}_{n/n-1}\mathbf{e}_{n/n-1}^T]C_n^T = P_{n/n-1}C_n^T \end{aligned}$$

Thus, the Kalman gain will be:

$$G_n = E[\mathbf{x}_n\boldsymbol{\varepsilon}_n^T]E[\boldsymbol{\varepsilon}_n\boldsymbol{\varepsilon}_n^T]^{-1} = P_{n/n-1}C_n^TD_n^{-1} = P_{n/n-1}C_n^T\big[C_nP_{n/n-1}C_n^T + R_n\big]^{-1} \tag{13.3.21}$$

The estimation errors $\mathbf{e}_{n/n}$ and $\mathbf{e}_{n/n-1}$ are related by,

$$\mathbf{e}_{n/n} = \mathbf{x}_n - \hat{\mathbf{x}}_{n/n} = \mathbf{x}_n - \hat{\mathbf{x}}_{n/n-1} - G_n\boldsymbol{\varepsilon}_n = \mathbf{e}_{n/n-1} - G_n\boldsymbol{\varepsilon}_n\,, \quad \text{or,}$$

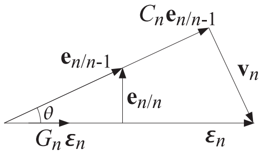
$$\mathbf{e}_{n/n-1} = \mathbf{e}_{n/n} + G_n\boldsymbol{\varepsilon}_n \tag{13.3.22}$$

and since one of the orthogonality conditions for $\hat{\mathbf{x}}_{n/n}$ is $E[\mathbf{e}_{n/n}\boldsymbol{\varepsilon}_n^T] = 0$, the two terms in the right-hand side will be orthogonal, which leads to the covariance relation:

$$P_{n/n-1} = P_{n/n} + G_nD_nG_n^T\,, \quad \text{or,}$$

$$P_{n/n} = P_{n/n-1} - G_nD_nG_n^T = P_{n/n-1} - P_{n/n-1}C_n^TD_n^{-1}C_nP_{n/n-1} \tag{13.3.23}$$

A nice geometrical interpretation of Eqs. (13.3.17) and (13.3.22) was given by Kronhamn [867] and is depicted below (see also Chap. 11):


$C_n\mathbf{e}_{n/n\text{-}1}$
$\mathbf{e}_{n/n\text{-}1}$
$\mathbf{v}_n$
$\mathbf{e}_{n/n}$
θ
$G_n\boldsymbol{\varepsilon}_n$
$\boldsymbol{\varepsilon}_n$

The similarity of the two orthogonal triangles leads to Eq. (13.3.21). Indeed, for the scalar case, the lengths of the triangle sides are given by the square roots of the covariances, e.g., $\sqrt{E[\varepsilon_n^2]} = \sqrt{D_n}$. Then, the Pythagorean theorem and the similarity of the triangles give,

$$\begin{aligned} &E[\varepsilon_n^2] = C_n^2E[e_{n/n-1}^2] + E[v_n^2] \quad \Rightarrow \quad D_n = C_nP_{n/n-1}C_n + R_n \\ &E[e_{n/n-1}^2] = E[e_{n/n}^2] + G_n^2E[\varepsilon_n^2] \quad \Rightarrow \quad P_{n/n-1} = P_{n/n} + G_nD_nG_n \\ &\cos\theta = \frac{G_n\sqrt{D_n}}{\sqrt{P_{n/n-1}}} = \frac{C_n\sqrt{P_{n/n-1}}}{\sqrt{D_n}} \quad \Rightarrow \quad G_n = \frac{P_{n/n-1}C_n}{D_n} \\ &\sin\theta = \frac{\sqrt{P_{n/n}}}{\sqrt{P_{n/n-1}}} = \frac{\sqrt{R_n}}{\sqrt{D_n}} \quad \Rightarrow \quad P_{n/n-1}D_n^{-1} = P_{n/n}R_n^{-1} \quad \Rightarrow \quad \text{Eq. (13.2.5)} \end{aligned}$$

Finally, we determine the next predicted estimate, which may be obtained by using the state equation $\mathbf{x}_{n+1} = A_n\mathbf{x}_n + \mathbf{w}_n$, and noting that $E[\mathbf{x}_{n+1}\boldsymbol{\varepsilon}_i^T] = E[(A_n\mathbf{x}_n + \mathbf{w}_n)\boldsymbol{\varepsilon}_i^T] = A_nE[\mathbf{x}_n\boldsymbol{\varepsilon}_i^T]$, for $0 \le i \le n$. Then, using $\bar{\mathbf{x}}_{n+1} = A_n\bar{\mathbf{x}}_n$, we find,

$$\begin{aligned} \hat{\mathbf{x}}_{n+1/n} &= \bar{\mathbf{x}}_{n+1} + \sum_{i=0}^{n} E[\mathbf{x}_{n+1}\boldsymbol{\varepsilon}_i^T]E[\boldsymbol{\varepsilon}_i\boldsymbol{\varepsilon}_i^T]^{-1}\boldsymbol{\varepsilon}_i \\ &= A_n\left[\bar{\mathbf{x}}_n + \sum_{i=0}^{n} E[\mathbf{x}_n\boldsymbol{\varepsilon}_i^T]E[\boldsymbol{\varepsilon}_i\boldsymbol{\varepsilon}_i^T]^{-1}\boldsymbol{\varepsilon}_i\right] = A_n\hat{\mathbf{x}}_{n/n} = A_n\big[\hat{\mathbf{x}}_{n/n-1} + G_n\boldsymbol{\varepsilon}_n\big] \\ &= A_n\hat{\mathbf{x}}_{n/n-1} + K_n\boldsymbol{\varepsilon}_n = (A_n - K_nC_n)\hat{\mathbf{x}}_{n/n-1} + K_n\mathbf{y}_n \end{aligned}$$

where we defined $K_n = A_nG_n$. The error covariance is obtained by noting that,

$$\mathbf{e}_{n+1/n} = \mathbf{x}_{n+1} - \hat{\mathbf{x}}_{n+1/n} = A_n\mathbf{e}_{n/n} + \mathbf{w}_n$$

and because $\mathbf{w}_n$ is orthogonal to $\mathbf{e}_{n/n}$, this leads to

$$P_{n+1/n} = E[\mathbf{e}_{n+1/n}\mathbf{e}_{n+1/n}^T] = A_nE[\mathbf{e}_{n/n}\mathbf{e}_{n/n}^T]A_n^T + E[\mathbf{w}_n\mathbf{w}_n^T] = A_nP_{n/n}A_n^T + Q_n$$

This completes the operations at the nth time step. The various equivalent expressions in Eqs. (13.2.4) and (13.2.5) are straightforward to derive. The Joseph form is useful because it guarantees the numerical positive-definiteness of the error covariance matrix. The information form is a consequence of the matrix inversion lemma. It can be shown directly as follows. To simplify the notation, we write the covariance update as,

$$\hat{P} = P - GDG^T = P - PC^TD^{-1}CP\,, \quad D = R + CPC^T$$

Multiply from the right by P^{-1} and from the left by $\hat{P}^{-1}$ to get:

$$P^{-1} = \hat{P}^{-1} - \hat{P}^{-1}PC^TD^{-1}C \tag{13.3.24}$$

Next, multiply from the right by PC^T to get:

$$\begin{aligned} C^T &= \hat{P}^{-1}PC^T - \hat{P}^{-1}PC^TD^{-1}CPC^T = \hat{P}^{-1}PC^T(I - D^{-1}CPC^T) \\ &= \hat{P}^{-1}PC^TD^{-1}(D - CPC^T) = \hat{P}^{-1}PC^TD^{-1}R \end{aligned}$$

which gives (assuming that R^{-1} exists):

$$C^TR^{-1} = \hat{P}^{-1}PC^TD^{-1}$$

Inserting this result into Eq. (13.3.24), we obtain

$$P^{-1} = \hat{P}^{-1} - \hat{P}^{-1}PC^TD^{-1}C = \hat{P}^{-1} - C^TR^{-1}C \quad \Rightarrow \quad \hat{P}^{-1} = P^{-1} + C^TR^{-1}C$$

and also obtain,

$$\hat{P}C^TR^{-1} = PC^TD^{-1} = G$$

Since the information form works with the inverse covariances, to complete the operations at each time step, we need to develop a recursion for the inverse $P^{-1}_{n+1/n}$ in terms of $P^{-1}_{n/n}$. Denoting $P_{n+1/n}$ by P_{next}, we have

$$P_{\text{next}} = A\hat{P}A^T + Q$$

If we assume that A^{-1} and Q^{-1} exist, then the application of the matrix inversion lemma to this equation allows us to rewrite it in terms of the matrix inverses:

$$P^{-1}_{\text{next}} = A^{-T}\hat{P}^{-1}A^{-1} - A^{-T}\hat{P}^{-1}A^{-1}\big[Q^{-1} + A^{-T}\hat{P}^{-1}A^{-1}\big]^{-1}A^{-T}\hat{P}^{-1}A^{-1}$$

To summarize, the information form of the Kalman filter is as follows:

$$\begin{aligned}
&P^{-1}_{n/n} = P^{-1}_{n/n-1} + C_n^T R_n^{-1} C_n \\
&P^{-1}_{n/n}\hat{\mathbf{x}}_{n/n} = P^{-1}_{n/n-1}\hat{\mathbf{x}}_{n/n-1} + C_n^T R_n^{-1}\mathbf{y}_n \\
&P^{-1}_{n+1/n} = A_n^{-T}P^{-1}_{n/n}A_n^{-1} - A_n^{-T}P^{-1}_{n/n}A_n^{-1}\big[Q_n^{-1} + A_n^{-T}P^{-1}_{n/n}A_n^{-1}\big]^{-1}A_n^{-T}P^{-1}_{n/n}A_n^{-1}
\end{aligned} \tag{13.3.25}$$

13.4 Forecasting and Missing Observations

The problem of forecasting ahead from the current time sample n and the problem of missing observations are similar. Suppose one has at hand the estimate $\hat{\mathbf{x}}_{n/n}$ based on $Y_n = \{\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_n\}$. Then the last part of the Kalman filtering algorithm (13.2.2) produces the prediction of $\mathbf{x}_{n+1}$ based on Y_n,

$$\hat{\mathbf{x}}_{n+1/n} = A_n\hat{\mathbf{x}}_{n/n}$$

This prediction can be continued to future times. For example, since $\mathbf{x}_{n+2} = A_{n+1}\mathbf{x}_{n+1} + \mathbf{w}_{n+1}$ and $\mathbf{w}_{n+1}$ is independent of Y_n, we have:

$$\begin{aligned}
\hat{\mathbf{x}}_{n+2/n} &= \text{Proj}\big[\mathbf{x}_{n+2}|Y_n\big] \\
&= \text{Proj}\big[A_{n+1}\mathbf{x}_{n+1} + \mathbf{w}_{n+1}|Y_n\big] \\
&= A_{n+1}\hat{\mathbf{x}}_{n+1/n} = A_{n+1}A_n\hat{\mathbf{x}}_{n/n} = \Phi_{n+2,n}\hat{\mathbf{x}}_{n/n}
\end{aligned}$$

and so on. Thus, the prediction of $\mathbf{x}_{n+p}$ based on Y_n is for $p \ge 1$,

$$\hat{\mathbf{x}}_{n+p/n} = \Phi_{n+p,n}\hat{\mathbf{x}}_{n/n} \tag{13.4.1}$$

The corresponding error covariance is found by applying (13.1.8), that is,

$$\mathbf{x}_{n+p} = \Phi_{n+p,n}\,\mathbf{x}_n + \sum_{k=n+1}^{n+p}\Phi_{n+p,k}\mathbf{w}_{k-1}\,, \quad p \ge 1 \tag{13.4.2}$$

which in conjunction with (13.4.1), gives for the forecast error $\mathbf{e}_{n+p/n} = \mathbf{x}_n - \hat{\mathbf{x}}_{n+p/n}$,

$$\mathbf{e}_{n+p/n} = \Phi_{n+p,n}\,\mathbf{e}_{n/n} + \sum_{k=n+1}^{n+p}\Phi_{n+p,k}\mathbf{w}_{k-1} \tag{13.4.3}$$

which implies for its covariance:

$$P_{n+p/n} = \Phi_{n+p,n}P_{n/n}\Phi^T_{n+p,n} + \sum_{k=n+1}^{n+p}\Phi_{n+p,k}Q_{k-1}\Phi^T_{n+p,k} \tag{13.4.4}$$

Eqs. (13.4.1) and (13.4.4) apply also in the case when a group of observations, say, $\{\mathbf{y}_{n+1}, \mathbf{y}_{n+2}, \dots, \mathbf{y}_{n+p-1}\}$, are missing. In such case, one simply predicts ahead from time n using the observation set Y_n. Once the observation $\mathbf{y}_{n+p}$ becomes available, one may resume the usual algorithm using the initial values $\hat{\mathbf{x}}_{n+p/n}$ and $P_{n+p/n}$.

This procedure is equivalent to setting, in the algorithm (13.2.2), $G_{n+i} = 0$ and $\boldsymbol{\varepsilon}_{n+i} = 0$ over the period of the missing observations, $i = 1, 2, \dots, p-1$, that is, ignoring the measurement updates, but not the time updates.

In some presentations of the Kalman filtering algorithm, it is assumed that the observations are available from $n \ge 1$, i.e., $Y_n = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_n\} = \{\boldsymbol{\varepsilon}_1, \boldsymbol{\varepsilon}_2, \dots, \boldsymbol{\varepsilon}_n\}$, and the algorithm is initialized at $\hat{\mathbf{x}}_{0/0} = \bar{\mathbf{x}}_0$ with $P_{0/0} = \Sigma_0$. We may view this as a case of a missing observation $\mathbf{y}_0$, and therefore, from the above rule, we may set $\boldsymbol{\varepsilon}_0 = 0$ and $G_0 = 0$, which leads to $\hat{\mathbf{x}}_{0/0} = \hat{\mathbf{x}}_{0/-1} = \bar{\mathbf{x}}_0$ and $P_{0/0} = P_{0/-1} = \Sigma_0$. The algorithm may be stated then somewhat differently, but equivalently, to Eq. (13.2.2):

Initialize at $n = 0$ by: $\hat{\mathbf{x}}_{0/0} = \bar{\mathbf{x}}_0$, $P_{0/0} = \Sigma_0$

At time $n \ge 1$, $\hat{\mathbf{x}}_{n-1/n-1}$, $P_{n-1/n-1}$, $\mathbf{y}_n$ are available,

$\hat{\mathbf{x}}_{n/n-1} = A_{n-1}\hat{\mathbf{x}}_{n-1/n-1}$	predicted estimate
$P_{n/n-1} = A_{n-1}P_{n-1/n-1}A^T_{n-1} + Q_{n-1}$	prediction error
$D_n = C_nP_{n/n-1}C_n^T + R_n$	innovations covariance
$G_n = P_{n/n-1}C_n^TD_n^{-1}$	Kalman gain for filtering
$\hat{\mathbf{y}}_{n/n-1} = C_n\hat{\mathbf{x}}_{n/n-1}$	predicted measurement
$\boldsymbol{\varepsilon}_n = \mathbf{y}_n - \hat{\mathbf{y}}_{n/n-1} = \mathbf{y}_n - C_n\hat{\mathbf{x}}_{n/n-1}$	innovations sequence
$\hat{\mathbf{x}}_{n/n} = \hat{\mathbf{x}}_{n/n-1} + G_n\boldsymbol{\varepsilon}_n$	filtered estimate
$P_{n/n} = P_{n/n-1} - G_nD_nG_n^T$	mean-square error

Go to time $n + 1$

13.5 Kalman Filter with Deterministic Inputs

A state/measurement model that has a deterministic input $\mathbf{u}_n$ in addition the noise input $\mathbf{w}_n$ can be formulated by,

$$\begin{aligned}
&\boxed{\begin{aligned}\mathbf{x}_{n+1} &= A_n\mathbf{x}_n + B_n\mathbf{u}_n + \mathbf{w}_n \\ \mathbf{y}_n &= C_n\mathbf{x}_n + \mathbf{v}_n\end{aligned}}
\quad
\begin{aligned}&\text{(state model)} \\ &\text{(measurement model)}\end{aligned}
\end{aligned} \tag{13.5.1}$$

As we mentioned earlier, this requires a minor modification of the algorithm (13.2.2), namely, replacing the time-update equation by that in Eq. (13.5.3) below. Using linear superposition, we may think of this model as two models, one driven by the white noise inputs, and the other by the deterministic input, that is,

$$\begin{aligned} \mathbf{x}_{n+1}^{(1)} &= A_n\mathbf{x}_n^{(1)} + \mathbf{w}_n \\ \mathbf{y}_n^{(1)} &= C_n\mathbf{x}_n^{(1)} + \mathbf{v}_n \end{aligned} \qquad \text{and} \qquad \begin{aligned} \mathbf{x}_{n+1}^{(2)} &= A_n\mathbf{x}_n^{(2)} + B_n\mathbf{u}_n \\ \mathbf{y}_n^{(2)} &= C_n\mathbf{x}_n^{(2)} \end{aligned} \tag{13.5.2}$$

If we adjust the initial conditions of the two systems to match that of (13.5.1), that is, $\mathbf{x}_0 = \mathbf{x}_0^{(1)} + \mathbf{x}_0^{(2)}$, then the solution of the system (13.5.1) will be the sum:

$$\mathbf{x}_n = \mathbf{x}_n^{(1)} + \mathbf{x}_n^{(2)}\,, \quad \mathbf{y}_n = \mathbf{y}_n^{(1)} + \mathbf{y}_n^{(2)}$$

System (2) is purely deterministic, and therefore, we have the estimates,

$$\begin{aligned} \hat{\mathbf{x}}_{n/n}^{(2)} &= \text{Proj}\big[\mathbf{x}_n^{(2)}|Y_n\big] = \mathbf{x}_n^{(2)} \\ \hat{\mathbf{x}}_{n/n-1}^{(2)} &= \text{Proj}\big[\mathbf{x}_n^{(2)}|Y_{n-1}\big] = \mathbf{x}_n^{(2)} \end{aligned}$$

and similarly, $\hat{\mathbf{y}}_{n/n-1}^{(2)} = \mathbf{y}_n^{(2)}$. For system (1), we may apply the Kalman filtering algorithm of Eq. (13.2.2). We note that

$$\boldsymbol{\varepsilon}_n = \mathbf{y}_n - \hat{\mathbf{y}}_{n/n-1} = \mathbf{y}_n^{(1)} + \mathbf{y}_n^{(2)} - \hat{\mathbf{y}}_{n/n-1}^{(1)} - \hat{\mathbf{y}}_{n/n-1}^{(2)} = \mathbf{y}_n^{(1)} - \hat{\mathbf{y}}_{n/n-1}^{(1)}$$

so that $D_n = D_n^{(1)}$. Similarly, we find,

$$\begin{aligned} \mathbf{e}_{n/n-1} &= \mathbf{x}_n - \hat{\mathbf{x}}_{n/n-1} = \mathbf{x}_n^{(1)} - \hat{\mathbf{x}}_{n/n-1}^{(1)} \quad &\Rightarrow \quad P_{n/n-1} = P_{n/n-1}^{(1)} \\ \mathbf{e}_{n/n} &= \mathbf{x}_n - \hat{\mathbf{x}}_{n/n} = \mathbf{x}_n^{(1)} - \hat{\mathbf{x}}_{n/n}^{(1)} \quad &\Rightarrow \quad P_{n/n} = P_{n/n}^{(1)} \end{aligned}$$

and similarly, $G_n = G_n^{(1)}$ and $K_n = K_n^{(1)}$. The measurement update equation remains the same, that is,

$$\hat{\mathbf{x}}_{n/n} = \hat{\mathbf{x}}_{n/n}^{(1)} + \mathbf{x}_n^{(2)} = \hat{\mathbf{x}}_{n/n-1}^{(1)} + G_n\boldsymbol{\varepsilon}_n + \mathbf{x}_n^{(2)} = \hat{\mathbf{x}}_{n/n-1} + G_n\boldsymbol{\varepsilon}_n$$

The only step of the algorithm (13.2.2) that changes is the time update equation:

$$\begin{aligned} \hat{\mathbf{x}}_{n+1/n} &= \hat{\mathbf{x}}_{n+1/n}^{(1)} + \hat{\mathbf{x}}_{n+1/n}^{(2)} = \hat{\mathbf{x}}_{n+1/n}^{(1)} + \mathbf{x}_{n+1}^{(2)} \\ &= \big[A_n\hat{\mathbf{x}}_{n/n-1}^{(1)} + K_n\boldsymbol{\varepsilon}_n\big] + \big[A_n\mathbf{x}_n^{(2)} + B_n\mathbf{u}_n\big] \\ &= A_n\big(\hat{\mathbf{x}}_{n/n-1}^{(1)} + \mathbf{x}_n^{(2)}\big) + B_n\mathbf{u}_n + K_n\boldsymbol{\varepsilon}_n \\ &= A_n\hat{\mathbf{x}}_{n/n-1} + B_n\mathbf{u}_n + K_n\boldsymbol{\varepsilon}_n = A_n\hat{\mathbf{x}}_{n/n} + B_n\mathbf{u}_n\,, \quad \text{or,} \end{aligned}$$

$$\boxed{\hat{\mathbf{x}}_{n+1/n} = A_n\hat{\mathbf{x}}_{n/n-1} + B_n\mathbf{u}_n + K_n\boldsymbol{\varepsilon}_n = A_n\hat{\mathbf{x}}_{n/n} + B_n\mathbf{u}_n} \tag{13.5.3}$$

13.6 Time-Invariant Models

In many applications, the model parameters $\{A_n, C_n, Q_n, R_n\}$ are constants in time, that is, $\{A, C, Q, R\}$, and the model takes the form:

$$\boxed{\begin{aligned} \mathbf{x}_{n+1} &= A\mathbf{x}_n + \mathbf{w}_n \\ \mathbf{y}_n &= C\mathbf{x}_n + \mathbf{v}_n \end{aligned}} \quad \begin{aligned} &\text{(state model)} \\ &\text{(measurement model)} \end{aligned} \tag{13.6.1}$$

The signals $\mathbf{w}_n, \mathbf{v}_n$ are again assumed to be mutually-independent, zero-mean, white-noise signals with known covariance matrices:

$$\boxed{\begin{aligned} E[\mathbf{w}_n\mathbf{w}_i^T] &= Q\delta_{ni} \\ E[\mathbf{v}_n\mathbf{v}_i^T] &= R\delta_{ni} \\ E[\mathbf{w}_n\mathbf{v}_i^T] &= 0 \end{aligned}} \tag{13.6.2}$$

The model is iterated starting at $n = 0$. The initial state vector $\mathbf{x}_0$ is assumed to be random and independent of $\mathbf{w}_n, \mathbf{v}_n$, but with a known mean $\bar{\mathbf{x}}_0 = E[\mathbf{x}_0]$ and covariance matrix $\Sigma_0 = E[(\mathbf{x}_0 - \bar{\mathbf{x}}_0)(\mathbf{x}_0 - \bar{\mathbf{x}}_0)^T]$. The Kalman filtering algorithm (13.2.2) then takes the form:

```
Initialize in time by: $\hat{\mathbf{x}}_{0/-1} = \bar{\mathbf{x}}_0$, $P_{0/-1} = \Sigma_0$
At time $n$, $\hat{\mathbf{x}}_{n/n-1}$, $P_{n/n-1}$, $\mathbf{y}_n$ are available,
    $D_n = CP_{n/n-1}C^T + R$                                      innovations covariance
    $G_n = P_{n/n-1}C^TD_n^{-1}$                                    Kalman gain for filtering
    $K_n = AG_n = AP_{n/n-1}C^TD_n^{-1}$                            Kalman gain for prediction
    $\hat{\mathbf{y}}_{n/n-1} = C\hat{\mathbf{x}}_{n/n-1}$                                   predicted measurement
    $\boldsymbol{\varepsilon}_n = \mathbf{y}_n - \hat{\mathbf{y}}_{n/n-1} = \mathbf{y}_n - C\hat{\mathbf{x}}_{n/n-1}$                  innovations sequence
Measurement update / correction:
    $\hat{\mathbf{x}}_{n/n} = \hat{\mathbf{x}}_{n/n-1} + G_n\boldsymbol{\varepsilon}_n$                            filtered estimate
    $P_{n/n} = P_{n/n-1} - G_nD_nG_n^T$                              estimaton error
Time update / prediction:
    $\hat{\mathbf{x}}_{n+1/n} = A\hat{\mathbf{x}}_{n/n} = A\hat{\mathbf{x}}_{n/n-1} + K_n\boldsymbol{\varepsilon}_n$           predicted estimate
    $P_{n+1/n} = AP_{n/n}A^T + Q$                                  prediction error
Go to time $n + 1$
```

(13.6.3)

Note also that Eqs. (13.2.12) become now,

$$\boxed{\begin{array}{ll} \hat{\mathbf{y}}_{n/n-1} = C\hat{\mathbf{x}}_{n/n-1} = CA\hat{\mathbf{x}}_{n-1/n-1} & \text{predicted measurement} \\ \boldsymbol{\varepsilon}_n = \mathbf{y}_n - \hat{\mathbf{y}}_{n/n-1} & \text{innovations sequence} \\ \hat{\mathbf{x}}_{n+1/n} = A\hat{\mathbf{x}}_{n/n-1} + K\boldsymbol{\varepsilon}_n & \text{predicted estimate} \\ \hat{\mathbf{x}}_{n/n} = A\hat{\mathbf{x}}_{n-1/n-1} + G\boldsymbol{\varepsilon}_n & \text{filtered estimate} \end{array}} \tag{13.6.4}$$

The MATLAB function, `kfilt.m`, implements the Kalman filtering algorithm of Eq. (13.6.3). It has usage:

```
[L,X,P,Xf,Pf] = kfilt(A,C,Q,R,Y,x0,S0);        % Kalman filtering
```

Its inputs are the state-space model parameters $\{A, C, Q, R\}$, the initial values $\bar{\mathbf{x}}_0$, Σ_0, and the observations $\mathbf{y}_n$, $0 \le n \le N$, arranged into an $r\times(N+1)$ matrix:

$$Y = \big[\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_n, \dots, \mathbf{y}_N\big]$$

The outputs are the predicted and filtered estimates arranged into $p\times(N+1)$ matrices:

$$X = \big[\hat{\mathbf{x}}_{0/-1},\, \hat{\mathbf{x}}_{1/0},\, \dots,\, \hat{\mathbf{x}}_{n/n-1},\, \dots,\, \hat{\mathbf{x}}_{N/N-1}\big]$$
$$X_f = \big[\hat{\mathbf{x}}_{0/0},\, \hat{\mathbf{x}}_{1/1},\, \dots,\, \hat{\mathbf{x}}_{n/n},\, \dots,\, \hat{\mathbf{x}}_{N/N}\big]$$

whose error covariance matrices are arranged into the $p\times p\times(N+1)$ three-dimensional arrays P, P_f, such that (in MATLAB notation):

$$\texttt{P(:,:,n+1)} = P_{n/n-1}\,, \quad \texttt{Pf(:,:,n+1)} = P_{n/n}\,, \quad 0 \le n \le N$$

The output L is the value of the negative-log-likelihood function calculated from Eq. (13.12.2).

Under certain conditions of stabilizability and detectability, the Kalman filter parameters $\{D_n, P_{n/n-1}, G_n, K_n\}$ converge to steady-state values $\{D, P, G, K\}$ such that P is unique and positive-semidefinite symmetric and the converged closed-loop state matrix $F = A - KC$ is stable, i.e., its eigenvalues are strictly inside the unit circle. The steady-state values are all given in terms of P, as follows:

$$\boxed{\begin{aligned} &D = CPC^T + R \\ &G = PC^TD^{-1} = [I + PC^TR^{-1}C]^{-1}PC^TR^{-1} \\ &K = AG = A[I + PC^TR^{-1}C]^{-1}PC^TR^{-1} \\ &F = A - KC = A[I + PC^TR^{-1}C]^{-1} \end{aligned}} \tag{13.6.5}$$

and P is determined as the unique positive-semidefinite symmetric solution of the so-called *discrete algebraic Riccati equation* (DARE), written in two alternative ways:

$$\boxed{\begin{aligned} &P = APA^T - APC^T(CPC^T + R)^{-1}CPA^T + Q \\ &P = A[I + PC^TR^{-1}C]^{-1}PA^T + Q \end{aligned}} \quad \text{(DARE)} \tag{13.6.6}$$

The required conditions are that the pair $[C, A]$ be completely detectable and the pair $[A, Q^{1/2}]$, completely stabilizable,[†] where $Q^{1/2}$ denotes a square root of the positive semidefinite matrix Q. Refs. [863,865] include a literature overview of various conditions for this and related weaker results. The convergence speed of the time-varying quantities to their steady-state values is determined essentially by the magnitude *square* of largest eigenvalue of the closed-loop matrix $F = A - KC$ (see for example [871,872]). If we define the eigenvalue radius $\rho = \max_i |\lambda_i|$, where λ_i are the eigenvalues of F, then a measure of the effective time constant is:

$$n_{\text{eff}} = \frac{\ln \epsilon}{\ln \rho^2} \tag{13.6.7}$$

[†] For definitions of complete stabilizability and detectability see [863], which is available online.

where ϵ is a small user-defined quantity, such as $\epsilon = 10^{-2}$ for the 40-dB time constant, or $\epsilon = 10^{-3}$ for the 60-dB time constant.

The MATLAB function, `dare()`, in the control systems toolbox allows the calculation of the solution P and Kalman gain K, with usage:

```
[P,L,KT] = dare(A', C', Q, R);
```

where the output `P` is the required solution, `KT` is the transposed of the gain K, and `L` is the vector of eigenvalues of the closed-loop matrix $F = A - KC$. The syntax requires that the input matrices A, C be entered in transposed form.

Example 13.6.1: *Benchmark Example.* Solution methods of the DARE are reviewed in [877]. The following two-dimensional model is a benchmark example from the collection [878]:

$$A = \begin{bmatrix} 4 & -4.5 \\ 3 & -3.5 \end{bmatrix}, \quad C = [1, -1]\,, \quad Q = \begin{bmatrix} 9 & 6 \\ 6 & 4 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \end{bmatrix}[3, 2]\,, \quad R = 1$$

The MATLAB call,

```
[P,L,K_tr] = dare(A', C', Q, R);
```

returns the values:

$$P = \begin{bmatrix} 14.5623 & 9.7082 \\ 9.7082 & 6.4721 \end{bmatrix}, \quad K = K_{\text{tr}}^T = \begin{bmatrix} 1.8541 \\ 1.2361 \end{bmatrix}, \quad L = \begin{bmatrix} 0.3820 \\ -0.5000 \end{bmatrix}$$

These agree with the exact solutions:

$$P = \frac{1+\sqrt{5}}{2}\begin{bmatrix} 9 & 6 \\ 6 & 4 \end{bmatrix}, \quad K = \frac{\sqrt{5}-1}{2}\begin{bmatrix} 3 \\ 2 \end{bmatrix}, \quad L = \begin{bmatrix} (3-\sqrt{5})/2 \\ -0.5 \end{bmatrix}$$

This example does satisfy the stabilizability and detectability requirements for convergence, even though the model itself is uncontrollable and unobservable. Indeed, using the square root factor $\mathbf{q} = [3, 2]^T$ for Q where $Q = \mathbf{q}\mathbf{q}^T$, we see that the controllability and observability matrices are rank defective:

$$[\mathbf{q},\ A\mathbf{q}] = \begin{bmatrix} 3 & 3 \\ 2 & 2 \end{bmatrix}, \quad \begin{bmatrix} C \\ CA \end{bmatrix} = \begin{bmatrix} 1 & -1 \\ 1 & -1 \end{bmatrix}$$

The largest eigenvalue of the matrix F is $\lambda_1 = -0.5$, which leads to an estimated 40-dB time constant of $n_{\text{eff}} = \log(0.01)/\log\big((0.5)^2\big) = 3.3$. The time-varying prediction-error matrix $P_{n/n-1}$ can be given in closed form. Using the methods of [871,872], we find:

$$P_{n/n-1} = P + F^n M_n F^{Tn}\,, \quad n \ge 0 \tag{13.6.8}$$

where, F^n is the nth power of F and can be expressed in terms of its eigenvalues, as follows,

$$F^n = \begin{bmatrix} 3\lambda_2^n - 2\lambda_1^n & 3\lambda_1^n - 3\lambda_2^n \\ 2\lambda_2^n - 2\lambda_1^n & 3\lambda_1^n - 2\lambda_2^n \end{bmatrix}, \quad \lambda_1 = -0.5\,, \quad \lambda_2 = \frac{3-\sqrt{5}}{2}$$

This is obtained from the eigenvalue decomposition:

$$F = V\Lambda V^{-1} = \begin{bmatrix} 1 & 1.5 \\ 1 & 1 \end{bmatrix}\begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}\begin{bmatrix} -2 & 3 \\ 2 & -2 \end{bmatrix} \quad\Rightarrow\quad F^n = V\Lambda^n V^{-1}$$

The matrix M_n is given by:

$$M_n = \frac{1}{ad - bc + (a+b+c+d)c_n}\begin{bmatrix} c_n + d & c_n - b \\ c_n - c & c_n + a \end{bmatrix}, \quad c_n = \frac{1}{\sqrt{5}}(1 - \lambda_2^{2n})$$

and the numbers a, b, c, d are related to an arbitrary initial value $P_{0/-1}$ via the definition:

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} = (P_{0/-1} - P)^{-1} \quad\Rightarrow\quad P_{0/-1} - P = \frac{1}{ad - bc}\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}$$

provided that the indicated matrix inverse exists. We note that at $n = 0$, $M_0 = P_{0/-1} - P$ and the above solution for $P_{n/n-1}$ correctly accounts for the initial condition. It can be verified that Eq. (13.6.8) is the solution to the difference equation with the prescribed initial value:

$$P_{n+1/n} = A\big[P_{n/n-1} - P_{n/n-1}C^T(CP_{n/n-1}C^T + R)^{-1}CP_{n/n-1}\big]A^T + Q\,, \quad \text{or,}$$

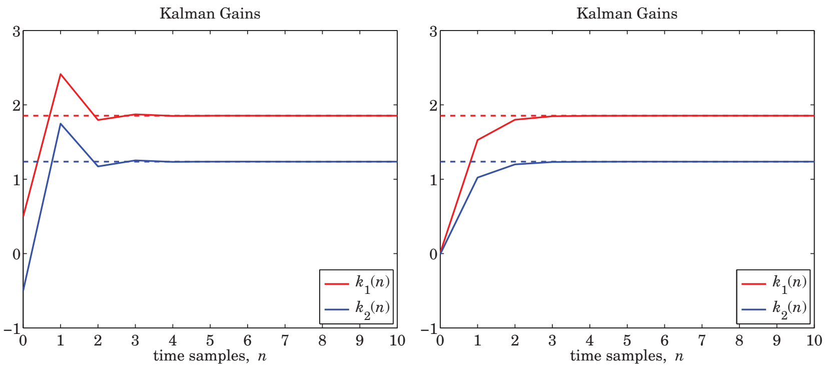
$$P_{n+1/n} = A\big[I + P_{n/n-1}C^T R^{-1}C\big]^{-1}P_{n/n-1}A^T + Q$$

Since $|\lambda_2| < |\lambda_1|$, it is evident from the above solution that the convergence time-constant is determined by $|\lambda_1|^2$. As $P_{n/n-1} \to P$, so does the Kalman gain:

$$K_n = AP_{n/n-1}C^T(CP_{n/n-1}C^T + R)^{-1} \longrightarrow K = APC^T(CPC^T + R)^{-1}$$

The figure below plots the two components of the gain vector $K_n = [k_1(n), k_2(n)]^T$ versus time n, for the two choices of initial conditions:

$$P_{0/-1} = \begin{bmatrix} 100 & 0 \\ 0 & 100 \end{bmatrix}, \quad P_{0/-1} = \begin{bmatrix} 1/100 & 0 \\ 0 & 1/100 \end{bmatrix}$$



We note that we did not initialize to $P_{0/-1} = 0$ because P is rank defective and the initial matrix $M_0 = -P$ would not be invertible. □

Example 13.6.2: *Local Level Model.* Consider the local-level model of Example 13.1.1 (see also Problem 11.13),

$$x_{n+1} = x_n + w_n$$

$$y_n = x_n + v_n$$

with $Q = \sigma_w^2$ and $R = \sigma_v^2$. The Kalman filtering algorithm (13.6.3) has the parameters $A = 1$, $C = 1$, and satisfies the Riccati difference and algebraic equations:

$$P_{n+1/n} = \frac{P_{n/n-1}R}{P_{n/n-1} + R} + Q \quad\Rightarrow\quad P = \frac{PR}{P+R} + Q \quad\Rightarrow\quad \frac{P^2}{P+R} = Q$$

and has time-varying and steady gains:

$$K_n = G_n = \frac{P_{n/n-1}}{P_{n/n-1} + R} \quad\Rightarrow\quad K = G = \frac{P}{P+R}$$

and closed-loop transition matrix:

$$F_n = 1 - K_n = \frac{R}{P_{n/n-1} + R} \quad\Rightarrow\quad F = 1 - K = \frac{R}{P+R}$$

The positive solution of the algebraic Riccati equation is:

$$P = \frac{Q}{2} + \sqrt{QR + \frac{Q^2}{4}}$$

The convergence properties to the steady values depend on the closed-loop matrix (here scalar) F. Again, using the methods of [871,872], we find the exact solutions:

$$P_{n/n-1} = P + \frac{(P_0 - P)F^{2n}}{1 + (P_0 - P)S(1 - F^{2n})}, \quad n \ge 0, \quad S = \frac{P+R}{P(P+2R)}$$

where P_0 is an arbitrary positive initial value for $P_{0/-1}$. Since $P_{n/n-1} \to P$ as $n \to \infty$, it follows that also $F_n \to F$ and $K_n \to K$, so that the Kalman filtering equations read,

$$\hat{x}_{n+1/n} = \hat{x}_{n/n} = \hat{x}_{n/n-1} + K_n(y_n - \hat{x}_{n/n}) = F_n\hat{x}_{n/n-1} + (1 - F_n)y_n$$

$$\hat{x}_{n+1/n} = \hat{x}_{n/n} = \hat{x}_{n/n-1} + K(y_n - \hat{x}_{n/n}) = F\hat{x}_{n/n-1} + (1 - F)y_n$$

where the second one is the steady-state version, which is recognized as the exponential smoother with parameter $\lambda = F$. We note that because $P > 0$, we have $0 < F < 1$. □

13.7 Steady-State Kalman Filters

As soon as the Kalman filter gains have converged to their asymptotic values, the Kalman filter can be operated as a time-invariant filter with the following input/output equations for the predicted estimate:

$$\boxed{\begin{aligned} \hat{\mathbf{x}}_{n+1/n} &= A\hat{\mathbf{x}}_{n/n-1} + K(\mathbf{y}_n - C\hat{\mathbf{x}}_{n/n-1}) \\ \hat{\mathbf{x}}_{n+1/n} &= (A - KC)\hat{\mathbf{x}}_{n/n-1} + K\mathbf{y}_n \end{aligned}} \quad \text{(steady-state Kalman filter)} \qquad (13.7.1)$$

or, in its prediction-correction form, where $K = AG$,

$$\boxed{\begin{aligned}\hat{\mathbf{x}}_{n/n} &= \hat{\mathbf{x}}_{n/n-1} + G(\mathbf{y}_n - C\hat{\mathbf{x}}_{n/n-1})\\ \hat{\mathbf{x}}_{n+1/n} &= A\hat{\mathbf{x}}_{n/n}\end{aligned}} \quad \text{(steady-state Kalman filter)} \tag{13.7.2}$$

or, in its filtered form, using Eq. (13.6.4),

$$\boxed{\begin{aligned}\hat{\mathbf{x}}_{n/n} &= A\hat{\mathbf{x}}_{n-1/n-1} + G(\mathbf{y}_n - CA\hat{\mathbf{x}}_{n-1/n-1})\\ \hat{\mathbf{x}}_{n/n} &= (A - GCA)\hat{\mathbf{x}}_{n-1/n-1} + G\mathbf{y}_n\end{aligned}} \quad \text{(steady-state Kalman filter)} \tag{13.7.3}$$

Since these depend only on the gains K, G, they may be viewed as state-estimators, or *observers*, independently of the Kalman filter context.

In cases when one does not know the state-model noise parameters Q, R, non-optimal values for the gains K, G may be used (as long as the closed-loop state-transition matrices $F = A - KC$ and $A - GCA$ are stable). Such non-optimal examples include the single and double exponential moving average filters and Holt's exponential smoothing discussed in Chap. 6, as well the general α–β and α–β–γ filters.

The corresponding transfer function matrices from the input $\mathbf{y}_n$ to the prediction $\hat{\mathbf{x}}_{n/n-1}$ and to the filtered estimate $\hat{\mathbf{x}}_{n/n}$ are found by taking z-transforms of Eqs. (13.7.1) and (13.7.2). Denoting the identity matrix by I, we have:

$$\begin{aligned}H_p(z) &= (zI - A + KC)^{-1}K\\ H_f(z) &= z(zI - A + GCA)^{-1}G\end{aligned} \tag{13.7.4}$$

Example 13.7.1: *α–β Tracking Filters.* The kinematic state-space models considered in Example 13.1.3 for a moving object subject to random accelerations were of the form:

$$\begin{aligned}\begin{bmatrix} x_{n+1}\\ \dot{x}_{n+1}\end{bmatrix} &= \begin{bmatrix} 1 & T\\ 0 & 1\end{bmatrix}\begin{bmatrix} x_n\\ \dot{x}_n\end{bmatrix} + \mathbf{w}_n\\ y_n &= [1, 0]\begin{bmatrix} x_n\\ \dot{x}_n\end{bmatrix} + v_n\end{aligned} \tag{13.7.5}$$

with measurement noise variance $R = \sigma_v^2$ and two possible choices for the noise term $\mathbf{w}_n$,

$$\mathbf{w}_n = \begin{bmatrix} 0\\ w_n\end{bmatrix}, \quad \mathbf{w}_n = \begin{bmatrix} T^2/2\\ T\end{bmatrix} a_n$$

where w_n represents a random velocity and a_n a random acceleration. The corresponding covariance matrices $Q = E[\mathbf{w}_n\mathbf{w}_n^T]$ are,

$$Q = \begin{bmatrix} 0 & 0\\ 0 & \sigma_w^2\end{bmatrix}, \quad Q = \begin{bmatrix} T^4/4 & T^3/2\\ T^3/2 & T^2\end{bmatrix}\sigma_a^2$$

An α–β tracking filter is an observer for the model (13.7.5) written in its prediction-correction form of Eq. (13.7.2) with a gain vector defined in terms of the α, β parameters:

$$G = \begin{bmatrix} \alpha\\ \beta/T\end{bmatrix}$$

Eqs. (13.7.2) then read:

$$\begin{aligned}\begin{bmatrix} \hat{x}_{n/n}\\ \hat{\dot{x}}_{n/n}\end{bmatrix} &= \begin{bmatrix} \hat{x}_{n/n-1}\\ \hat{\dot{x}}_{n/n-1}\end{bmatrix} + \begin{bmatrix} \alpha\\ \beta/T\end{bmatrix}(y_n - \hat{x}_{n/n-1})\\ \begin{bmatrix} \hat{x}_{n+1/n}\\ \hat{\dot{x}}_{n+1/n}\end{bmatrix} &= \begin{bmatrix} 1 & T\\ 0 & 1\end{bmatrix}\begin{bmatrix} \hat{x}_{n/n}\\ \hat{\dot{x}}_{n/n}\end{bmatrix}\end{aligned} \tag{13.7.6}$$

where we used $\hat{\mathbf{y}}_{n/n-1} = C\hat{\mathbf{x}}_{n/n-1} = [1, 0]\begin{bmatrix} \hat{x}_{n/n-1}\\ \hat{\dot{x}}_{n/n-1}\end{bmatrix} = \hat{x}_{n/n-1}$. Explicitly, we write,

$$\boxed{\begin{aligned}\hat{x}_{n/n} &= \hat{x}_{n/n-1} + \alpha(y_n - \hat{x}_{n/n-1})\\ \hat{\dot{x}}_{n/n} &= \hat{\dot{x}}_{n/n-1} + \frac{\beta}{T}(y_n - \hat{x}_{n/n-1})\\ \hat{x}_{n+1/n} &= \hat{x}_{n/n} + T\hat{\dot{x}}_{n/n}\\ \hat{\dot{x}}_{n+1/n} &= \hat{\dot{x}}_{n/n}\end{aligned}} \quad (\alpha\text{–}\beta \text{ tracking filter})$$

These are essentially equivalent to Holt's exponential smoothing method discussed in Sec. 6.12. The corresponding prediction and filtering transfer functions of Eq. (13.7.4) are easily found to be:

$$\begin{aligned}H_p(z) &= \frac{1}{z^2 + (\alpha + \beta - 2)z + 1 - \alpha}\begin{bmatrix} (\alpha + \beta)z - \alpha\\ \beta(z - 1)/T\end{bmatrix}\\ H_f(z) &= \frac{1}{z^2 + (\alpha + \beta - 2)z + 1 - \alpha}\begin{bmatrix} z(\beta - \alpha + \alpha z)\\ \beta z(z - 1)/T\end{bmatrix}\end{aligned} \tag{13.7.7}$$

The particular choices $\alpha = 1 - \lambda^2$ and $\beta = (1 - \lambda)^2$ result in the double-exponential smoothing transfer functions for the local level and local slope of Eq. (6.8.5):

$$H_f(z) = \frac{1}{(1 - \lambda z^{-1})^2}\begin{bmatrix} (1 - \lambda)(1 + \lambda - 2\lambda z^{-1})\\ (1 - \lambda)^2(1 - z^{-1})/T\end{bmatrix} \tag{13.7.8}$$

The noise-reduction ratios for the position and velocity components of $\mathbf{H}_f(z)$ are easily found to be [870,874]:

$$\mathcal{R}_x = \frac{2\alpha^2 + 2\beta - 3\alpha\beta}{\alpha(4 - 2\alpha - \beta}, \quad \mathcal{R}_{\dot{x}} = \frac{2\beta^2/T^2}{\alpha(4 - 2\alpha - \beta}$$

Example 13.7.2: *α–β Tracking Filters as Kalman Filters.* Optimum values of the α, β parameters can be obtained if one thinks of the α–β tracking filter as the steady-state Kalman filter of the model (13.7.5). We start with the case defined by the parameters,

$$A = \begin{bmatrix} 1 & T\\ 0 & 1\end{bmatrix}, \quad C = [1, 0], \quad \mathbf{w}_n = \begin{bmatrix} 0\\ w_n\end{bmatrix}, \quad Q = \begin{bmatrix} 0 & 0\\ 0 & \sigma_w^2\end{bmatrix}, \quad R = \sigma_v^2$$

Let P denote the solution of the DARE, $P = A(P - GDG^T)A^T + Q$, where the gain G is:

$$P = \begin{bmatrix} P_{11} & P_{12}\\ P_{12} & P_{22}\end{bmatrix}, \quad D = CPC^T + R = P_{11} + R, \quad G = PC^TD^{-1} = \frac{1}{P_{11} + R}\begin{bmatrix} P_{11}\\ P_{12}\end{bmatrix}$$

and we set $P_{21} = P_{12}$. If G is to be identified with the gain of the α–β tracking filter, we must have:

$$G = \frac{1}{P_{11}+R}\begin{bmatrix} P_{11} \\ P_{12}\end{bmatrix} = \begin{bmatrix} \alpha \\ \beta/T \end{bmatrix} \quad \Rightarrow \quad \frac{P_{11}}{P_{11}+R} = \alpha\,, \quad \frac{P_{12}}{P_{11}+R} = \frac{\beta}{T}$$

which may be solved for P_{11}, P_{12}:

$$P_{11} = R\,\frac{\alpha}{1-\alpha}\,, \quad P_{12} = \frac{R}{T}\,\frac{\beta}{1-\alpha}\,, \quad D = \frac{R}{1-\alpha} \tag{13.7.9}$$

The three parameters α, β, P_{22} fix the matrix P completely. The DARE provides three equations from which these three parameters can be determined in terms of the model statistics σ_w^2, σ_v^2. To this end, let us define the so-called *tracking index* [869], as the dimensionless ratio (note that σ_w has units of velocity, and σ_v, units of length):

$$\boxed{\lambda^2 = \frac{\sigma_w^2 T^2}{\sigma_v^2}} \quad \text{(tracking index)} \tag{13.7.10}$$

Using Eqs. (13.7.9), we obtain

$$P - GDG^T = \begin{bmatrix} R\alpha & R\beta/T \\ R\beta/T & P_{22} - \dfrac{\beta^2/T^2}{1-\alpha}\end{bmatrix}$$

Then, the DARE, $P = A(P - GDG^T)A^T + Q$, reads explicitly,

$$\begin{bmatrix} P_{11} & P_{12} \\ P_{12} & P_{22}\end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1\end{bmatrix}\begin{bmatrix} R\alpha & R\beta/T \\ R\beta/T & P_{22} - \dfrac{\beta^2/T^2}{1-\alpha}\end{bmatrix}\begin{bmatrix} 1 & 0 \\ T & 1\end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & \sigma_w^2\end{bmatrix} \tag{13.7.11}$$

Forming the difference of the two sides, we obtain:

$$A(P - GDG^T)A^T + Q - P = \begin{bmatrix} P_{22}T^2 - \dfrac{R\big((\alpha+\beta)^2 - 2\beta\big)}{1-a} & P_{22}T - \dfrac{R\beta(\alpha+\beta)}{T(1-\alpha)} \\ P_{22}T - \dfrac{R\beta(\alpha+\beta)}{T(1-\alpha)} & \sigma_w^2 - \dfrac{R\beta^2}{T^2(1-\alpha)}\end{bmatrix}$$

Equating the off-diagonal matrix elements to zero provides an expression for P_{22}:

$$P_{22} = \frac{R\beta(\alpha+\beta)}{T^2(1-\alpha)}$$

Then, setting the diagonal elements to zero, gives the two equations for α, β:

$$\beta = \frac{\alpha^2}{2-\alpha}\,, \quad \frac{\beta^2}{1-\alpha} = \frac{\sigma_w^2 T^2}{\sigma_v^2} = \lambda^2 \tag{13.7.12}$$

The first of these was arrived at by [870] using different methods. The system (13.7.12) can be solved explicitly in terms of λ^2 as follows [873]:

$$r = \sqrt{\frac{1}{2} + \sqrt{\frac{1}{4} + \frac{4}{\lambda^2}}}\,, \quad \alpha = \frac{2}{r+1}\,, \quad \beta = \frac{2}{r(r+1)} \tag{13.7.13}$$

Next, consider the alternative kinematic model defined by the parameters [868,869]:

$$A = \begin{bmatrix} 1 & T \\ 0 & 1\end{bmatrix}, \quad C = [1, 0]\,, \quad Q = \begin{bmatrix} T^4/4 & T^3/2 \\ T^3/2 & T^2\end{bmatrix}\sigma_a^2\,, \quad R = \sigma_v^2$$

A similar calculation leads to the DARE solution for the covariance matrix:

$$P_{11} = R\,\frac{\alpha}{1-\alpha}\,, \quad P_{12} = \frac{R}{T}\,\frac{\beta}{1-\alpha}\,, \quad P_{22} = \frac{R}{2T^2}\,\frac{\beta(2\alpha+\beta)}{1-\alpha}\,, \quad D = P_{11} + R = \frac{R}{1-\alpha}$$

with α, β satisfying the conditions:

$$\frac{2\beta - \alpha\beta - \alpha^2}{1-\alpha} = \frac{\lambda^2}{4}\,, \quad \frac{\beta^2}{1-\alpha} = \lambda^2 \tag{13.7.14}$$

where now the tracking index is defined as the dimensionless ratio:

$$\lambda^2 = \frac{\sigma_a^2 T^4}{\sigma_v^2} \tag{13.7.15}$$

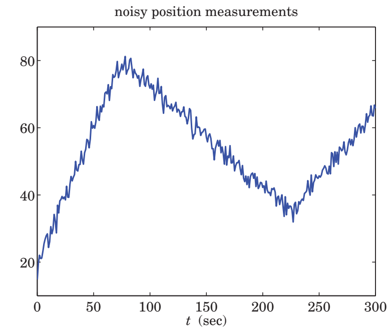
The solution of the system (13.7.14) is found to be [868]:

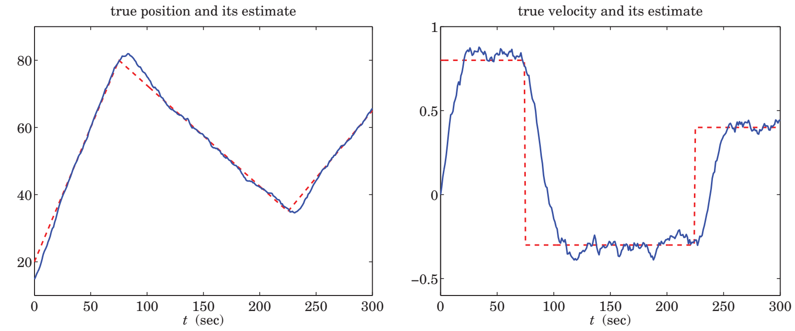
$$r = \sqrt{1 + \frac{8}{\lambda}}\,, \quad \alpha = \frac{4r}{(r+1)^2}\,, \quad \beta = \frac{8}{(r+1)^2} \tag{13.7.16}$$

It is easily verified that these satisfy the Kalata relationship [869]:

$$\beta = 2(2-\alpha) - 4\sqrt{1-\alpha} \tag{13.7.17}$$

For both models, the optimum solutions for α, β given in Eqs. (13.7.13) and (13.7.16) lead to a stable closed-loop matrix $F = A - KC$, that is, its eigenvalues lie inside the unit circle. These eigenvalues are the two roots of the denominator polynomial of the transfer functions (13.7.7), that is, the roots of $z^2 + (\alpha+\beta-2)z + 1 - \alpha = 0$. The graphs below show a simulation.





The following parameter values were chosen $\sigma_a = 0.02$, $\sigma_v = 2$, $T = 1$, which lead to a tracking index (13.7.15) of $\lambda = 0.01$, and the value of the parameter $r = 28.3019$ in Eq. (13.7.16), which gives the $\alpha\beta$-parameters $\alpha = 0.1319$ and $\beta = 0.0093$. The algorithm (13.7.6) was iterated with an initial value $\hat{\mathbf{x}}_{0/-1} = [y_0, 0]^T$.

The following MATLAB code segment shows the generation of the input signal y_n, the computation of α, β, and the filtering operation:

```
t0 = 0;    t1 = 75;    t2 = 225; t3 = 300;   % turning times
b0 = 0.8; b1 = -0.3; b2 = 0.4;               % segment slopes

m0 = 20;
m1 = m0 + b0 * (t1-t0);                     % segment turning points
m2 = m1 + b1 * (t2-t1);

t = (t0:t3); T=1;

s = (m0+b0*t).*upulse(t,t1) + (m1+b1*(t-t1)).*upulse(t-t1,t2-t1) +...
    (m2+b2*(t-t2)).*upulse(t-t2,t3-t2+1);

sdot = b0*upulse(t,t1) + b1*upulse(t-t1,t2-t1) + b2*upulse(t-t2,t3-t2+1);

seed = 1000; randn('state',seed);
sv = 2; v = sv * randn(1,length(t));

y = s + v;                                  % noisy position measurements

sa = 0.02; lambda = sa*T^2/sv; r = sqrt(1+8/lambda);

a = 4*r/(r+1)^2; b = 8/(r+1)^2;                   % alpha-beta parameters

A = [1, T; 0, 1]; C=[1,0]; G = [a; b/T];

xp = [y(1); 0];                                 % initial prediction

for n=1:length(t),
   x(:,n) = xp + G*(y(n) - C*xp);
   xp = A*x(:,n);
end

figure; plot(t,y,'b-');                         % noisy positions
figure; plot(t,s,'r--', t,x(1,:),'b-');         % true & estimated position
figure; plot(t,sdot,'r--', t,x(2,:),'b-');      % true & estimated velocity
```

Example 13.7.3: *Transients of α–β Tracking Kalman Filters.* Here, we look at a simulation of the random-acceleration model of Eq. (13.1.15) and of the time-varying Kalman filter as it converges to steady-state. The model is defined by

$$\begin{bmatrix} x_{n+1} \\ \dot{x}_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_n \\ \dot{x}_n \end{bmatrix} + \begin{bmatrix} T^2/2 \\ T \end{bmatrix} a_n\,, \quad y_n = [1, 0] \begin{bmatrix} x_n \\ \dot{x}_n \end{bmatrix} + v_n$$

with model matrices:

$$A = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \quad C = [1, 0]\,, \quad Q = \begin{bmatrix} T^4/4 & T^3/2 \\ T^3/2 & T^2 \end{bmatrix} \sigma_a^2\,, \quad R = \sigma_v^2$$

The corresponding Kalman filter is defined by Eq. (13.6.3). If we denote the elements of the time-varying Kalman gain G_n by

$$G_n = \begin{bmatrix} \alpha_n \\ \beta_n / T \end{bmatrix}$$

then, we expect α_n, β_n to eventually converge to the steady-state values given in (13.7.16). The Kalman filtering algorithm reads explicitly,

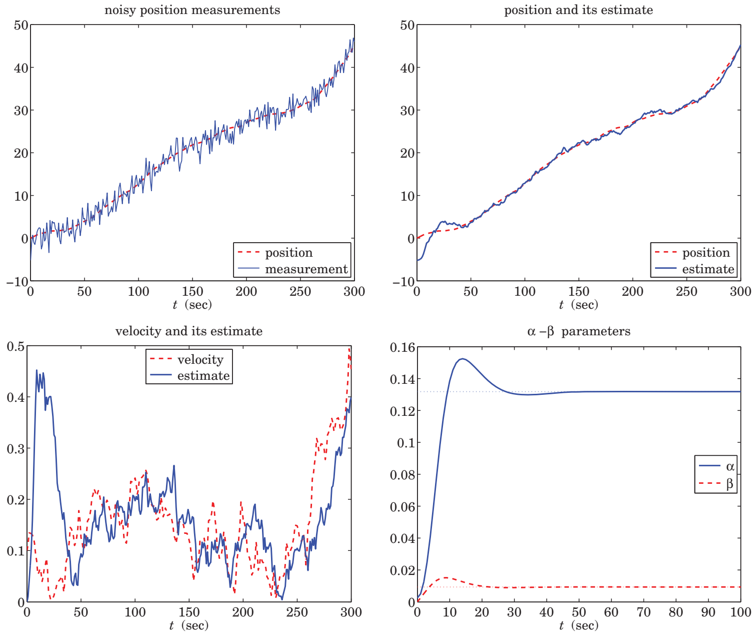
$$\begin{bmatrix} \hat{x}_{n/n} \\ \hat{\dot{x}}_{n/n} \end{bmatrix} = \begin{bmatrix} \hat{x}_{n/n-1} \\ \hat{\dot{x}}_{n/n-1} \end{bmatrix} + \begin{bmatrix} \alpha_n \\ \beta_n / T \end{bmatrix} (y_n - \hat{x}_{n/n-1})\,, \quad \begin{bmatrix} \hat{x}_{n+1/n} \\ \hat{\dot{x}}_{n+1/n} \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \hat{x}_{n/n} \\ \hat{\dot{x}}_{n/n} \end{bmatrix}$$

where

$$D_n = CP_{n/n-1}C^T + R\,, \quad G_n = P_{n/n-1}C^T / D_n$$

$$P_{n/n} = P_{n/n-1} - G_n D_n G_n^T\,, \quad P_{n+1/n} = AP_{n/n}A^T + Q$$

The figures below show a simulation with the same parameter values as in the previous example, $\sigma_a = 0.02$, $\sigma_v = 2$, and $T = 1$.



The upper-left graph shows the noisy measurement y_n plotted together with the position x_n to be estimated. The upper-right graph plots the estimate $\hat{x}_{n/n}$ together with x_n. The lower-left graph shows the velocity and its estimate, $\dot{x}_n$ and $\hat{\dot{x}}_{n/n}$. The lower-right graph shows α_n, β_n as they converge to their steady values $\alpha = 0.1319$, $\beta = 0.0093$, which were calculated from (13.7.16):

$$\lambda = \frac{\sigma_a T^2}{\sigma_v} = 0.01\,, \quad r = \sqrt{1 + \frac{8}{\lambda}} = 28.3019\,, \quad \alpha = \frac{4r}{(r+1)^2}\,, \quad \beta = \frac{8}{(r+1)^2}$$

The model was simulated by generating two independent, zero-mean, gaussian, length-300 acceleration and measurement noise signals a_n, v_n. The initial state vector was chosen at zero position, but with a finite velocity,

$$\mathbf{x}_0 = \begin{bmatrix} x_0 \\ \dot{x}_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0.1 \end{bmatrix}$$

The Kalman filter was initialized to the following predicted state vector and covariance:

$$\hat{\mathbf{x}}_{0/-1} = \begin{bmatrix} \hat{x}_{0/-1} \\ \hat{\dot{x}}_{0/-1} \end{bmatrix} = \begin{bmatrix} y_0 \\ 0 \end{bmatrix}, \quad P_{0/-1} = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}$$

The following MATLAB code illustrates the generation of the graphs:

```
N = 301; T = 1; Tmax = (N-1)*T; t = 0:T:Tmax;

seed = 1000; randn('state',seed);

sv = 2; sa = 0.02; lambda = sa*T^2/sv; r = sqrt(1+8/lambda);
a = 4*r/(r+1)^2; b = 8/(r+1)^2;

v = sv * randn(1,length(t));                % measurement noise
w = [T^2/2; T] * sa * randn(1,length(t));   % state noise

R = sv^2; Q = [T^4/4, T^3/2; T^3/2, T^2]*sa^2;

A = [1,T; 0,1]; C = [1,0];

x0 = [0; 0.1]; x(:,1) = x0;         % initial state

for n=1:N-1                         % generate states and measurements
   x(:,n+1) = A*x(:,n) + w(n);
   y(n) = C*x(:,n) + v(n);
end
   y(N) = C*x(:,N) + v(N);

xp = [y(1);0]; P0 = diag([1,1]/100); P = P0;    % initialize Kalman filter

for n=1:length(t),                              % run Kalman filter
   D = C*P*C'+R;
   G(:,n) = P*C'/D;                             % G = Kalman gain
   X(:,n) = xp + G(:,n)*(y(n) - C*xp);          % X(:,n) = filtered state
   Pf = P - G(:,n)*D*G(:,n)';                   % error covariance of X
   xp = A*X(:,n);                               % xp = predicted state
   P = A*Pf*A' + Q;                             % error covariance of xp
end
```

```
figure; plot(t,x(1,:),'r--', t,y,'b-');
figure; plot(t,x(1,:),'r--', t,X(1,:),'b-');
figure; plot(t,x(2,:),'r--', t,X(2,:),'b-');
figure; plot(t,G(1,:),'b-', t,G(2,:),'r--');
```

The eigenvalues of the asymptotic closed-loop state matrix $F = A - KC$, which are the roots of the polynomial, $z^2 + (\alpha + \beta_2)z + (1 - \alpha)$, can be expressed directly in terms of the parameter r as follows:

$$\lambda_1 = \frac{r^2 - 3 - 2j\sqrt{r^2 - 2}}{(r+1)^2}\,, \quad \lambda_2 = \frac{r^2 - 3 + 2j\sqrt{r^2 - 2}}{(r+1)^2}$$

The eigenvalues are complex conjugates whenever $r^2 > 2$, or equivalently, when the tracking index is $\lambda < 8$, which is usually the case in practice. For $\lambda \geq 8$, or $1 < r^2 \leq 2$, they are real-valued. For the complex case, the eigenvalues have magnitude:

$$|\lambda_1| = |\lambda_2| = \frac{r-1}{r+1}$$

One can then determine an estimate of the convergence time-constant of the Kalman filter,

$$n_{\text{eff}} = \frac{\ln \epsilon}{2 \ln\left(\dfrac{r-1}{r+1}\right)}$$

For the present example, we find $n_{\text{eff}} = 49$ samples for the 60-dB time constant ($\epsilon = 10^{-3}$), which is evident from the above plot of α_n, β_n. Using the methods of [871,872] one may also construct closed-form solutions for the time-varying covariance $P_{n/n-1}$ and gain G_n. First, we determine the eigenvector decomposition of F:

$$F = A - KC = \frac{1}{(r+1)^2} \begin{bmatrix} r^2 - 2r - 7 & T(r+1)^2 \\ -8T & (r+1)^2 \end{bmatrix} = V\Lambda V^{-1}$$

$$V = \begin{bmatrix} v_1 & v_2 \\ 1 & 1 \end{bmatrix}, \quad \Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}, \quad V^{-1} = \frac{1}{v_1 - v_2} \begin{bmatrix} 1 & -v_2 \\ -1 & v_1 \end{bmatrix}$$

$$v_1 = \frac{T}{4}\left[r + 2 - j\sqrt{r^2 - 2}\right], \quad v_2 = \frac{T}{4}\left[r + 2 + j\sqrt{r^2 - 2}\right]$$

Then, the nth power of F is given by:

$$F = V\Lambda V^{-1} = \frac{1}{v_1 - v_2} \begin{bmatrix} v_1\lambda_1 - v_2\lambda_2 & v_1 v_2(\lambda_2 - \lambda_1) \\ \lambda_1 - \lambda_2 & v_1\lambda_2 - v_2\lambda_1 \end{bmatrix}$$

$$F^n = V\Lambda^n V^{-1} = \frac{1}{v_1 - v_2} \begin{bmatrix} v_1\lambda_1^n - v_2\lambda_2^n & v_1 v_2(\lambda_2^n - \lambda_1^n) \\ \lambda_1^n - \lambda_2^n & v_1\lambda_2^n - v_2\lambda_1^n \end{bmatrix}$$

The converged steady-state value of $P_{n/n-1}$, which is the solution of the DARE, may also be expressed in terms of the parameter r, as follows:

$$P = \frac{4rR}{(r+1)^2} \begin{bmatrix} 1 & \dfrac{2}{Tr} \\ \dfrac{2}{Tr} & \dfrac{8}{T^2 r(r+1)} \end{bmatrix}$$

Given an initial 2×2 matrix $P_{0/-1}$, we construct the matrix $E_0 = P_{0/-1} - P$. Then, the exact solution of the Riccati difference equation is given by:

$$P_{n/n-1} = P + F^n E_0\big[I + S_n E_0\big]^{-1} F^{Tn}$$

where I is the 2×2 identity matrix and S_n is defined as follows, for $n \ge 0$,

$$S_n = S - F^{Tn} S F^n\,,\quad S = \frac{r^2-1}{8rR}\begin{bmatrix} 1 & -T/2 \\ -T/2 & T^2(r^2+1)/8 \end{bmatrix}$$

These expressions demonstrate why the convergence time-constant depends on the square of the maximum eigenvalue of F. □

Example 13.7.4: *Local Trend Model.* Holt's exponential smoothing model is an effective way of tracking the local level a_n and local slope b_n of a signal and represents a generalization of the double exponential moving average (DEMA) model. Its state-space form was considered briefly in Sec. 6.13. The following time-invariant linear trend state-space model has steady-state Kalman filtering equations that are equivalent to Holt's method,

$$\begin{bmatrix} a_{n+1} \\ b_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} a_n \\ b_n \end{bmatrix} + \begin{bmatrix} w_n \\ u_n \end{bmatrix},\quad y_n = [1,0]\begin{bmatrix} a_n \\ b_n \end{bmatrix} + v_n \tag{13.7.18}$$

so that its state-model matrices are,

$$A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix},\quad C = [1,0]$$

where a_n, b_n represent the local level and local slope, and w_n, u_n, v_n are zero-mean, mutually uncorrelated, white-noise signals with variances $Q_a = \sigma_w^2$, $Q_b = \sigma_u^2$, $R = \sigma_v^2$. Denote the state vector and its filtered and predicted estimates by,

$$\mathbf{x}_n = \begin{bmatrix} a_n \\ b_n \end{bmatrix},\quad \hat{\mathbf{x}}_{n/n} = \begin{bmatrix} \hat{a}_{n/n} \\ \hat{b}_{n/n} \end{bmatrix},\quad \hat{\mathbf{x}}_{n+1/n} = \begin{bmatrix} \hat{a}_{n+1/n} \\ \hat{b}_{n+1/n} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} \hat{a}_{n/n} \\ \hat{b}_{n/n} \end{bmatrix}$$

so that,

$$\hat{a}_{n+1/n} = \hat{a}_{n/n} + \hat{b}_{n/n}\,,\quad \hat{b}_{n+1/n} = \hat{b}_{n/n}$$

Then, the predicted measurement can be expressed in two ways as follows,

$$\hat{y}_{n/n-1} = C\hat{\mathbf{x}}_{n/n-1} = [1,0]\begin{bmatrix} \hat{a}_{n/n-1} \\ \hat{b}_{n/n-1} \end{bmatrix} = \hat{a}_{n/n-1} = \hat{a}_{n-1/n-1} + \hat{b}_{n-1/n-1}$$

Denote the two-dimensional steady-state Kalman gains G and $K = AG$ by,

$$G = \begin{bmatrix} \alpha \\ \beta \end{bmatrix},\quad K = AG = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha+\beta \\ \beta \end{bmatrix}$$

Then, the steady-state Kalman filtering equations Eq. (13.7.1) and (13.7.3) take the form,

$$\begin{aligned} \hat{\mathbf{x}}_{n/n} &= A\hat{\mathbf{x}}_{n-1/n-1} + G(y_n - \hat{y}_{n/n-1}) \\ \hat{\mathbf{x}}_{n+1/n} &= A\hat{\mathbf{x}}_{n/n-1} + K(y_n - \hat{y}_{n/n-1}) \end{aligned}$$

which are precisely Holt's exponential smoothing formulas,

$$\begin{aligned} \begin{bmatrix} \hat{a}_{n/n} \\ \hat{b}_{n/n} \end{bmatrix} &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} \hat{a}_{n-1/n-1} \\ \hat{b}_{n-1/n-1} \end{bmatrix} + \begin{bmatrix} \alpha \\ \beta \end{bmatrix}(y_n - \hat{a}_{n-1/n-1} - \hat{b}_{n-1/n-1}) \\ \begin{bmatrix} \hat{a}_{n+1/n} \\ \hat{b}_{n+1/n} \end{bmatrix} &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} \hat{a}_{n/n-1} \\ \hat{b}_{n/n-1} \end{bmatrix} + \begin{bmatrix} \alpha+\beta \\ \beta \end{bmatrix}(y_n - \hat{a}_{n/n-1}) \end{aligned}$$

The corresponding model parameters Q_a, Q_b and the error covariance matrix P can be reconstructed in terms of R and the gains α, β, as follows,

$$Q = \begin{bmatrix} Q_a & 0 \\ 0 & Q_b \end{bmatrix} = \frac{R}{1-\alpha}\begin{bmatrix} \alpha^2+\alpha\beta-2\beta & 0 \\ 0 & \beta^2 \end{bmatrix},\quad P = \frac{R}{1-\alpha}\begin{bmatrix} \alpha & \beta \\ \beta & \beta(\alpha+\beta) \end{bmatrix}$$

One can easily verify, according to Eq. (13.6.5), that,

$$D = CPC^T + R = \frac{R}{1-\alpha}\,,\quad G = PC^T D^{-1} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

and that P satisfies the algebraic Riccati equation (13.6.6), that is,

$$P = APA^T - APC^T(CPC^T + R)^{-1}CPA^T + Q$$

Assuming that α, β are positive and that $\alpha < 1$, the positivity of Q_a requires the condition, $\alpha^2 + \alpha\beta > 2\beta$, which also implies that P is positive definite since its determinant is,

$$\det P = R^2\,\frac{(\alpha^2+\alpha\beta-\beta)\beta}{(1-\alpha)^2} = R^2\,\frac{(\alpha^2+\alpha\beta-2\beta+\beta)\beta}{(1-\alpha)^2} > 0$$

Thus, Holt's method admits a Kalman filtering interpretation. □

13.8 Continuous-Time Kalman Filter

The continuous-time Kalman filter, known as the Kalman-Bucy filter [853], is based on the state-space model:

$$\boxed{\begin{aligned} \dot{\mathbf{x}}(t) &= A(t)\mathbf{x}(t) + \mathbf{w}(t) \\ \mathbf{y}(t) &= C(t)\mathbf{x}(t) + \mathbf{v}(t) \end{aligned}} \tag{13.8.1}$$

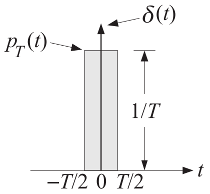
where $\mathbf{w}(t), \mathbf{v}(t)$ are mutually uncorrelated white-noise signals with covariances:

$$\boxed{\begin{aligned} E[\mathbf{w}(t)\mathbf{w}(\tau)^T] &= Q(t)\delta(t-\tau) \\ E[\mathbf{v}(t)\mathbf{v}(\tau)^T] &= R(t)\delta(t-\tau) \\ E[\mathbf{w}(t)\mathbf{v}(\tau)^T] &= 0 \end{aligned}} \tag{13.8.2}$$

We assume also that $\mathbf{w}(t), \mathbf{v}(t)$ are uncorrelated with the initial state vector $\mathbf{x}(0)$. More precisely, one should write the stochastic differential equation for the state in the form: $d\mathbf{x}(t)=A(t)\mathbf{x}(t)dt+\mathbf{w}(t)dt$ and view the quantity $d\mathbf{b}(t)=\mathbf{w}(t)dt$ as a vector-valued Brownian motion, with covariance $E[d\mathbf{b}(t)d\mathbf{b}(t)^T]=Q(t)dt$. However, for the above linear model, the formal manipulations using $\mathbf{w}(t)$ lead to equivalent results.

The continuous-time Kalman filter can be obtained from the discrete one in the limit as the sampling interval T tends to zero. One of the issues that arises is how to define the white-noise sequences $\mathbf{w}_n, \mathbf{v}_n$ of the discrete-time model of Eq. (13.1.1) in terms of the sampled values of the continuous-time signals $\mathbf{w}(t), \mathbf{v}(t)$. The covariance matrix at a single time instant is not well-defined for white noise signals. Indeed, setting $t=\tau=t_n=nT$ in Eq. (13.8.2) would give an infinite value for the covariance $E[\mathbf{w}(t_n)\mathbf{w}(t_n)^T]$.

Since the delta function $\delta(t)$ can be thought of as the limit as $T\to 0$ of a square pulse function $p_T(t)$ of width T and height $1/T$ shown below, we may replace $\delta(t-\tau)$ by $p_T(t-\tau)$ in the right-hand-side of Eq. (13.8.2).



This leads to the approximate but finite values:

$$E[\mathbf{w}(t_n)\mathbf{w}(t_n)^T]=\frac{Q(t_n)}{T}\,,\quad E[\mathbf{v}(t_n)\mathbf{v}(t_n)^T]=\frac{R(t_n)}{T} \tag{13.8.3}$$

The same conclusion can be reached from the Brownian motion point of view, which would give formally,

$$E[\mathbf{w}(t)\mathbf{w}(t)^T]=E\left[\frac{d\mathbf{b}(t)}{dt}\,\frac{d\mathbf{b}(t)^T}{dt}\right]=\frac{E[d\mathbf{b}(t)\,d\mathbf{b}(t)^T]}{dt^2}=\frac{Q(t)dt}{dt^2}=\frac{Q(t)}{dt}$$

and identifying dt by the sampling time T. We may apply now Eq. (13.8.3) to the sampled version of the measurement equation:

$$\mathbf{y}(t_n)=C(t_n)\mathbf{x}(t_n)+\mathbf{v}(t_n)$$

Thus, the discrete-time measurement noise signal $\mathbf{v}_n$ is identified as $\mathbf{v}(t_n)$, with covariance matrix:

$$E[\mathbf{v}_n\mathbf{v}_n^T]=R_n=\frac{R(t_n)}{T} \tag{13.8.4}$$

To identify $\mathbf{w}_n$, we consider the discretized version of the state equation:

$$\dot{\mathbf{x}}(t_n)\approx\frac{\mathbf{x}(t_{n+1})-\mathbf{x}(t_n)}{T}=A(t_n)\mathbf{x}(t_n)+\mathbf{w}(t_n)$$

which gives,

$$\mathbf{x}(t_{n+1})=\big[I+TA(t_n)\big]\mathbf{x}(t_n)+T\mathbf{w}(t_n)$$

and we may identify the discrete-time model quantities:

$$A_n=I+TA(t_n)\,,\quad \mathbf{w}_n=T\mathbf{w}(t_n)$$

with noise covariance matrix $E[\mathbf{w}_n\mathbf{w}_n^T]=T^2E[\mathbf{w}(t_n)\mathbf{w}(t_n)^T]$, or using (13.8.3),

$$E[\mathbf{w}_n\mathbf{w}_n^T]=T^2\cdot\frac{Q(t_n)}{T}=TQ(t_n)\equiv Q_n \tag{13.8.5}$$

To summarize, for small T, the discrete-time and continuous-time signals and model parameters are related by

$$\begin{gathered}\mathbf{x}_n=\mathbf{x}(t_n)\,,\quad \mathbf{w}_n=T\mathbf{w}(t_n)\,,\quad \mathbf{y}_n=\mathbf{y}(t_n)\,,\quad \mathbf{v}_n=\mathbf{v}(t_n)\\ A_n=I+TA(t_n)\,,\quad C_n=C(t_n)\,,\quad Q_n=TQ(t_n)\,,\quad R_n=\frac{R(t_n)}{T}\end{gathered} \tag{13.8.6}$$

Next, consider the limit of the discrete-time Kalman filter. Since $T\to 0$, there will be no distinction between $P_{n/n}$ and $P_{n/n-1}$, and we may set $P(t_n)\approx P_{n/n}\approx P_{n/n-1}$. Using (13.8.6), the innovations covariance and the Kalman gains become approximately (to lowest order in T):

$$\begin{aligned}D_n&=C_nP_{n/n-1}C_n^T+R_n=C(t_n)P(t_n)C(t_n)+\frac{R(t_n)}{T}\approx\frac{R(t_n)}{T}\\ G_n&=P_{n/n-1}C_n^TD_n^{-1}=P(t_n)C(t_n)^TR(t_n)^{-1}T\equiv K(t_n)T\\ K_n&=A_nP_{n/n-1}C_n^TD_n^{-1}=\big[I+TA(t_n)\big]P(t_n)C(t_n)^TR(t_n)^{-1}T\approx K(t_n)T\end{aligned}$$

where we set $K(t_n)=P(t_n)C(t_n)^TR(t_n)^{-1}$. Setting $\hat{\mathbf{x}}(t_n)=\hat{\mathbf{x}}_{n/n-1}$ and hence $\hat{\mathbf{x}}(t_{n+1})=\hat{\mathbf{x}}_{n+1/n}$, the Kalman filtering equation $\hat{\mathbf{x}}_{n+1/n}=A_n\hat{\mathbf{x}}_{n/n-1}+K_n\boldsymbol{\varepsilon}_n$ becomes:

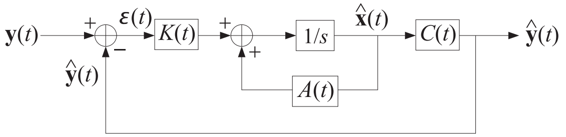
$$\hat{\mathbf{x}}(t_{n+1})=\big[I+TA(t_n)\big]\hat{\mathbf{x}}(t_n)+TK(t_n)\boldsymbol{\varepsilon}(t_n)\,,\quad\text{or,}$$

$$\frac{\hat{\mathbf{x}}(t_{n+1})-\hat{\mathbf{x}}(t_n)}{T}=A(t_n)\hat{\mathbf{x}}(t_n)+K(t_n)\boldsymbol{\varepsilon}(t_n)=A(t_n)\hat{\mathbf{x}}(t_n)+K(t_n)\big[\mathbf{y}(t_n)-C(t_n)\hat{\mathbf{x}}(t_n)\big]$$

which becomes the differential equation in the limit $T\to 0$:

$$\dot{\hat{\mathbf{x}}}(t)=A(t)\hat{\mathbf{x}}(t)+K(t)\boldsymbol{\varepsilon}(t)=A(t)\hat{\mathbf{x}}(t)+K(t)\big[\mathbf{y}(t)-C(t)\hat{\mathbf{x}}(t)\big]$$

with a realization depicted below.



Finally, we consider the limiting form of the Riccati difference equation:

$$P_{n+1/n}=A_n\big[P_{n/n-1}-G_nD_nG_n^T\big]A_n^T+Q_n$$

Using Eqs. (13.8.6) and noting that $P_{n+1/n}=P(t_{n+1})$, we obtain:

$$P(t_{n+1})=\big[I+TA(t_n)\big]\big[P(t_n)-K(t_n)TR(t_n)T^{-1}TK(t_n)^T\big]\big[I+TA(t_n)^T\big]+TQ(t_n)$$

which may be written to order T as follows:

$$\frac{P(t_{n+1})-P(t_n)}{T} = A(t_n)P(t_n)+P(t_n)A(t_n)^T-K(t_n)R(t_n)K(t_n)^T+Q(t_n)$$

which becomes the differential equation in the limit $T\to 0$,

$$\dot{P}(t)=A(t)P(t)+P(t)A(t)^T-K(t)R(t)K(t)^T+Q(t)$$

Substituting $K(t)=P(t)C(t)^TR(t)^{-1}$, we obtain the *Riccati differential equation*:

$$\dot{P}(t)=A(t)P(t)+P(t)A(t)^T-P(t)C(t)^TR(t)^{-1}C(t)P(t)+Q(t)$$

To summarize, the continuous-time Kalman filter for the model (13.8.1) is given by:

$$\boxed{\begin{aligned} K(t) &= P(t)C(t)^TR(t)^{-1}\\ \dot{\hat{\mathbf{x}}}(t) &= A(t)\hat{\mathbf{x}}(t)+K(t)\big[\mathbf{y}(t)-C(t)\hat{\mathbf{x}}(t)\big]\\ \dot{P}(t) &= A(t)P(t)+P(t)A(t)^T-P(t)C(t)^TR(t)^{-1}C(t)P(t)+Q(t)\end{aligned}} \qquad (13.8.7)$$

where $P(t)$ represents the covariance $E[\mathbf{e}(t)\mathbf{e}(t)^T]$ of the estimation error $\mathbf{e}(t)=\mathbf{x}(t)-\hat{\mathbf{x}}(t)$, and the initial conditions are taken to be:

$$\hat{\mathbf{x}}(0)=E[\mathbf{x}(0)]\,,\quad P(0)=E\big[(\mathbf{x}(0)-\hat{\mathbf{x}}(0))(\mathbf{x}(0)-\hat{\mathbf{x}}(0))^T\big]$$

For time-invariant models, i.e. with time-independent model parameters $\{A,C,Q,R\}$, and under the same type of complete stabilizability and detectability assumptions as in the discrete-time case, the Riccati solution $P(t)$ tends to the unique positive-semidefinite symmetric solution P of the *continuous-time algebraic Riccati equation* (CARE):

$$\boxed{AP+PA^T-PC^TR^{-1}CP+Q=0}\quad \text{(CARE)} \qquad (13.8.8)$$

and the Kalman gain tends to the corresponding steady gain $K(t)\to K\equiv PC^TR^{-1}$, resulting in a strictly stable closed-loop state matrix $F=A-KC$, i.e., with eigenvalues in the left-hand s-plane.

Example 13.8.1: *Local Level Model.* The continuous-time version of the local-level model of Example 13.6.2 is defined by the one-dimensional model:

$$\begin{aligned} \dot{x}(t)&=w(t)\,, & E[w(t)w(\tau)]&=Q\delta(t-\tau)\\ y(t)&=x(t)+v(t)\,, & E[v(t)v(\tau)]&=R\delta(t-\tau)\end{aligned}$$

It represents a Wiener (Brownian) process $x(t)$ observed in noise. The Kalman filtering algorithm (13.8.7) has parameters $A=0$, $C=1$, and satisfies the Riccati differential and algebraic equations:

$$\dot{P}(t)=Q-\frac{P^2(t)}{R}\,,\quad Q-\frac{P^2}{R}=0\quad\Rightarrow\quad P=\sqrt{QR}$$

and has time-varying and steady gains:

$$K(t)=\frac{P(t)}{R}\quad\Rightarrow\quad K=\frac{P}{R}=\sqrt{\frac{Q}{R}}$$

and closed-loop transition matrices:

$$F(t)=-K(t)\quad\Rightarrow\quad F=-K=-\sqrt{\frac{Q}{R}}$$

and time-varying and steady Kalman filtering equations:

$$\begin{aligned}\dot{\hat{x}}(t)&=-K(t)\hat{x}(t)+K(t)y(t)\\ \dot{\hat{x}}(t)&=-K\hat{x}(t)+Ky(t)\end{aligned}$$

with the latter representing a continuous-time version of the exponential smoother. The convergence properties to the steady values depend on F. Using the methods of [871,872], we find the exact solution for $P(t)$, and hence $K(t)=P(t)/R$:

$$P(t)=P+\frac{2P(P_0-P)e^{2Ft}}{P_0+P-(P_0-P)e^{2Ft}}\,,\quad t\ge 0$$

where P_0 is an arbitrary positive initial value for $P(0)$, and $e^{2Ft}=e^{-2Kt}$, which decays to zero exponentially with a time constant determined by $2F$, a result analogous to the discrete-time case. □

13.9 Equivalence of Kalman and Wiener Filtering

We saw in Chap. 11 that for the case of a simple scalar state-space model the steady-state Kalman filter was equivalent to the corresponding Wiener filter, and that the innovations signal model of the observation signal was embedded in the Kalman filter. Similar results can be derived in the multichannel case.

Consider the problem of estimating a p-dimensional vector-valued signal $\mathbf{x}_n$ from an r-dimensional signal of observations $\mathbf{y}_n$. In the stationary case, the solution of this problem depends on the following $p\times r$ cross-correlation and $r\times r$ autocorrelation functions and corresponding z-transform spectral densities:

$$\begin{aligned} R_{xy}(k)&=E[\mathbf{x}_n\mathbf{y}_{n-k}^T]\,, & S_{xy}(z)&=\sum_{k=-\infty}^{\infty}R_{xy}(k)z^{-k}\\ R_{yy}(k)&=E[\mathbf{y}_n\mathbf{y}_{n-k}^T]\,, & S_{yy}(z)&=\sum_{k=-\infty}^{\infty}R_{yy}(k)z^{-k}\end{aligned} \qquad (13.9.1)$$

The desired causal estimate of $\mathbf{x}_n$ is given by the convolutional equation:

$$\hat{\mathbf{x}}_n=\sum_{k=0}^{\infty}h_k\mathbf{y}_{n-k}\,,\quad H(z)=\sum_{k=0}^{\infty}h_kz^{-k} \qquad (13.9.2)$$

where h_k is the optimum $p\times r$ causal impulse response matrix to be determined. The optimality conditions are equivalent to the orthogonality between the estimation error $\mathbf{e}_n = \mathbf{x}_n - \hat{\mathbf{x}}_n$ and the observations $\mathbf{y}_{n-k}$, $k \geq 0$, that make up the estimate:

$$R_{ey}(k) = E[\mathbf{e}_n \mathbf{y}_{n-k}^T] = 0\,, \quad k \geq 0 \tag{13.9.3}$$

These are equivalent to the matrix-valued Wiener-Hopf equations for h_k:

$$\sum_{m=0}^{\infty} h_m R_{yy}(k-m) = R_{xy}(k)\,, \quad k \geq 0 \tag{13.9.4}$$

The solution can be constructed in the z-domain with the help of a causal and causally invertible signal model $B(z)$ for the observations $\mathbf{y}_n$, driven by an r-dimensional white-noise sequence $\boldsymbol{\varepsilon}_n$ of (time-independent) covariance $E[\boldsymbol{\varepsilon}_n \boldsymbol{\varepsilon}_{n-k}^T] = D\delta(k)$, that is,

$$\mathbf{y}_n = \sum_{k=0}^{\infty} b_k \boldsymbol{\varepsilon}_{n-k}\,, \quad B(z) = \sum_{k=0}^{\infty} b_k z^{-k} \tag{13.9.5}$$

where b_k is the causal $r\times r$ impulse response matrix of the model. The model implies the spectral factorization of observations spectral density $S_{yy}(z)$:

$$\boxed{S_{yy}(z) = B(z)\, D B^T(z^{-1})} \tag{13.9.6}$$

We will construct the optimal filter $H(z)$ using the gapped-function technique of Chap. 11. To this end, we note that for any k,

$$\begin{aligned} R_{ey}(k) &= E[\mathbf{e}_n \mathbf{y}_{n-k}^T] = E[(\mathbf{x}_n - \hat{\mathbf{x}}_n)\mathbf{y}_{n-k}^T] = E\left[\left(\mathbf{x}_n - \sum_{m=0}^{\infty} h_m \mathbf{y}_{n-m}\right)\mathbf{y}_{n-k}^T\right] \\ &= R_{xy}(k) - \sum_{m=0}^{\infty} h_m R_{yy}(k-m) \end{aligned}$$

and in the z-domain:

$$S_{ey}(z) = S_{xy}(z) - H(z) S_{yy}(z) = S_{xy}(z) - H(z) B(z)\, D B^T(z^{-1})$$

The orthogonality conditions (13.9.3) require that $S_{ey}(z)$ be a strictly left-sided, or anticausal z-transform, and so will be the z-transform obtained by multiplying both sides by the matrix inverse of $B^T(z^{-1})$ because we assumed the $B(z)$ and $B^{-1}(z)$ are causal, the therefore, $B(z^{-1})$ and $B^{-1}(z^{-1})$ will be anticausal. Thus,

$$S_{ey}(z) B^{-T}(z^{-1}) = S_{xy}(z) B^{-T}(z^{-1}) - H(z) B(z) D = \text{strictly anticausal}$$

and therefore, its causal part will be zero:

$$\Big[S_{xy}(z) B(z^{-1})^{-T} - H(z) B(z) D\Big]_+ = \Big[S_{xy}(z) B^{-T}(z^{-1})\Big]_+ - \big[H(z) B(z) D\big]_+ = 0$$

Removing the causal instruction from the second term because it is already causal, we may solve for the optimum Wiener filter for estimating $\mathbf{x}_n$ from $\mathbf{y}_n$:

$$\boxed{H(z) = \Big[S_{xy}(z) B^{-T}(z^{-1})\Big]_+ D^{-1} B^{-1}(z)} \quad \text{(multichannel Wiener filter)} \tag{13.9.7}$$

This generalizes the results of Chap. 11 to vector-valued signals. Similarly, we may obtain for the minimized value of the estimation error covariance:

$$E[\mathbf{e}_n \mathbf{e}_n^T] = R_{ee}(0) = \oint_{\text{u.c.}} \big[S_{xx}(z) - H(z) S_{yx}(z)\big] \frac{dz}{2\pi j z} \tag{13.9.8}$$

The results may be applied to the one-step-ahead prediction problem by replacing the signal $\mathbf{x}_n$ by the signal $\mathbf{x}_1(n) = \mathbf{x}_{n+1}$. Noting that $\mathbf{X}_1(z) = z\mathbf{X}(z)$, we have:

$$H_1(z) = \Big[S_{x_1 y}(z) B^{-T}(z^{-1})\Big]_+ D^{-1} B^{-1}(z)$$

and since $S_{x_1 y}(z) = z S_{xy}(z)$, we find:

$$\boxed{H_1(z) = \Big[z S_{xy}(z) B^{-T}(z^{-1})\Big]_+ D^{-1} B^{-1}(z)} \quad \text{(prediction filter)} \tag{13.9.9}$$

and for the covariance of the error $\mathbf{e}_{n+1/n} = \mathbf{x}_n - \hat{\mathbf{x}}_1(n) = \mathbf{x}_n - \hat{\mathbf{x}}_{n+1/n}$,

$$E[\mathbf{e}_{n+1/n} \mathbf{e}_{n+1/n}^T] = \oint \Big[S_{xx}(z) - H_1(z) S_{yx}(z) z^{-1}\Big] \frac{dz}{2\pi j z} \tag{13.9.10}$$

where in the first term we used $S_{x_1 x_1}(z) = z S_{xx}(z) z^{-1} = S_{xx}(z)$, and in the second term, $S_{y x_1}(z) = S_{yx}(z) z^{-1}$.

Next, we show that Eqs. (13.9.9) and (13.9.10) agree with the results obtained from the steady-state Kalman filter. In particular, we expect the contour integral in Eq. (13.9.10) to be equal to the steady-state solution P of the DARE. We recall from Sec. 13.6 that the steady-state Kalman filter parameters are, where $D = CPC^T + R$:

$$K = APC^T D^{-1} = FPC^T R^{-1}\,, \quad F = A - KC = A - APC^T D^{-1} C \tag{13.9.11}$$

where P is the unique positive-semidefinite symmetric solution of the DARE:

$$P = APA^T - APC^T (CPC^T + R)^{-1} CPA^T + Q \tag{13.9.12}$$

which can also be written as

$$Q = P - FPA^T \tag{13.9.13}$$

The state-space model for the Kalman filter can be written in the z-domain as follows:

$$\begin{aligned} \mathbf{x}_{n+1} &= A\mathbf{x}_n + \mathbf{w}_n \\ \mathbf{y}_n &= C\mathbf{x}_n + \mathbf{v}_n \end{aligned} \quad \Rightarrow \quad \begin{aligned} \mathbf{X}(z) &= (zI - A)^{-1}\mathbf{W}(z) \\ \mathbf{Y}(z) &= C(zI - A)^{-1}\mathbf{W}(z) + \mathbf{V}(z) \end{aligned}$$

from which we obtain the spectral densities:

$$\begin{aligned} S_{xx}(z) &= (zI - A)^{-1} Q (z^{-1}I - A^T)^{-1} \\ S_{xy}(z) &= S_{xx}(z) C^T = (zI - A)^{-1} Q (z^{-1}I - A^T)^{-1} C^T \\ S_{yy}(z) &= C S_{xx}(z) C^T + R = C(zI - A)^{-1} Q (z^{-1}I - A^T)^{-1} C^T + R \end{aligned} \tag{13.9.14}$$

The steady-state Kalman prediction filter is given by

$$\hat{\mathbf{x}}_{n+1/n} = A\hat{\mathbf{x}}_{n/n-1} + K\boldsymbol{\varepsilon}_n = A\hat{\mathbf{x}}_{n/n-1} + K(\mathbf{y}_n - C\hat{\mathbf{x}}_{n/n-1}) = F\hat{\mathbf{x}}_{n/n-1} + K\mathbf{y}_n$$

which may be rewritten in terms of $\hat{\mathbf{x}}_1(n) = \hat{\mathbf{x}}_{n+1/n}$, or, $\hat{\mathbf{x}}_1(n-1) = \hat{\mathbf{x}}_{n/n-1}$,

$$\hat{\mathbf{x}}_1(n) = A\hat{\mathbf{x}}_1(n-1) + K\boldsymbol{\varepsilon}_n = F\hat{\mathbf{x}}_1(n-1) + K\mathbf{y}_n \tag{13.9.15}$$

and in the z-domain, noting that $\hat{\mathbf{y}}_{n/n-1} = C\hat{\mathbf{x}}_{n/n-1} = C\hat{\mathbf{x}}_1(n-1)$,

$$\begin{aligned} \hat{\mathbf{X}}_1(z) &= (I - z^{-1}A)^{-1}K\boldsymbol{\mathcal{E}}(z) = (I - z^{-1}F)^{-1}K\mathbf{Y}(z) \\ \hat{\mathbf{Y}}(z) &= z^{-1}C\hat{\mathbf{X}}_1(z) = C(zI - A)^{-1}K\boldsymbol{\mathcal{E}}(z) = C(zI - F)^{-1}K\mathbf{Y}(z) \end{aligned} \tag{13.9.16}$$

From the first of these, we obtain the prediction filter transfer function $H_1(z)$ relating the observations $\mathbf{y}_n$ to the prediction $\hat{\mathbf{x}}_1(n) = \hat{\mathbf{x}}_{n+1/n}$, i.e., $\hat{\mathbf{X}}_1(z) = H_1(z)\mathbf{Y}(z)$:

$$H_1(z) = (I - z^{-1}F)^{-1}K \tag{13.9.17}$$

Since $\boldsymbol{\varepsilon}_n = \mathbf{y}_n - \hat{\mathbf{y}}_{n/n-1}$, or, $\boldsymbol{\mathcal{E}}(z) = \mathbf{Y}(z) - \hat{\mathbf{Y}}(z)$, the second of Eqs. (13.9.16) allows us to determine the signal model transfer function matrix $B(z)$, i.e., $\mathbf{Y}(z) = B(z)\boldsymbol{\mathcal{E}}(z)$:

$$\begin{aligned} \mathbf{Y}(z) &= \boldsymbol{\mathcal{E}}(z) + \hat{\mathbf{Y}}(z) = \left[I + C(zI - A)^{-1}K\right]\boldsymbol{\mathcal{E}}(z) \\ \boldsymbol{\mathcal{E}}(z) &= \mathbf{Y}(z) - \hat{\mathbf{Y}}(z) = \left[I - C(zI - F)^{-1}K\right]\mathbf{Y}(z) \end{aligned}$$

from which we obtain $B(z)$ and its inverse $B^{-1}(z)$:

$$\begin{aligned} B(z) &= I + C(zI - A)^{-1}K \\ B^{-1}(z) &= I - C(zI - F)^{-1}K \end{aligned} \tag{13.9.18}$$

It can easily be verified that $[I + C(zI - A)^{-1}K][I - C(zI - F)^{-1}K] = I$ by direct multiplication, using the fact that $F = A - KC$. Next, we must verify the spectral factorization of $S_{yy}(z)$ and show that Eq. (13.9.17) agrees with (13.9.9). We will make use of the following relationships:

$$\begin{aligned} &(zI - F)^{-1}KB(z) = (zI - A)^{-1}K \\ &B(z)C(zI - F)^{-1} = C(zI - A)^{-1} \\ &(z^{-1}I - F^T)^{-1}C^T B^T(z^{-1}) = (z^{-1}I - A^T)^{-1}C^T \end{aligned} \tag{13.9.19}$$

where the third is obtained by transposing the second and replacing z by z^{-1}. These can be shown in a straightforward way, for example,

$$\begin{aligned} B(z)C(zI - F)^{-1} &= [I + C(zI - A)^{-1}K]C(zI - F)^{-1} \\ &= [C + C(zI - A)^{-1}KC](zI - F)^{-1} \\ &= C(zI - A)^{-1}(zI - A + KC)(zI - F)^{-1} \\ &= C(zI - A)^{-1}(zI - F)(zI - F)^{-1} = C(zI - A)^{-1} \end{aligned}$$

We will also need the following relationship and its transposed/reflected version:

$$\begin{aligned} &(zI - F)^{-1}Q(z^{-1}I - A^T)^{-1} = (I - z^{-1}F)^{-1}P + PA^T(z^{-1}I - A^T)^{-1} \\ &(zI - A)^{-1}Q(z^{-1}I - F^T)^{-1} = P(I - zF^T)^{-1} + (zI - A)^{-1}AP \end{aligned} \tag{13.9.20}$$

These are a consequence of the DARE written in the form of Eq. (13.9.13). Indeed,

$$\begin{aligned} &(I - z^{-1}F)^{-1}P + PA^T(z^{-1}I - A^T)^{-1} = \\ &\quad = (I - z^{-1}F)^{-1}[P(z^{-1}I - A^T) + (I - z^{-1}F)PA^T](z^{-1}I - A^T)^{-1} \\ &\quad = (I - z^{-1}F)^{-1}[z^{-1}P - PA^T + PA^T - z^{-1}FPA^T](z^{-1}I - A^T)^{-1} \\ &\quad = (I - z^{-1}F)^{-1}z^{-1}(P - FPA^T)(z^{-1}I - A^T)^{-1} = (zI - F)^{-1}Q(z^{-1}I - A^T)^{-1} \end{aligned}$$

Next, we verify the spectral factorization (13.9.6). Using (13.9.19) we have:

$$\begin{aligned} S_{yy}(z) &= C(zI - A)^{-1}Q(z^{-1}I - A^T)^{-1}C^T + R \\ &= B(z)C(zI - F)^{-1}Q(z^{-1}I - A^T)^{-1} + R \end{aligned}$$

Multiplying from the left by $B^{-1}(z)$ and using (13.9.18) and (13.9.20), we obtain:

$$\begin{aligned} &B^{-1}(z)S_{yy}(z) = C(zI - F)^{-1}Q(z^{-1}I - A^T)^{-1} + B^{-1}(z)R \\ &\quad = C[(I - z^{-1}F)^{-1}P + PA^T(z^{-1}I - A^T)^{-1}]C^T + [I - C(z - F)^{-1}K]R \\ &\quad = C[(I - z^{-1}F)^{-1}P + PA^T(z^{-1}I - A^T)^{-1}]C^T + R - Cz^{-1}(I - z^{-1}F)^{-1}KR \\ &\quad = C[(I - z^{-1}F)^{-1}P + PA^T(z^{-1}I - A^T)^{-1}]C^T + R - Cz^{-1}(I - z^{-1}F)^{-1}FPC^T \\ &\quad = C(I - z^{-1}F)^{-1}(I - z^{-1}F)PC^T + CPA^T(z^{-1}I - A^T)^{-1}C^T + R \\ &\quad = CPC^T + CPA^T(z^{-1}I - A^T)^{-1}C^T + R \\ &\quad = CPA^T(z^{-1}I - A^T)^{-1}C^T + D = DK^T(z^{-1}I - A^T)^{-1}C^T + D \\ &\quad = D[I + K^T(z^{-1}I - A^T)^{-1}C^T] = DB^T(z^{-1}) \end{aligned}$$

where we replaced $KR = FPC^T$ and $CPA^T = DK^T$ from Eq. (13.9.11). This verifies Eq. (13.9.6). Next, we obtain the prediction filter using the Wiener filter solution (13.9.9). Using (13.9.19), we have:

$$S_{xy}(z) = (zI - A)^{-1}Q(z^{-1}I - A^T)^{-1}C^T = (zI - A)^{-1}Q(z^{-1}I - F^T)^{-1}C^T B^T(z^{-1})$$

Multiplying by the inverse of $B^T(z)$ from the right and using (13.9.20), we obtain:

$$\begin{aligned} zS_{xy}(z)B^{-T}(z^{-1}) &= z(zI - A)^{-1}Q(z^{-1}I - F^T)^{-1}C^T \\ &= zP(I - zF^T)^{-1}C^T + z(zI - A)^{-1}APC^T \\ &= zP(I - zF^T)^{-1}C^T + (I - z^{-1}A)^{-1}KD \end{aligned}$$

where we replaced $APC^T = KD$. The first term is anti-causal (if it is to have a stable inverse z-transform), while the second term is causal (assuming here that A is strictly stable). Thus, we find the causal part:

$$\Big[zS_{xy}(z)B^{-T}(z^{-1})\Big]_+ = (I-z^{-1}A)^{-1}KD = (I-z^{-1}F)^{-1}KB(z)D \tag{13.9.21}$$

where we used the first of Eqs. (13.9.19). It follows that the Wiener prediction filter is:

$$H_1(z) = \Big[zS_{xy}(z)B^{-T}(z^{-1})\Big]_+ D^{-1}B^{-1}(z) = (I-z^{-1}F)^{-1}K \tag{13.9.22}$$

and agrees with (13.9.17). Finally, we consider the prediction error covariance given by (13.9.10). Noting that $S_{yx}(z) = CS_{xx}(z)$, the integrand of (13.9.10) becomes:

$$\begin{aligned}
S_{xx}(z) - H_1(z)S_{yx}(z)z^{-1} &= \big[I - z^{-1}H_1(z)C\big]S_{xx}(z) = \big[I - (zI-F)^{-1}KC\big]S_{xx}(z)\\
&= (zI-F)^{-1}(zI-F-KC)S_{xx}(z) = (zI-F)^{-1}(zI-A)S_{xx}(z)\\
&= (zI-F)^{-1}(zI-A)(zI-A)^{-1}Q(z^{-1}I-A^T)^{-1} = (zI-F)^{-1}Q(z^{-1}I-A^T)^{-1}\\
&= (I-z^{-1}F)^{-1}P + PA^T(z^{-1}I-A^T)^{-1} = z(zI-F)^{-1}P + zPA^T(I-zA^T)^{-1}
\end{aligned}$$

and the contour integral (13.9.10) becomes:

$$\oint_{\text{u.c.}}\Big[S_{xx}(z) - H_1(z)S_{yx}(z)z^{-1}\Big]\frac{dz}{2\pi jz} = \oint_{\text{u.c.}}\Big[(zI-F)^{-1}P + PA^T(I-zA^T)^{-1}\Big]\frac{dz}{2\pi j}$$

The poles of the second term lie outside the unit circle and do not contribute to the integral. The poles of the first term are the eigenvalues of the matrix F, which all lie inside the unit circle. It is not hard to see (e.g., using the eigenvalue decomposition of F) that the first term integrates into:

$$\oint_{\text{u.c.}}\Big[(zI-F)^{-1}P\Big]\frac{dz}{2\pi j} = P$$

Thus, as expected the Wiener and Kalman expressions for $E[\mathbf{e}_{n+1/n}\mathbf{e}_{n+1/n}^T]$ agree with each other.

13.10 Fixed-Interval Smoothing

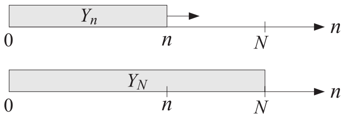
The Kalman filtering algorithm proceeds recursively in time using an ever increasing observations subspace:

$$Y_n = \{\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_n\} = \{\boldsymbol{\varepsilon}_0, \boldsymbol{\varepsilon}_1, \dots, \boldsymbol{\varepsilon}_n\}\,, \quad n = 0, 1, 2, \dots$$

with the current estimate $\hat{\mathbf{x}}_{n/n}$ based on Y_n. In the fixed-interval Kalman smoothing problem, the observations $\mathbf{y}_n$ are available over a fixed time interval $0 \le n \le N$, so that the observation subspace is:

$$Y_N = \{\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_N\} = \{\boldsymbol{\varepsilon}_0, \boldsymbol{\varepsilon}_1, \dots, \boldsymbol{\varepsilon}_N\}$$

and the estimate of $\mathbf{x}_n$, for $0 \le n \le N$, is based on the entire subspace Y_N. The two cases are depicted below:



At each n, the subspace Y_N can be decomposed in the direct sums:

$$\begin{aligned}
Y_N &= \{\boldsymbol{\varepsilon}_0, \boldsymbol{\varepsilon}_1, \dots, \boldsymbol{\varepsilon}_n\} \oplus \{\boldsymbol{\varepsilon}_{n+1}, \dots, \boldsymbol{\varepsilon}_N\} = Y_n \oplus \{\boldsymbol{\varepsilon}_{n+1}, \dots, \boldsymbol{\varepsilon}_N\}\\
Y_N &= \{\boldsymbol{\varepsilon}_0, \boldsymbol{\varepsilon}_1, \dots, \boldsymbol{\varepsilon}_{n-1}\} \oplus \{\boldsymbol{\varepsilon}_n, \dots, \boldsymbol{\varepsilon}_N\} = Y_{n-1} \oplus \{\boldsymbol{\varepsilon}_n, \dots, \boldsymbol{\varepsilon}_N\}
\end{aligned} \tag{13.10.1}$$

and therefore, we expect the estimate of $\mathbf{x}_n$ based on Y_N to be equal to the sum of the filtered estimate $\hat{\mathbf{x}}_{n/n}$, or the predicted estimate $\hat{\mathbf{x}}_{n/n-1}$, plus a correction coming from the rest of the subspace. We will work with the latter decomposition. We will find that once the ordinary Kalman filter has been run forward from $n = 0$ to $n = N$, and we have constructed the innovations basis for Y_N and the estimates $\hat{\mathbf{x}}_{n/n-1}$, the required correction can be constructed recursively, but running backwards from N down to n.

We begin by noting that at each n within $0 \le n \le N$, the state vector $\mathbf{x}_n$ can be written in its unique orthogonal decomposition relative to the subspace Y_{n-1}:

$$\mathbf{x}_n = \hat{\mathbf{x}}_{n/n-1} + \mathbf{e}_{n/n-1} \tag{13.10.2}$$

where $\hat{\mathbf{x}}_{n/n-1}$ is the ordinary predicted estimate of $\mathbf{x}_n$ as defined in the forward Kalman algorithm (13.2.2), and $\mathbf{e}_{n/n-1}$ is the prediction error whose covariance matrix is $P_{n/n-1}$. We observe that $\hat{\mathbf{x}}_{n/n-1}$ can be expressed in terms of the innovations basis of Y_{n-1}, as in Eq. (13.3.16):

$$\hat{\mathbf{x}}_{n/n-1} = \bar{\mathbf{x}}_n + \sum_{m=0}^{n-1} E[\mathbf{x}_n\boldsymbol{\varepsilon}_m^T]D_m^{-1}\boldsymbol{\varepsilon}_m \tag{13.10.3}$$

where $D_m = E[\boldsymbol{\varepsilon}_m\boldsymbol{\varepsilon}_m^T]$. The smoothed estimate of $\mathbf{x}_n$ based on the full subspace Y_N is the projection of $\mathbf{x}_n$ onto Y_N. Denoting this estimate by $\hat{\mathbf{x}}_{n/N}$, it is given in the innovations basis for $Y_N = \{\boldsymbol{\varepsilon}_0, \boldsymbol{\varepsilon}_1, \dots, \boldsymbol{\varepsilon}_N\}$:

$$\hat{\mathbf{x}}_{n/N} = \text{Proj}[\mathbf{x}_n|Y_N] = \bar{\mathbf{x}}_n + \sum_{m=0}^{N} E[\mathbf{x}_n\boldsymbol{\varepsilon}_m^T]D_m^{-1}\boldsymbol{\varepsilon}_m \tag{13.10.4}$$

The summation may be split into two terms:

$$\begin{aligned}
\hat{\mathbf{x}}_{n/N} &= \bar{\mathbf{x}}_n + \sum_{m=0}^{n-1} E[\mathbf{x}_n\boldsymbol{\varepsilon}_m^T]D_m^{-1}\boldsymbol{\varepsilon}_m + \sum_{m=n}^{N} E[\mathbf{x}_n\boldsymbol{\varepsilon}_m^T]D_m^{-1}\boldsymbol{\varepsilon}_m\\
&= \hat{\mathbf{x}}_{n/n-1} + \sum_{m=n}^{N} E[\mathbf{x}_n\boldsymbol{\varepsilon}_m^T]D_m^{-1}\boldsymbol{\varepsilon}_m
\end{aligned}$$

The second term is recognized as the estimate of $\mathbf{e}_{n/n-1}$ based on Y_N, that is,

$$\begin{aligned}\hat{\mathbf{e}}_{n/n-1} &= \sum_{m=0}^{N} E[\mathbf{e}_{n/n-1}\boldsymbol{\varepsilon}_m^T]D_m^{-1}\boldsymbol{\varepsilon}_m \\ &= \sum_{m=0}^{n-1} E[\mathbf{e}_{n/n-1}\boldsymbol{\varepsilon}_m^T]D_m^{-1}\boldsymbol{\varepsilon}_m + \sum_{m=n}^{N} E[\mathbf{e}_{n/n-1}\boldsymbol{\varepsilon}_m^T]D_m^{-1}\boldsymbol{\varepsilon}_m \\ &= \sum_{m=n}^{N} E[(\mathbf{x}_n-\hat{\mathbf{x}}_{n/n-1})\boldsymbol{\varepsilon}_m^T]D_m^{-1}\boldsymbol{\varepsilon}_m = \sum_{m=n}^{N} E[\mathbf{x}_n\boldsymbol{\varepsilon}_m^T]D_m^{-1}\boldsymbol{\varepsilon}_m\end{aligned}$$

where we dropped the terms $E[\mathbf{e}_{n/n-1}\boldsymbol{\varepsilon}_m^T]=0$ for $0 \le m \le n-1$, because of the orthogonality conditions for the estimate $\hat{\mathbf{x}}_{n/n-1}$ (i.e., the estimation error must be orthogonal to the observations that make up the estimate), and then we dropped $E[\hat{\mathbf{x}}_{n/n-1}\boldsymbol{\varepsilon}_m^T]=0$ for $n \le m \le N$ because these $\boldsymbol{\varepsilon}_m$s are orthogonal to the $\boldsymbol{\varepsilon}_m$s making up $\hat{\mathbf{x}}_{n/n-1}$, as seen from the direct sum (13.10.1). Thus, we have:

$$\hat{\mathbf{x}}_{n/N} = \hat{\mathbf{x}}_{n/n-1} + \hat{\mathbf{e}}_{n/n-1} \tag{13.10.5}$$

$$\hat{\mathbf{e}}_{n/n-1} = \sum_{m=n}^{N} E[\mathbf{e}_{n/n-1}\boldsymbol{\varepsilon}_m^T]D_m^{-1}\boldsymbol{\varepsilon}_m \tag{13.10.6}$$

In other words, the term $\hat{\mathbf{e}}_{n/n-1}$ is the correction to the predicted estimate $\hat{\mathbf{x}}_{n/n-1}$ and represents the estimate of the prediction error $\mathbf{e}_{n/n-1}$ based on the subspace $\{\boldsymbol{\varepsilon}_n,\dots,\boldsymbol{\varepsilon}_N\}$ that lies in the future of $\hat{\mathbf{x}}_{n/n-1}$. The same result can be obtained by taking the projections of both sides of Eq. (13.10.2) onto Y_N and noting that the projection of $\hat{\mathbf{x}}_{n/n-1}$ is itself because Y_{n-1} is a subspace of Y_N. The estimation error for the smoothed estimate is equal to the estimation error for $\mathbf{e}_{n/n-1}$, indeed,

$$\mathbf{e}_{n/N} = \mathbf{x}_n - \hat{\mathbf{x}}_{n/N} = \mathbf{x}_n - \hat{\mathbf{x}}_{n/n-1} - \hat{\mathbf{e}}_{n/n-1}\,, \quad \text{or,}$$

$$\mathbf{e}_{n/N} = \mathbf{e}_{n/n-1} - \hat{\mathbf{e}}_{n/n-1} \tag{13.10.7}$$

The error covariance matrices can be obtained by writing $\mathbf{e}_{n/n-1} = \mathbf{e}_{n/N} + \hat{\mathbf{e}}_{n/n-1}$ and noting that the two terms on the right-hand-side are orthogonal because $\hat{\mathbf{e}}_{n/n-1}$ is composed of observations that appear in the $\hat{\mathbf{x}}_{n/N}$ and therefore, they must be orthogonal to the corresponding estimation error $\mathbf{e}_{n/N}$. Let,

$$P_{n/N} = E[\mathbf{e}_{n/N}\mathbf{e}_{n/N}^T]\,, \quad \hat{P}_{n/n-1} = E[\hat{\mathbf{e}}_{n/n-1}\hat{\mathbf{e}}_{n/n-1}^T] \tag{13.10.8}$$

then, the above orthogonality property implies:

$$E[\mathbf{e}_{n/n-1}\mathbf{e}_{n/n-1}^T] = E[\mathbf{e}_{n/N}\mathbf{e}_{n/N}^T] + E[\hat{\mathbf{e}}_{n/n-1}\hat{\mathbf{e}}_{n/n-1}^T]\,, \quad \text{or,}$$

$$P_{n/N} = P_{n/n-1} - \hat{P}_{n/n-1} \tag{13.10.9}$$

The term $\hat{P}_{n/n-1}$ quantifies the improvement in the estimate of $\mathbf{x}_n$ afforded by using all the data Y_N instead of only Y_{n-1}.

Next, we develop the backward recursions satisfied by $\hat{\mathbf{e}}_{n/n-1}$ and $\hat{P}_{n/n-1}$, which will allow the calculation of $\hat{\mathbf{x}}_{n/N}$ and $P_{n/N}$. We recall that $\boldsymbol{\varepsilon}_m = \mathbf{y}_m - C_m\hat{\mathbf{x}}_{m/m-1} = C_m\mathbf{x}_m + \mathbf{v}_m - C_m\hat{\mathbf{x}}_{m/m-1} = C_m\mathbf{e}_{m/m-1} + \mathbf{v}_m$. This implies $E[\mathbf{e}_{n/n-1}\boldsymbol{\varepsilon}_m^T] = E[\mathbf{e}_{n/n-1}\mathbf{e}_{m/m-1}^T]C_m^T$. And it is straightforward to show that for $m \ge n$:

$$E[\mathbf{e}_{n/n-1}\mathbf{e}_{m/m-1}^T] = P_{n/n-1}\Psi_{n,m}\,, \qquad \Psi_{n,m} = \begin{cases} F_n^T F_{n+1}^T \cdots F_{m-1}^T\,, & m > n \\ I\,, & m = n \end{cases} \tag{13.10.10}$$

where $F_n = A_n - K_nC_n$ is the closed-loop transition matrix. For example, consider the case $m = n+1$. Then, $\mathbf{e}_{n+1/n} = \mathbf{x}_{n+1} - \hat{\mathbf{x}}_{n+1/n} = A_n\mathbf{x}_n + \mathbf{w}_n - A_n\hat{\mathbf{x}}_{n/n-1} - K_n\boldsymbol{\varepsilon}_n$, or,

$$\begin{aligned}\mathbf{e}_{n+1/n} &= A_n\mathbf{e}_{n/n-1} + \mathbf{w}_n - K_n\boldsymbol{\varepsilon}_n = A_n\mathbf{e}_{n/n-1} + \mathbf{w}_n - K_n(C_n\mathbf{e}_{n/n-1} + \mathbf{v}_n) \\ &= (A_n - K_nC_n)\mathbf{e}_{n/n-1} + \mathbf{w}_n - K_n\mathbf{v}_n = F_n\mathbf{e}_{n/n-1} + \mathbf{w}_n - K_n\mathbf{v}_n\end{aligned}$$

and because $\mathbf{e}_{n/n-1}$ depends on $\{\mathbf{x}_0, \mathbf{w}_0, \dots, \mathbf{w}_{n-1}, \mathbf{v}_0, \dots, \mathbf{v}_{n-1}\}$, it will be orthogonal to $\mathbf{w}_n, \mathbf{v}_n$, and we find:

$$E[\mathbf{e}_{n/n-1}\mathbf{e}_{n+1/n}^T] = E[\mathbf{e}_{n/n-1}\mathbf{e}_{n/n-1}^T]F_n^T = P_{n/n-1}F_n^T \tag{13.10.11}$$

For $m = n+2$, we have similarly, $\mathbf{e}_{n+2/n+1} = F_{n+1}\mathbf{e}_{n+1/n} + \mathbf{w}_{n+1} - K_{n+1}\mathbf{v}_{n+1}$, and,

$$E[\mathbf{e}_{n/n-1}\mathbf{e}_{n+2/n+1}^T] = E[\mathbf{e}_{n/n-1}\mathbf{e}_{n+1/n}^T]F_{n+1}^T = P_{n/n-1}F_n^T F_{n+1}^T$$

and so on for $m > n$. Thus, we can write $\hat{\mathbf{e}}_{n/n-1}$ in the form:

$$\hat{\mathbf{e}}_{n/n-1} = P_{n/n-1}\sum_{m=n}^{N}\Psi_{n,m}C_m^T D_m^{-1}\boldsymbol{\varepsilon}_m \tag{13.10.12}$$

Separating out the first term and recalling the Kalman gain $G_n = P_{n/n-1}C_n^T D_n^{-1}$, we have:

$$\hat{\mathbf{e}}_{n/n-1} = G_n\boldsymbol{\varepsilon}_n + P_{n/n-1}\sum_{m=n+1}^{N}\Psi_{n,m}C_m^T D_m^{-1}\boldsymbol{\varepsilon}_m \tag{13.10.13}$$

On the other hand, we have:

$$\hat{\mathbf{e}}_{n+1/n} = P_{n+1/n}\sum_{m=n+1}^{N}\Psi_{n+1,m}C_m^T D_m^{-1}\boldsymbol{\varepsilon}_m \quad \Rightarrow \quad \sum_{m=n+1}^{N}\Psi_{n+1,m}C_m^T D_m^{-1}\boldsymbol{\varepsilon}_m = P_{n+1/n}^{-1}\hat{\mathbf{e}}_{n+1/n}$$

Noting that $\Psi_{n,m} = F_n^T\Psi_{n+1,m}$, for $m \ge n+1$, we obtain:

$$\sum_{m=n+1}^{N}\Psi_{n,m}C_m^T D_m^{-1}\boldsymbol{\varepsilon}_m = F_n^T\sum_{m=n+1}^{N}\Psi_{n+1,m}C_m^T D_m^{-1}\boldsymbol{\varepsilon}_m = F_n^T P_{n+1/n}^{-1}\hat{\mathbf{e}}_{n+1/n}$$

and using this into Eq. (13.10.13), we find:

$$\hat{\mathbf{e}}_{n/n-1} = G_n\boldsymbol{\varepsilon}_n + P_{n/n-1}F_n^T P_{n+1/n}^{-1}\hat{\mathbf{e}}_{n+1/n} \tag{13.10.14}$$

Thus, the required backward recursion for $\hat{\mathbf{e}}_{n/n-1}$ may be written as:

$$\boxed{\begin{aligned} L_n &= P_{n/n-1}F_n^T P_{n+1/n}^{-1} \\ \hat{\mathbf{e}}_{n/n-1} &= G_n\boldsymbol{\varepsilon}_n + L_n\hat{\mathbf{e}}_{n+1/n}\end{aligned}} \tag{13.10.15}$$

for $n = N, N-1, \dots, 0$. At $n = N$, Eq. (13.10.12) gives:

$$\hat{\mathbf{e}}_{N/N-1} = P_{N/N-1}\sum_{m=N}^{N}\Psi_{N,m}C_m^T D_m^{-1}\boldsymbol{\varepsilon}_m = P_{N/N-1}\Psi_{N,N}C_N^T D_N^{-1}\boldsymbol{\varepsilon}_N = G_N\boldsymbol{\varepsilon}_N$$

Therefore, the initialization of the recursion (13.10.15) at $n = N$ is:

$$\hat{\mathbf{e}}_{N+1/N} = 0$$

The covariance $\hat{P}_{n/n-1}$ satisfies a similar recursion. Since $\boldsymbol{\varepsilon}_n$ is orthogonal to all the terms of $\hat{\mathbf{e}}_{n+1/n}$, which depend on $\boldsymbol{\varepsilon}_m$, $m \geq n+1$, it follows by taking covariances of both sides of (13.10.15) that:

$$\hat{P}_{n/n-1} = G_n D_n G_n^T + L_n \hat{P}_{n+1/n} L_n^T\,, \quad n = N, N-1, \dots, 0 \tag{13.10.16}$$

and initialized with $\hat{P}_{N+1/N} = 0$.

To summarize, the smoothed estimate $\hat{\mathbf{x}}_{n/N}$ is computed by first running the ordinary Kalman filtering algorithm (13.2.2) forward in time for $n = 0, 1, \dots, N$, saving the quantities $\hat{\mathbf{x}}_{n/n-1}, \boldsymbol{\varepsilon}_n$, along with $P_{n/n-1}, G_n, D_n, F_n = A_n - K_n C_n$, and then, carrying out the following backward recursions from $n = N$ down to $n = 0$,

$$\boxed{\begin{array}{l} \text{Initialize: } \hat{\mathbf{e}}_{N+1/N} = 0,\ \hat{P}_{N+1/N} = 0 \\ \text{for } n = N, N-1, \dots, 0, \text{ do:} \\ \qquad L_n = P_{n/n-1}F_n^T P_{n+1/n}^{-1} \\ \qquad \hat{\mathbf{e}}_{n/n-1} = G_n\boldsymbol{\varepsilon}_n + L_n\hat{\mathbf{e}}_{n+1/n} \\ \qquad \hat{P}_{n/n-1} = G_n D_n G_n^T + L_n\hat{P}_{n+1/n}L_n^T \\ \qquad \hat{\mathbf{x}}_{n/N} = \hat{\mathbf{x}}_{n/n-1} + \hat{\mathbf{e}}_{n/n-1} \\ \qquad P_{n/N} = P_{n/n-1} - \hat{P}_{n/n-1} \end{array}} \tag{13.10.17}$$

We note also that L_n may be written in the form:

$$L_n = P_{n/n}A_n^T P_{n+1/n}^{-1} \tag{13.10.18}$$

Indeed,

$$\begin{aligned} L_n &= P_{n/n-1}F_n^T P_{n+1/n}^{-1} = P_{n/n-1}(A_n - K_n C_n)^T P_{n+1/n}^{-1} \\ &= P_{n/n-1}(A_n^T - C_n^T D_n^{-1} C_n P_{n/n-1} A_n^T) P_{n+1/n}^{-1} \\ &= (P_{n/n-1} - P_{n/n-1}C_n^T D_n^{-1} C_n P_{n/n-1}) A_n^T P_{n+1/n}^{-1} = P_{n/n}A_n^T P_{n+1/n}^{-1} \end{aligned}$$

There exist a number of alternative re-formulations of the smoothing problem that can be derived from algorithm (13.10.17). The so-called Rauch-Tung-Striebel (RTS) version [883,884] is obtained by eliminating the variable $\hat{\mathbf{e}}_{n/n-1}$ in favor of $\hat{\mathbf{x}}_{n/N}$. Applying the equations for the estimate and estimation error at time $n+1$, we have:

$$\begin{aligned} \hat{\mathbf{x}}_{n+1/N} &= \hat{\mathbf{x}}_{n+1/n} + \hat{\mathbf{e}}_{n+1/n} \\ P_{n+1/N} &= P_{n+1/n} - \hat{P}_{n+1/n} \end{aligned} \quad\Rightarrow\quad \begin{aligned} \hat{\mathbf{e}}_{n+1/n} &= \hat{\mathbf{x}}_{n+1/N} - \hat{\mathbf{x}}_{n+1/n} \\ \hat{P}_{n+1/n} &= P_{n+1/n} - P_{n+1/N} \end{aligned}$$

and substituting these into the recursions in (13.10.17), we obtain:

$$\begin{aligned} \hat{\mathbf{e}}_{n/n-1} &= G_n\boldsymbol{\varepsilon}_n + L_n(\hat{\mathbf{x}}_{n+1/N} - \hat{\mathbf{x}}_{n+1/n}) \\ \hat{P}_{n/n-1} &= G_n D_n G_n^T + L_n(P_{n+1/n} - P_{n+1/N})L_n^T \\ \hat{\mathbf{x}}_{n/N} &= \hat{\mathbf{x}}_{n/n-1} + \hat{\mathbf{e}}_{n/n-1} = \hat{\mathbf{x}}_{n/n-1} + G_n\boldsymbol{\varepsilon}_n + L_n(\hat{\mathbf{x}}_{n+1/N} - \hat{\mathbf{x}}_{n+1/n}) \\ P_{n/N} &= P_{n/n-1} - \hat{P}_{n/n-1} = P_{n/n-1} - G_n D_n G_n^T + L_n(P_{n+1/N} - P_{n+1/n})L_n^T \end{aligned}$$

but from the ordinary Kalman filter, we have the filtered estimate and its covariance:

$$\begin{aligned} \hat{\mathbf{x}}_{n/n} &= \hat{\mathbf{x}}_{n/n-1} + G_n\boldsymbol{\varepsilon}_n \\ P_{n/n} &= P_{n/n-1} - G_n D_n G_n^T \end{aligned}$$

Hence, the above smoothed estimates can be written in the RTS form:

$$\boxed{\begin{aligned} L_n &= P_{n/n}A_n^T P_{n+1/n}^{-1} \\ \hat{\mathbf{x}}_{n/N} &= \hat{\mathbf{x}}_{n/n} + L_n(\hat{\mathbf{x}}_{n+1/N} - \hat{\mathbf{x}}_{n+1/n}) \\ P_{n/N} &= P_{n/n} + L_n(P_{n+1/N} - P_{n+1/n})L_n^T \end{aligned}} \quad \text{(RTS smoothing)} \tag{13.10.19}$$

This is to be iterated from $n = N$ down to $n = 0$, where the differences in the second terms are initialized to zero, e.g., at $n = N$, we have $L_N(\hat{\mathbf{x}}_{N+1/N} - \hat{\mathbf{x}}_{N+1/N}) = 0$.

A disadvantage of the algorithm (13.10.17) and of the RTS form is that the computation of L_n requires an additional matrix inversion of the quantity $P_{n+1/n}$. Such inversion is avoided in the so-called Bryson-Frazier (BF) smoothing formulation [881,882]. To derive it, we use Eq. (13.10.12) to define the quantity:

$$\mathbf{g}_n = P_{n/n-1}^{-1}\hat{\mathbf{e}}_{n/n-1} = \sum_{m=n}^{N}\Psi_{n,m}C_m^T D_m^{-1}\boldsymbol{\varepsilon}_m \tag{13.10.20}$$

It follows from Eq. (13.10.14) and $G_n = P_{n/n-1}C_n^T D_n^{-1}$, and $\mathbf{g}_{n+1} = P_{n+1/n}^{-1}\hat{\mathbf{e}}_{n+1/n}$, that

$$\mathbf{g}_n = P_{n/n-1}^{-1}\hat{\mathbf{e}}_{n/n-1} = P_{n/n-1}^{-1}\big(G_n\boldsymbol{\varepsilon}_n + P_{n/n-1}F_n^T P_{n+1/n}^{-1}\hat{\mathbf{e}}_{n+1/n}\big)\,, \quad \text{or,}$$

$$\mathbf{g}_n = C_n^T D_n^{-1}\boldsymbol{\varepsilon}_n + F_n^T\mathbf{g}_{n+1} \tag{13.10.21}$$

with initial value $\mathbf{g}_{N+1} = 0$, which follows from $\hat{\mathbf{e}}_{n+1/N} = 0$. From Eq. (13.10.20) we note that the two terms $\boldsymbol{\varepsilon}_n$ and $\mathbf{g}_{n+1}$ in the right-hand-side are orthogonal, and therefore, we obtain the following recursion for the covariance $\Gamma_n = E[\mathbf{g}_n\mathbf{g}_n^T]$:

$$\Gamma_n = C_n^T D_n^{-1} C_n + F_n^T\Gamma_{n+1}F_n \tag{13.10.22}$$

where we used $C_n^T D_n^{-1}E[\boldsymbol{\varepsilon}_n\boldsymbol{\varepsilon}_n^T]D_n^{-1}C_n = C_n^T D_n^{-1}D_n D_n^{-1}C_n = C_n^T D_n^{-1}C_n$. The recursion is to be initialized at $\Gamma_{N+1} = 0$. Noting that,

$$\hat{P}_{n/n-1} = E[\hat{\mathbf{e}}_{n/n-1}\hat{\mathbf{e}}_{n/n-1}^T] = P_{n/n-1}E[\mathbf{g}_n\mathbf{g}_n^T]P_{n/n-1} = P_{n/n-1}\Gamma_n P_{n/n-1}$$

we obtain the Bryson-Frasier smoothing algorithm:

$$\boxed{\begin{aligned}&\text{Initialize: } \mathbf{g}_{N+1}=0,\ \Gamma_{N+1}=0\\ &\text{for } n=N,N-1,\dots,0,\text{ do:}\\ &\qquad \mathbf{g}_n = C_n^T D_n^{-1}\boldsymbol{\varepsilon}_n + F_n^T\mathbf{g}_{n+1}\\ &\qquad \Gamma_n = C_n^T D_n^{-1}C_n + F_n^T\Gamma_{n+1}F_n\\ &\qquad \hat{\mathbf{x}}_{n/N} = \hat{\mathbf{x}}_{n/n-1} + P_{n/n-1}\mathbf{g}_n\\ &\qquad P_{n/N} = P_{n/n-1} - P_{n/n-1}\Gamma_n P_{n/n-1}\end{aligned}}\qquad\text{(BF smoothing)}\qquad(13.10.23)$$

The algorithm requires no new inversions—the quantity $D_n^{-1}C_n$ was computed as part of the forward Kalman algorithm. The RTS and BF algorithms have also been studied within the statistical time-series analysis literature [885,886,903,904]. Further details on smoothing algorithms may be found in [865].

The MATLAB function, `ksmooth.m`, implements the Kalman smoothing algorithm of Eq. (13.10.23). It has usage:

```
[L,Xs,Ps,V] = ksmooth(A,C,Q,R,Y,x0,S0);      % Bryson-Frazier smoothing
```

Its inputs are the state-space model parameters $\{A,C,Q,R\}$, the initial values $\bar{\mathbf{x}}_0$, Σ_0, and the observations $\mathbf{y}_n$, $0\le n\le N$, arranged into a $r\times(N+1)$ matrix:

$$Y=\big[\mathbf{y}_0,\mathbf{y}_1,\dots,\mathbf{y}_n,\dots,\mathbf{y}_N\big]$$

The outputs are the smoothed estimates arranged into the $p\times(N+1)$ matrix:

$$X_s=\big[\hat{\mathbf{x}}_{0/N},\,\hat{\mathbf{x}}_{1/N},\,\dots,\,\hat{\mathbf{x}}_{n/N},\,\dots,\,\hat{\mathbf{x}}_{N/N}\big]$$

with corresponding error covariance matrices arranged into the $p\times p\times(N+1)$ three-dimensional array P_s, such that (in MATLAB notation):

$$\texttt{Ps(:,:,n+1)} = P_{n/N}\,,\quad 0\le n\le N$$

The output L is the value of the negative-log-likelihood function calculated from Eq. (13.12.2). The quantity V is an optional output that stores the matrix $V_{n+1,n}= E[\mathbf{e}_{n+1/N}\mathbf{e}_{n/N}^T]$ into a $p\times p\times(N+1)$ array. This quantity is used in Sec. 13.13 in the maximum likelihood estimation of the state-space model parameters using the EM algorithm. A convenient expression for it can be derived as follows. We rewrite Eq. (13.10.7) in its orthogonal decomposition forms:

$$\begin{aligned}\mathbf{e}_{n/n-1}&=\mathbf{e}_{n/N}+\hat{\mathbf{e}}_{n/n-1}\\ \mathbf{e}_{n+1/n}&=\mathbf{e}_{n+1/N}+\hat{\mathbf{e}}_{n+1/n}\end{aligned}\qquad(13.10.24)$$

where $E[\mathbf{e}_{n+1/N}\hat{\mathbf{e}}_{n/n-1}^T]= E[\mathbf{e}_{n/N}\hat{\mathbf{e}}_{n+1/n}^T]=0$, which follow from the fact that $\mathbf{e}_{n/N},\mathbf{e}_{n+1/N}$ are estimation errors and must be orthogonal to all the observations Y, and therefore, must also be orthogonal to $\hat{\mathbf{e}}_{n/n-1},\hat{\mathbf{e}}_{n+1/n}$ because the latter are made up from a subset of Y. Then, we find for the cross-covariance:

$$E[\mathbf{e}_{n+1/n}\mathbf{e}_{n/n-1}^T] = E[\mathbf{e}_{n+1/N}\mathbf{e}_{n/N}^T]+E[\hat{\mathbf{e}}_{n+1/n}\hat{\mathbf{e}}_{n/n-1}^T]\,,\quad\text{or,}$$

$$V_{n+1,n}=E[\mathbf{e}_{n+1/N}\mathbf{e}_{n/N}^T] = E[\mathbf{e}_{n+1/n}\mathbf{e}_{n/n-1}^T]-E[\hat{\mathbf{e}}_{n+1/n}\hat{\mathbf{e}}_{n/n-1}^T]$$

From Eq. (13.10.11) we have, $E[\mathbf{e}_{n+1/n}\mathbf{e}_{n/n-1}^T]= F_nP_{n/n-1}$, and from Eq. (13.10.20) we may replace $\hat{\mathbf{e}}_{n+1/n}=P_{n/n-1}\mathbf{g}_n$, to obtain:

$$V_{n+1,n}=E[\mathbf{e}_{n+1/N}\mathbf{e}_{n/N}^T]= F_nP_{n/n-1}-P_{n+1/n}E[\mathbf{g}_{n+1}\mathbf{g}_n^T]P_{n/n-1}$$

From the recursion (13.10.21), $\mathbf{g}_n = C_n^TD_n^{-1}\boldsymbol{\varepsilon}_n+F_n^T\mathbf{g}_{n+1}$, and $E[\mathbf{g}_{n+1}\boldsymbol{\varepsilon}_n^T]=0$, we find:

$$E[\mathbf{g}_{n+1}\mathbf{g}_n^T]= E[\mathbf{g}_{n+1}(C_n^TD_n^{-1}\boldsymbol{\varepsilon}_n+F_n^T\mathbf{g}_{n+1})^T]= E[\mathbf{g}_{n+1}\mathbf{g}_{n+1}^T]F_n=\Gamma_{n+1}F_n$$

It follows then that:

$$\begin{aligned}V_{n+1,n}=E[\mathbf{e}_{n+1/N}\mathbf{e}_{n/N}^T] &= F_nP_{n/n-1}-P_{n+1/n}\Gamma_{n+1}F_nP_{n/n-1}\\ &=[I-P_{n+1/n}\Gamma_{n+1}]F_nP_{n/n-1}\end{aligned}\qquad(13.10.25)$$

13.11 Square-Root Algorithms

In its most basic form the Kalman filtering algorithm reads:

$$\begin{aligned}&D=CPC^T+R\\ &G=PC^TD^{-1},\ K=AG\\ &P_f=P-PC^TD^{-1}CP\,, &&\hat{\mathbf{x}}_f=\hat{\mathbf{x}}+G(\mathbf{y}-C\hat{\mathbf{x}})=(I-GC)\hat{\mathbf{x}}+G\mathbf{y}\\ &P_{\text{new}}=AP_fA^T+Q\,, &&\hat{\mathbf{x}}_{\text{new}}=A\hat{\mathbf{x}}_f=(A-KC)\hat{\mathbf{x}}+K\mathbf{y}\end{aligned}\qquad(13.11.1)$$

where $\hat{\mathbf{x}},\hat{\mathbf{x}}_f,\hat{\mathbf{x}}_{\text{new}}$ denote the current prediction, filtered estimate, and next prediction, $\hat{\mathbf{x}}_{n/n-1},\hat{\mathbf{x}}_{n/n},\hat{\mathbf{x}}_{n+1/n}$, and P,P_f,P_{new} denote the corresponding mean-square error covariance matrices $P_{n/n-1},P_{n/n},P_{n+1/n}$, and we dropped the time indices to simplify the notation. The matrices P,P_f,P_{new} must remain positive semi-definite during the iteration of the algorithm. Because of the subtraction required to calculate P_f, it is possible that rounding errors may destroy its positivity. The "square-root" formulations operate on the square-root factors of the covariance matrices, and thus, guarantee the positivity at each iteration step.

A positive semi-definite symmetric matrix P can always be written as the product of a *lower triangular* square-root factor S and its transpose S^T:

$$P=SS^T\qquad(13.11.2)$$

For example in MATLAB, one may use the built-in Cholesky factorization function `chol`, if P is strictly positive definite, with S constructed as:

```
S = chol(P)';                % S = lower triangular such that P = S*S'
```

If P is semi-definite with some positive and some zero eigenvalues, then one can apply the QR factorization to the square root of P obtained from its eigenvalue or SVD decomposition, with the following MATLAB construction:

```
[v,s] = eig(P);                  % v = eigenvector matrix, s = eigenvalues
P_sqrt = v*real(sqrt(s))*v';     % eigenvalues s are non-negative
[q,r] = qr(P_sqrt);              % QR-factorization, P_sqrt = q*r
S = r';                          % S = lower triangular such that P = S*S'
```

Example 13.11.1: For the following positive-definite case, we find:

$$P=\begin{bmatrix}6&5&4\\5&6&4\\4&4&3\end{bmatrix}=\begin{bmatrix}2.4495&0&0\\2.0412&1.3540&0\\1.6330&0.4924&0.3015\end{bmatrix}\begin{bmatrix}2.4495&0&0\\2.0412&1.3540&0\\1.6330&0.4924&0.3015\end{bmatrix}^T$$

This P has full rank as the product of two rank-3 matrices:

$$P=\begin{bmatrix}6&5&4\\5&6&4\\4&4&3\end{bmatrix}=\begin{bmatrix}2&-1&1\\1&-2&1\\1&-1&1\end{bmatrix}\begin{bmatrix}2&-1&1\\1&-2&1\\1&-1&1\end{bmatrix}^T$$

Similarly, for the following semi-definite P, we have:

$$P=\begin{bmatrix}5&4&3\\4&5&3\\3&3&2\end{bmatrix}=\begin{bmatrix}2.2361&0&0\\1.7889&1.3416&0\\1.3416&0.4472&0\end{bmatrix}\begin{bmatrix}2.2361&0&0\\1.7889&1.3416&0\\1.3416&0.4472&0\end{bmatrix}^T$$

This P has rank two as the product of the rank-2 matrices:

$$P=\begin{bmatrix}5&4&3\\4&5&3\\3&3&2\end{bmatrix}=\begin{bmatrix}2&1\\1&2\\1&1\end{bmatrix}\begin{bmatrix}2&1&1\\1&2&1\end{bmatrix}$$

and has eigenvalues $\{0,1,11\}$. □

The Joseph forms of the covariance updating equations promote the numerical stability of the algorithm because they consist of the sum of positive semidefinite terms (provided of course that P is already positive semidefinite), but require twice as many operations as the conventional forms:

$$\begin{aligned}P_f&=(I-GC)P(I-GC)^T+GRG^T\\P_{\text{new}}&=AP_fA^T+Q=(A-KC)P(A-KC)^T+KRK^T+Q\end{aligned}\tag{13.11.3}$$

Let $\bar R,\bar Q$ be lower-triangular square root factors for R,Q, so that $R=\bar R\bar R^T$ and $Q=\bar Q\bar Q^T$, and substitute the factorizations $P=SS^T$ and $P_f=S_fS_f^T$ in Eq. (13.11.3):

$$S_fS_f^T=(I-GC)SS^T(I-GC)^T+G\bar R\bar R^TG=\big[(I-GC)S,G\bar R\big]\begin{bmatrix}S^T(I-GC)^T\\\bar R^TG^T\end{bmatrix}$$

Then, S_f can be obtained from the upper-triangular factor resulting by applying the QR-factorization to the $(p+r)\times p$ matrix:

$$\begin{bmatrix}S^T(I-GC)^T\\\bar R^TG^T\end{bmatrix}=U\begin{bmatrix}S_f^T\\0\end{bmatrix},\quad S_f^T=\text{upper-triangular}$$

where U is a $(p+r)\times(p+r)$ orthogonal matrix. The lower-triangular version of this relationship reads:

$$\big[(I-GC)S,\,G\bar R\big]=[S_f\,,0]U^T\tag{13.11.4}$$

where the S_f is lower-triangular and the dimensions of the matrices are:

$$\big[\underbrace{(I-GC)S}_{p\times p},\,\underbrace{G\bar R}_{p\times r}\big]=\big[\underbrace{S_f}_{p\times p},\,\underbrace{0}_{p\times r}\big]U^T$$

Since $U^TU=I$, we verify:

$$\big[(I-GC)S,G\bar R\big]\begin{bmatrix}S^T(I-GC)^T\\\bar R^TG^T\end{bmatrix}=[S_f\,,0]U^TU\begin{bmatrix}S_f^T\\0\end{bmatrix}=S_fS_f^T$$

For the time-update step, we note that,

$$P_{\text{new}}=S_{\text{new}}S_{\text{new}}^T=AP_fA^T+Q=AS_fS_f^T+\bar Q\bar Q^T=[AS_f,\,\bar Q]\begin{bmatrix}S_f^TA^T\\\bar Q^T\end{bmatrix}$$

so that we may obtain S_{new} from the lower-triangular version of the QR-algorithm applied to the $p\times(2p)$ matrix:

$$[AS_f,\,\bar Q]=[S_{\text{new}},\,0]U^T\tag{13.11.5}$$

with another $(2p)\times(2p)$ orthogonal matrix U. Combining Eqs. (13.11.4) and (13.11.5), we summarize the measurement and time updating algorithm that propagates the square-root lower-triangular factors S,S_f,S_{new}:

$$\begin{aligned}&P=SS^T\,,\quad D=CPC^T+R\,,\quad G=PC^TD^{-1}\\&\big[(I-GC)S,\,G\bar R\big]=[S_f\,,0]U^T\\&[AS_f,\,\bar Q]=[S_{\text{new}},\,0]U^T\end{aligned}$$

The intermediate step of computing S_f can be avoided by applying the lower-triangular version of the QR-algorithm to the larger $p\times(2p+r)$ matrix:

$$\big[(A-KC)S,\,K\bar R,\,\bar Q\big]=[S_{\text{new}},\,0,\,0]U^T\,,\quad S_{\text{new}}=\text{lower-triangular}\tag{13.11.6}$$

which is equivalent to the second Joseph form in Eq. (13.11.3):

$$(A-KC)SS^T(A-KC)^T+K\bar R\bar R^TK^T+\bar Q\bar Q^T=\big[(A-KC)S,\,K\bar R,\,\bar Q\big]\begin{bmatrix}S^T(A-KC)^T\\\bar R^TK^T\\\bar Q^T\end{bmatrix}$$

$$=[S_{\text{new}},\,0,\,0]U^TU\begin{bmatrix}S_{\text{new}}^T\\0\\0\end{bmatrix}=S_{\text{new}}S_{\text{new}}^T$$

Restoring the time-indices, we obtain the following square-root algorithms based on the Joseph forms. Given $S_{n/n-1}$, calculate:

$$\boxed{\begin{aligned}&P_{n/n-1}=S_{n/n-1}S_{n/n-1}^T\\&D_n=C_nP_{n/n-1}C_n^T+R_n\,,\quad G_n=P_{n/n-1}C_n^TD_n^{-1}\\&\big[(I-G_nC_n)S_{n/n-1},\,G_n\bar R_n\big]=[S_{n/n}\,,0]U^T\\&[A_nS_{n/n},\,\bar Q_n]=[S_{n+1/n},\,0]U^T\end{aligned}}\tag{13.11.7}$$

or, going directly to $S_{n+1/n}$:

$$\boxed{\big[(A_n - K_nC_n)S_{n/n-1},\, K_n\bar{R}_n,\, \bar{Q}_n\big] = [S_{n+1/n},\, 0,\, 0]U^T} \qquad (13.11.8)$$

Example 13.11.2: Consider a system with constant model parameters:

$$A = \begin{bmatrix} 0.5 & 0.1 \\ 0.2 & 0.4 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \quad Q = \begin{bmatrix} 1 & 2 \\ 2 & 5 \end{bmatrix}, \quad R = \begin{bmatrix} 9 & 6 \\ 6 & 8 \end{bmatrix}$$

and lower-triangular square-root factors $\bar{Q}, \bar{R}$ such that $Q = \bar{Q}\bar{Q}^T$ and $R = \bar{R}\bar{R}^T$:

$$\bar{Q} = \begin{bmatrix} 1 & 0 \\ 2 & 1 \end{bmatrix}, \quad \bar{R} = \begin{bmatrix} 3 & 0 \\ 2 & 2 \end{bmatrix}$$

and initial square-root factor S and covariance matrix P:

$$S = \begin{bmatrix} 1.3184 & 0 \\ 1.8820 & 1.4731 \end{bmatrix}, \quad P = SS^T = \begin{bmatrix} 1.7383 & 2.4813 \\ 2.4813 & 5.7118 \end{bmatrix}$$

Then, we calculate D and G:

$$D = CPC^T + R = \begin{bmatrix} 21.4126 & 14.1931 \\ 14.1931 & 13.7118 \end{bmatrix}, \quad G = PC^TD^{-1} = \begin{bmatrix} 0.2457 & -0.0733 \\ 0.3393 & 0.0653 \end{bmatrix}$$

and the covariance matrices P_f, P_{new} from the conventional algorithm:

$$P_f = P - GDG^T = \begin{bmatrix} 0.8836 & 0.8874 \\ 0.8874 & 2.5585 \end{bmatrix}, \quad P_{\text{new}} = AP_fA^T + Q = \begin{bmatrix} 1.3352 & 2.3859 \\ 2.3859 & 5.5867 \end{bmatrix}$$

Next, we form the matrix $X \equiv \big[(I - GC)S,\, G\bar{R}\big]$ and apply the QR-algorithm to its transpose to get the factorization:

$$X = \big[(I - GC)S,\, G\bar{R}\big] = \left[\begin{array}{cc|cc} 0.6702 & -0.2539 & 0.5903 & -0.1467 \\ 0.6730 & 0.8770 & 1.1486 & 0.1306 \end{array}\right]$$

$$= \left[\begin{array}{cc|cc} 0.9400 & 0 & 0 & 0 \\ 0.9440 & 1.2913 & 0 & 0 \end{array}\right] U^T = [S_f,\, 0]U^T \equiv LU^T$$

$$U = \begin{bmatrix} 0.7130 & -0.0000 & -0.3139 & -0.6270 \\ -0.2701 & 0.8766 & 0.1350 & -0.3747 \\ 0.6280 & 0.4304 & 0.1856 & 0.6212 \\ -0.1560 & 0.2152 & -0.9213 & 0.2839 \end{bmatrix}$$

This was obtained from the MATLAB code:

```
[U,r] = qr(X'); L = r';
```

Thus, we determine S_f, and verify that $P_f = S_fS_f^T$ is the same as above:

$$S_f = \begin{bmatrix} 0.9400 & 0 \\ 0.9440 & 1.2913 \end{bmatrix}, \quad P_f = S_fS_f^T = \begin{bmatrix} 0.8836 & 0.8874 \\ 0.8874 & 2.5585 \end{bmatrix}$$

Next, with this S_f, we calculate the matrix $X = [AS_f, \bar{Q}]$ and apply the QR-factorization to its transpose to get:

$$X = [AS_f, \bar{Q}] = \left[\begin{array}{cc|cc} 0.5644 & 0.1291 & 1 & 0 \\ 0.5656 & 0.5165 & 2 & 1 \end{array}\right]$$

$$= \left[\begin{array}{cc|cc} 1.1555 & 0 & 0 & 0 \\ 2.0648 & 1.1503 & 0 & 0 \end{array}\right] U^T = [S_{\text{new}},\, 0] \equiv LU^T$$

$$U = \begin{bmatrix} 0.4884 & -0.3851 & 0.5883 & 0.5168 \\ 0.1117 & 0.2484 & -0.5947 & 0.7564 \\ 0.8654 & 0.1852 & -0.2552 & -0.3894 \\ 0.0000 & 0.8693 & 0.4849 & 0.0957 \end{bmatrix}$$

with L, U obtained from the same MATLAB code as above. Thus, we identify,

$$S_{\text{new}} = \begin{bmatrix} 1.1555 & 0 \\ 2.0648 & 1.1503 \end{bmatrix} \quad \Rightarrow \quad P_{\text{new}} = S_{\text{new}}S_{\text{new}}^T = \begin{bmatrix} 1.3352 & 2.3859 \\ 2.3859 & 5.5867 \end{bmatrix}$$

For the direct method of Eq. (13.11.8), we calculate the gain $K = AG$,

$$K = \begin{bmatrix} 0.1568 & -0.0301 \\ 0.1849 & 0.0115 \end{bmatrix}$$

and the matrix:

$$\big[(A - KC)S,\, K\bar{R},\, \bar{Q}\big] = \left[\begin{array}{cc|cc|cc} 0.4024 & -0.0392 & 0.4100 & -0.0603 & 1 & 0 \\ 0.4033 & 0.3000 & 0.5775 & 0.0229 & 2 & 1 \end{array}\right]$$

$$= \left[\begin{array}{cc|cc|cc} 1.1555 & 0 & 0 & 0 & 0 & 0 \\ 2.0648 & 1.1503 & 0 & 0 & 0 & 0 \end{array}\right] U^T = [S_{\text{new}},\, 0, 0]U^T$$

which generates the same S_{new} as the two-step procedure. □

Using the Joseph forms in conjunction with the square root factorizations provides, in effect, double protection at the expense of increased computation. There exist a variety of other square-root algorithms [863,865,888–896], including some for the smoothing problem. One of the standard ones [889] employs the triangularization:

$$\begin{bmatrix} \bar{R} & CS \\ 0 & S \end{bmatrix} = \begin{bmatrix} \bar{D} & 0 \\ PC^T\bar{D}^{-T} & S_f \end{bmatrix} U^T \qquad (13.11.9)$$

where $\bar{D}$ is a lower-triangular square root factor of $D = CPC^T + R = \bar{D}\bar{D}^T$. Its correctness can be verified by noting that $P_f = P - PC^TD^{-1}CP$ and by forming the products:

$$\begin{bmatrix} \bar{R} & CS \\ 0 & S \end{bmatrix} \begin{bmatrix} \bar{R} & CS \\ 0 & S \end{bmatrix}^T = \begin{bmatrix} D & CP \\ PC^T & P \end{bmatrix}$$

$$\begin{bmatrix} \bar{D} & 0 \\ PC^T\bar{D}^{-T} & S_f \end{bmatrix} U^TU \begin{bmatrix} \bar{D} & 0 \\ PC^T\bar{D}^{-T} & S_f \end{bmatrix}^T = \begin{bmatrix} D & CP \\ PC^T & P_f + PC^TD^{-1}CP \end{bmatrix}$$

The triangularization operation produces $\bar{D}$, S_f, and $PC^T\bar{D}^{-T}$, the latter being part of the required Kalman gain $G = PC^TD^{-1} = [PC^T\bar{D}^{-T}]\bar{D}^{-1}$, where the division by the lower-triangular matrix $\bar{D}$ is an efficient operation. Therefore, the computation of the filtered estimate can also be done efficiently:

$$\hat{\mathbf{x}}_f = \hat{\mathbf{x}} + G(\mathbf{y} - C\hat{\mathbf{x}}) = \hat{\mathbf{x}} + \big[PC^T\bar{D}^{-T}\big]\bar{D}^{-1}(\mathbf{y} - C\hat{\mathbf{x}})$$

A direct-updating version [891] is possible in this case too by the triangularization of the following matrix:

$$\begin{bmatrix} \bar{R} & CS & 0 \\ 0 & AS & \bar{Q} \end{bmatrix} = \begin{bmatrix} \bar{D} & 0 & 0 \\ APC^T\bar{D}^{-T} & S_{\text{new}} & 0 \end{bmatrix} U^T$$

Its correctness follows from $P_{\text{new}} = S_{\text{new}}S_{\text{new}}^T = APA^T - APC^TD^{-1}CPA^T + Q$ and by comparing the products:

$$\begin{bmatrix} \bar{R} & CS & 0 \\ 0 & AS & \bar{Q} \end{bmatrix} \begin{bmatrix} \bar{R}^T & 0 \\ S^TC^T & S^TA^T \\ 0 & \bar{Q}^T \end{bmatrix} = \begin{bmatrix} D & CPA^T \\ APC^T & APA^T + Q \end{bmatrix}$$

$$\begin{bmatrix} \bar{D} & 0 & 0 \\ APC^T\bar{D}^{-T} & S_{\text{new}} & 0 \end{bmatrix} U^TU \begin{bmatrix} \bar{D}^T & \bar{D}^{-1}CPA^T \\ 0 & S_{\text{new}}^T \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} D & CPA^T \\ APC^T & P_{\text{new}} + APC^TD^{-1}CPA^T \end{bmatrix}$$

This factorization allows also the efficient computation of the predicted estimate:

$$\hat{\mathbf{x}}_{\text{new}} = A\hat{\mathbf{x}} + \big[APC^T\bar{D}^{-T}\big]\bar{D}^{-1}(\mathbf{y} - C\hat{\mathbf{x}})$$

Restoring the time indices, we summarize the two-step algorithm for the computation of the square-root factors of the covariances and the estimates:

$$\boxed{\begin{aligned}
&\begin{bmatrix} \bar{R}_n & C_nS_{n/n-1} \\ 0 & S_{n/n-1} \end{bmatrix} = \begin{bmatrix} \bar{D}_n & 0 \\ P_{n/n-1}C_n^T\bar{D}_n^{-T} & S_{n/n} \end{bmatrix} U^T \\
&\hat{\mathbf{x}}_{n/n} = \hat{\mathbf{x}}_{n/n-1} + \big[P_{n/n-1}C_n^T\bar{D}_n^{-T}\big]\bar{D}_n^{-1}(\mathbf{y}_n - C_n\hat{\mathbf{x}}_{n/n-1}) \\
&\big[A_nS_{n/n},\, \bar{Q}_n\big] = \big[S_{n+1/n},\, 0\big]U^T \\
&\hat{\mathbf{x}}_{n+1/n} = A_n\hat{\mathbf{x}}_{n/n}
\end{aligned}} \tag{13.11.10}$$

and for the direct method:

$$\boxed{\begin{aligned}
&\begin{bmatrix} \bar{R}_n & C_nS_{n/n-1} & 0 \\ 0 & A_nS_n & \bar{Q}_n \end{bmatrix} = \begin{bmatrix} \bar{D}_n & 0 & 0 \\ A_nP_{n/n-1}C_n^T\bar{D}_n^{-T} & S_{n+1/n} & 0 \end{bmatrix} U^T \\
&\hat{\mathbf{x}}_{n+1/n} = A_n\hat{\mathbf{x}}_{n/n-1} + \big[A_nP_{n/n-1}C_n^T\bar{D}_n^{-T}\big]\bar{D}_n^{-1}(\mathbf{y}_n - C_n\hat{\mathbf{x}}_{n/n-1})
\end{aligned}} \tag{13.11.11}$$

Example 13.11.3: For the model defined in Example 13.11.2 and the given starting S, we carry out the triangularization of the following matrix using the QR-factorization:

$$\begin{bmatrix} \bar{R} & CS \\ 0 & S \end{bmatrix} = \left[\begin{array}{cc|cc} 3 & 0 & 3.2004 & 1.4731 \\ 2 & 2 & 1.8820 & 1.4731 \\ \hline 0 & 0 & 1.3184 & 0 \\ 0 & 0 & 1.8820 & 1.4731 \end{array}\right]$$

$$= \left[\begin{array}{cc|cc} 4.6274 & 0 & 0 & 0 \\ 3.0672 & 2.0746 & 0 & 0 \\ \hline 0.9119 & -0.1521 & 0.9400 & 0 \\ 1.7706 & 0.1355 & 0.9440 & 1.2913 \end{array}\right] U^T = \begin{bmatrix} \bar{D} & 0 \\ PC^T\bar{D}^{-T} & S_f \end{bmatrix} U^T$$

from which we extract:

$$\bar{D} = \begin{bmatrix} 4.6274 & 0 \\ 3.0672 & 2.0746 \end{bmatrix}, \quad PC^T\bar{D}^{-T} = \begin{bmatrix} 0.9119 & -0.1521 \\ 1.7706 & 0.1355 \end{bmatrix}, \quad S_f = \begin{bmatrix} 0.9400 & 0 \\ 0.9440 & 1.2913 \end{bmatrix}$$

Using the quantities D, P computed in Example 13.11.2, we verify the factorization:

$$D = \begin{bmatrix} 21.4126 & 14.1931 \\ 14.1931 & 13.7118 \end{bmatrix} = \begin{bmatrix} 4.6274 & 0 \\ 3.0672 & 2.0746 \end{bmatrix} \begin{bmatrix} 4.6274 & 0 \\ 3.0672 & 2.0746 \end{bmatrix}^T = \bar{D}\bar{D}^T$$

Similarly, we may verify the correctness of $PC^T\bar{D}^{-T}$ and S_f. Next, we illustrate the direct method. We form the following matrix and triangularize it by applying the QR-factorization to its transpose:

$$\begin{bmatrix} \bar{R} & CS & 0 \\ 0 & AS & \bar{Q} \end{bmatrix} = \left[\begin{array}{cc|cc|cc} 3 & 0 & 3.2004 & 1.4731 & 0 & 0 \\ 2 & 2 & 1.8820 & 1.4731 & 0 & 0 \\ \hline 0 & 0 & 0.8474 & 0.1473 & 1 & 0 \\ 0 & 0 & 1.0165 & 0.5892 & 2 & 1 \end{array}\right]$$

$$= \left[\begin{array}{cc|cc|cc} 4.6274 & 0 & 0 & 0 & 0 & 0 \\ 3.0672 & 2.0746 & 0 & 0 & 0 & 0 \\ \hline 0.6330 & -0.0625 & 1.1555 & 0 & 0 & 0 \\ 0.8906 & 0.0238 & 2.0648 & 1.1503 & 0 & 0 \end{array}\right] U^T = \begin{bmatrix} \bar{D} & 0 & 0 \\ APC^T\bar{D}^{-T} & S_{\text{new}} & 0 \end{bmatrix} U^T$$

and we extract the same $\bar{D}$ as above and the same S_{new} as in Example 13.11.2. □

13.12 Maximum Likelihood Parameter Estimation

One issue in applying the Kalman filter is the determination of the state-space model parameters $\{A, C, Q, R\}$ and the initial values $\{\bar{\mathbf{x}}_0, \Sigma_0\}$. If the dynamics is known, as for example in α–β radar tracking, then $\{A, C\}$ are known, but not necessarily the noise covariances $\{Q, R\}$ or the initial values. If the dynamics is not known, as for example in the so-called unobserved components models for microeconomic applications, then all the parameters must be estimated.

Maximum likelihood (ML) is a commonly used method for estimating the model parameters. Assuming a time-invariant model, then given a set of $N + 1$ observations, $Y = \{\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_N\}$, one determines the parameters $\{A, C, Q, R, \bar{\mathbf{x}}_0, \Sigma_0\}$ that maximize the joint probability density $p(\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_N)$ or $p(Y)$ for short. Equivalently, one may minimize the negative of the log-likelihood function:

$$L(Y) = -\log p(Y) = \min \tag{13.12.1}$$

This problem becomes tractable under the gaussian assumption for the state-space model. The Kalman filtering algorithm generates the equivalent orthogonalized observation basis of innovations $Y = \{\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_N\} = \{\boldsymbol{\varepsilon}_0, \boldsymbol{\varepsilon}_1, \dots, \boldsymbol{\varepsilon}_N\}$, which are mutually uncorrelated and gaussian, and hence, mutually independent. Therefore, the joint density factors into the marginal densities:

$$p(\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_N) = p(\boldsymbol{\varepsilon}_0, \boldsymbol{\varepsilon}_1, \dots, \boldsymbol{\varepsilon}_N) = \prod_{n=0}^{N} p(\boldsymbol{\varepsilon}_n) = \prod_{n=0}^{N} \frac{\exp(-\boldsymbol{\varepsilon}_n^T D_n^{-1} \boldsymbol{\varepsilon}_n / 2)}{(2\pi)^{r/2} (\det D_n)^{1/2}}$$

where $r = \dim(\mathbf{y}_n)$ and D_n is the covariance of $\boldsymbol{\varepsilon}_n$ generated by the Kalman filtering algorithm. Thus, the log-likelihood function can be expressed up to a constant by [897]:

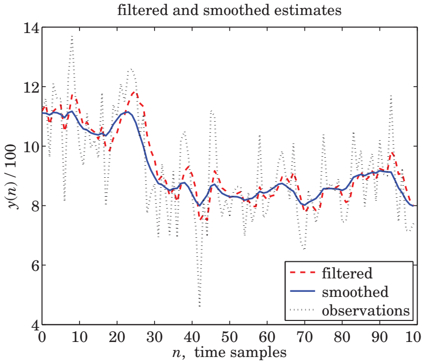
$$L(Y) = -\log p(Y) = \frac{1}{2} \sum_{n=0}^{N} \Big[\log(\det D_n) + \boldsymbol{\varepsilon}_n^T D_n^{-1} \boldsymbol{\varepsilon}_n \Big] + \text{const.} \tag{13.12.2}$$

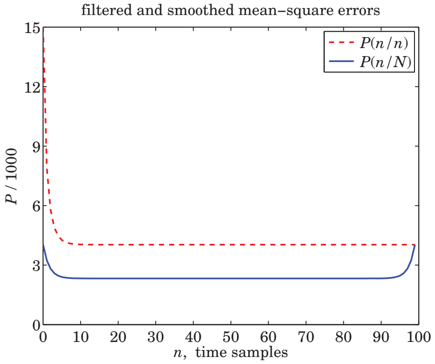
Let θ denote all or a subset of the model parameters $\{A, C, Q, R, \bar{\mathbf{x}}_0, \Sigma_0\}$ that one wishes to estimate. The dependence of $L(Y)$ on θ will be indicated explicitly by $L_\theta(Y)$.

There exist several methods [897–911] of minimizing $L_\theta(Y)$, some requiring only the evaluation of the function $L_\theta(Y)$ for various values θ, some requiring also the first, and some the second, derivatives of $L_\theta(Y)$ with respect to θ. The EM algorithm is an alternative, iterative, minimization method and we discuss it in the next section.

The MATLAB function `kfilt` can evaluate $L_\theta(Y)$ at any θ, and thus, it can be used, in conjunction with the built-in function `fminsearch` of MATLAB's optimization toolbox to perform the minimization. This function uses the Nelder-Mead simplex direct search method in which the derivatives of L_θ are not required. We illustrate the approach with two examples.

Example 13.12.1: The Nile River data set has been used as a benchmark in a variety of statistical methods [912]. It represents the annual volume of the Nile River (discharged at Aswan, in units of 10^8 m^3) from 1871 to 1970. It is depicted in the left figure below (dotted line of observations).





Following [903], we model it as a local-level (random walk in noise) model:

$$\begin{aligned} x_{n+1} &= x_n + w_n \\ y_n &= x_n + v_n \end{aligned}$$

with noise variances $Q = \sigma_w^2$ and $R = \sigma_v^2$ to be estimated, so that the parameter vector is $\theta = [\sigma_w^2, \sigma_v^2]^T$. The MATLAB code below defines the function L_θ with the help of `kfilt`, with the Kalman filter initialized to the arbitrary values $\bar{x}_0 = 0$ and $\Sigma_0 = 10^7$. It then calls `fminsearch` with the arbitrary initial values of the parameters $\theta_0 = [1, 1]^T$, and returns the "optimum" values:

$$\theta = \begin{bmatrix} \sigma_w^2 \\ \sigma_v^2 \end{bmatrix} = \begin{bmatrix} 1468.5 \\ 15099.7 \end{bmatrix}$$

```
y = [ 1120  1160   963  1210  1160  1160   813  1230  1370  1140 ...
       995   935  1110   994  1020   960  1180   799   958  1140 ...
      1100  1210  1150  1250  1260  1220  1030  1100   774   840 ...
       874   694   940   833   701   916   692  1020  1050   969 ...
       831   726   456   824   702  1120  1100   832   764   821 ...
       768   845   864   862   698   845   744   796  1040   759 ...
       781   865   845   944   984   897   822  1010   771   676 ...
       649   846   812   742   801  1040   860   874   848   890 ...
       744   749   838  1050   918   986   797   923   975   815 ...
      1020   906   901  1170   912   746   919   718   714   740 ];

A = 1; C = 1; x0 = 0; S0 = 1e7;                   % fixed model parameters

L = @(th) kfilt(A, C, th(1), th(2), y, x0, S0);   % likelihood function

th0 = [1; 1];                                     % initialize search
th = fminsearch(L, th0);                          % Nelder-Mead search

Q = th(1); R = th(2);                             % estimated Q,R

[Lmin,x,P,xf,Pf] = kfilt(A,C,Q,R,y,x0,S0);        % run Kalman filter
[Lmin,xs,Ps] = ksmooth(A,C,Q,R,y,x0,S0);          % run Kalman smoother

t = 0:length(y)-1;
figure; plot(t,xf/100,'--',t,xs/100,'-' ,t,y/100,'k:');
figure; plot(t,Pf(:)/1e3,'--',t,Ps(:)/1e3,'-');
```

The Kalman filter and smoother are then run with the optimum parameter values for Q, R, and the resulting filtered and smoothed estimates, $\hat{x}_{n/n}$ and $\hat{x}_{n/N}$ are plotted on the left figure. The right figure plots the corresponding mean-square errors, $P_{n/n}$ and $P_{n/N}$. We note that $P_{n/N}$ is smaller than $P_{n/n}$ since it uses all the observations. It rises up at the end to agree with the filtered error, that is, $P_{n/n} = P_{n/N}$, when $n = N$. □

Example 13.12.2: Consider the α–β tracking model discussed in Example 13.7.3 and defined by the state-space model:

$$\begin{bmatrix} x_{n+1} \\ \dot{x}_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_n \\ \dot{x}_n \end{bmatrix} + \begin{bmatrix} T^2/2 \\ T \end{bmatrix} a_n \,, \quad y_n = [1, 0] \begin{bmatrix} x_n \\ \dot{x}_n \end{bmatrix} + v_n$$

with model matrices:

$$A=\begin{bmatrix}1 & T\\ 0 & 1\end{bmatrix},\quad C=[1,0]\,,\quad Q=\begin{bmatrix}T^4/4 & T^3/2\\ T^3/2 & T^2\end{bmatrix}\sigma_a^2\equiv Q_T\,\sigma_a^2\,,\quad R=\sigma_v^2$$

The model is simulated with the following values of the parameters $T=1$, $\sigma_a=0.02$, $\sigma_v=2$, and generating $N+1$ noisy position measurements, y_n, $0\le n\le N$, where $N=300$, starting with the initial state-vector $\bar{\mathbf{x}}_0=[0,0.1]^T$. The generated observations y_n are then used to estimate the model parameters $\theta=[\sigma_a,\sigma_v]^T$ starting from the initial values $\theta_0=[0,0]^T$. The likelihood function is defined as in the previous example, with the Kalman filter run with the initial values:

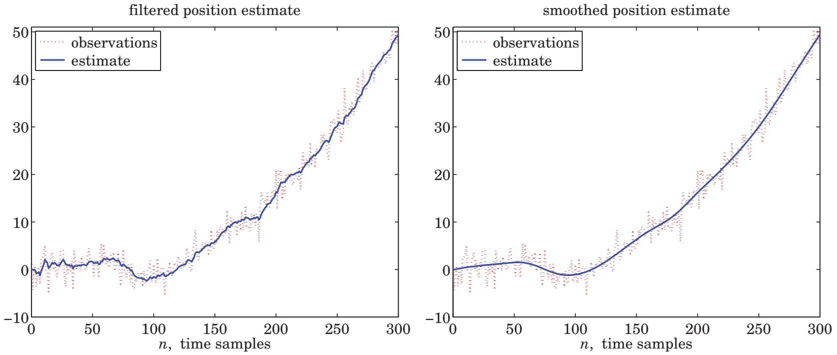
$$\bar{\mathbf{x}}_0=\begin{bmatrix}0\\ 0.1\end{bmatrix},\quad \Sigma_0=\begin{bmatrix}0.1 & 0\\ 0 & 0.1\end{bmatrix}$$

The resulting estimated parameters returned from `mfinsearch`, and the corresponding estimated covariance matrices Q,R, are:

$$\theta=\begin{bmatrix}\sigma_a\\ \sigma_v\end{bmatrix}=\begin{bmatrix}-0.0199\\ 2.0237\end{bmatrix}\quad\Rightarrow\quad Q=Q_T\,\sigma_a^2\,,\quad R=\sigma_v^2$$

The absolute value $|\sigma_a|$ is close to the true value of 0.02. The sign does not matter since Q,R depend on the squares of σ_a,σ_v.

The Kalman filter and smoother are then run with the estimated Q,R and the resulting filtered and smoothed estimates are shown in the figure below.



The MATLAB code used to generate these graphs is as follows:

```
N = 301; T = 1; Tmax = (N-1)*T; t = 0:T:Tmax;

sa = 0.02; sv = 2;
A = [1, T; 0, 1]; C = [1, 0];                        % model parameters
QT = [T^4/4, T^3/2; T^3/2, T^2];
Q = QT*sa^2; R = sv^2;

seed = 100;  randn('state',seed);                    % generate noise inputs
v = sv * randn(1,length(t));
w = [T^2/2; T] * sa * randn(1,length(t));
```

```
x0 = [0; 0.1]; S0 = 0.1 * eye(2);

x(:,1) = x0;
for n=1:N-1                                          % generate model signals
   x(:,n+1) = A*x(:,n) + w(n);
   y(n) = C*x(:,n) + v(n);
end
   y(N) = C*x(:,N) + v(N);

L = @(th) kfilt(A,C, QT * th(1)^2, th(2)^2, y,x0,S0);     % likelihood

th0 = [0, 0]';                                            % initialize search
th = fminsearch(L, th0);                                  % Nelder-Mead search

Q = QT * th(1)^2; R = th(2)^2;                            % estimated Q,R

[Lmin,X,P,Xf,Pf] = kfilt(A, C, Q, R, y, x0, S0);          % run Kalman filter
[Lmin,Xs,Ps] = ksmooth(A, C, Q, R, y, x0, S0);            % run Kalman smoother

figure; plot(t,y,':', t,Xf(1,:),'-');                     % plot position only
figure; plot(t,y,':', t,Xs(1,:),'-');
```

13.13 Parameter Estimation with the EM Algorithm

The application of the Expectation-Maximization (EM) algorithm [905,906] to the estimation of the state-space model parameters has been discussed in [909–911].

We begin by developing a solution for the model parameters $\{A,C,Q,R,\bar{\mathbf{x}}_0,\Sigma_0\}$ in terms of the signals $\mathbf{x}_n,\mathbf{y}_n$ of the state-space model, then convert that into a computable iterative algorithm, and then show that it is equivalent to the EM algorithm. The approach allows us to handle also the case of a noise covariance matrix Q of less than full rank. We start with a standard time-invariant state-space model iterated over the time interval $0\le n\le N$:

$$\begin{aligned}\mathbf{x}_{n+1}&=A\,\mathbf{x}_n+\mathbf{w}_n\\ \mathbf{y}_n&=C\,\mathbf{x}_n+\mathbf{v}_n\end{aligned}\tag{13.13.1}$$

with start-up value $\mathbf{x}_0$ with mean and covariance $\bar{\mathbf{x}}_0,\Sigma_0$. The noise covariances are:

$$\begin{gathered}E[\mathbf{w}_n\mathbf{w}_i^T]=Q\delta_{ni}\,,\quad E[\mathbf{v}_n\mathbf{v}_i^T]=R\delta_{ni}\,,\quad E[\mathbf{w}_n\mathbf{v}_i^T]=0\\ E[\mathbf{w}_n\mathbf{x}_0^T]=0\,,\quad E[\mathbf{v}_n\mathbf{x}_0^T]=0\end{gathered}\tag{13.13.2}$$

These assumptions imply that $E[\mathbf{v}_n\mathbf{x}_n^T]=0$, which leads to

$$E[\mathbf{v}_n\mathbf{x}_n^T]=E[(\mathbf{y}_n-C\mathbf{x}_n)\mathbf{x}_n^T]=0\quad\Rightarrow\quad E[\mathbf{y}_n\mathbf{x}_n^T]=CE[\mathbf{x}_n\mathbf{x}_n^T]$$

and using this result, we obtain:

$$\begin{aligned}R&=E[\mathbf{v}_n\mathbf{v}_n^T]=E[(\mathbf{y}_n-C\mathbf{x}_n)(\mathbf{y}_n-C\mathbf{x}_n)^T]=\\ &=E[\mathbf{y}_n\mathbf{y}_n^T]-CE[\mathbf{x}_n\mathbf{y}_n^T]-E[\mathbf{y}_n\mathbf{x}_n^T]C^T+CE[\mathbf{x}_n\mathbf{x}_n^T]C^T=E[\mathbf{y}_n\mathbf{y}_n^T]-CE[\mathbf{x}_n\mathbf{y}_n^T]\end{aligned}$$

Similarly, using $E[\mathbf{w}_n\mathbf{x}_n^T]=0$, we find:

$$
\begin{aligned}
E[\mathbf{w}_n\mathbf{x}_n^T] &= E\big[(\mathbf{x}_{n+1} - A\mathbf{x}_n)\mathbf{x}_n^T\big] = 0 \quad\Rightarrow\quad E[\mathbf{x}_{n+1}\mathbf{x}_n^T] = AE[\mathbf{x}_n\mathbf{x}_n^T]\\
Q &= E[\mathbf{w}_n\mathbf{w}_n^T] = E\big[(\mathbf{x}_{n+1} - A\mathbf{x}_n)\,(\mathbf{x}_{n+1} - A\mathbf{x}_n)^T\big] =\\
&\quad = E[\mathbf{x}_{n+1}\mathbf{x}_{n+1}^T] - AE[\mathbf{x}_n\mathbf{x}_{n+1}^T] - E[\mathbf{x}_{n+1}\mathbf{x}_n^T]A^T + AE[\mathbf{x}_n\mathbf{x}_n^T]A^T\\
&\quad = E[\mathbf{x}_{n+1}\mathbf{x}_{n+1}^T] - AE[\mathbf{x}_n\mathbf{x}_{n+1}^T]
\end{aligned}
$$

We collect the above together,

$$
\begin{aligned}
E[\mathbf{y}_n\mathbf{x}_n^T] &= CE[\mathbf{x}_n\mathbf{x}_n^T]\\
R &= E[\mathbf{y}_n\mathbf{y}_n^T] - CE[\mathbf{x}_n\mathbf{y}_n^T] - E[\mathbf{y}_n\mathbf{x}_n^T]C^T + CE[\mathbf{x}_n\mathbf{x}_n^T]C^T =\\
&= E[\mathbf{y}_n\mathbf{y}_n^T] - CE[\mathbf{x}_n\mathbf{y}_n^T]\\
E[\mathbf{x}_{n+1}\mathbf{x}_n^T] &= AE[\mathbf{x}_n\mathbf{x}_n^T]\\
Q &= E[\mathbf{x}_{n+1}\mathbf{x}_{n+1}^T] - AE[\mathbf{x}_n\mathbf{x}_{n+1}^T] - E[\mathbf{x}_{n+1}\mathbf{x}_n^T]A^T + AE[\mathbf{x}_n\mathbf{x}_n^T]A^T =\\
&= E[\mathbf{x}_{n+1}\mathbf{x}_{n+1}^T] - AE[\mathbf{x}_n\mathbf{x}_{n+1}^T]
\end{aligned}
\tag{13.13.3}
$$

Since these are valid for each n in the interval $0 \le n \le N$, they will also be valid if we form the average sums over the same interval, for example,

$$
\sum_{n=0}^{N} E[\mathbf{y}_n\mathbf{x}_n^T] = C\sum_{n=0}^{N} E[\mathbf{x}_n\mathbf{x}_n^T] \quad\text{and}\quad \sum_{n=0}^{N-1} E[\mathbf{x}_{n+1}\mathbf{x}_n^T] = A\sum_{n=0}^{N-1} E[\mathbf{x}_n\mathbf{x}_n^T]
$$

This leads us to define the average quantities:

$$
\begin{aligned}
U_{xx} &= \frac{1}{N+1}\sum_{n=0}^{N} E[\mathbf{x}_n\mathbf{x}_n^T] & V_{xx} &= \frac{1}{N}\sum_{n=0}^{N-1} E[\mathbf{x}_n\mathbf{x}_n^T]\\
U_{yx} &= \frac{1}{N+1}\sum_{n=0}^{N} E[\mathbf{y}_n\mathbf{x}_n^T] & V_{x_1x} &= \frac{1}{N}\sum_{n=0}^{N-1} E[\mathbf{x}_{n+1}\mathbf{x}_n^T]\\
U_{yy} &= \frac{1}{N+1}\sum_{n=0}^{N} E[\mathbf{y}_n\mathbf{y}_n^T] & V_{x_1x_1} &= \frac{1}{N}\sum_{n=0}^{N-1} E[\mathbf{x}_{n+1}\mathbf{x}_{n+1}^T]
\end{aligned}
\tag{13.13.4}
$$

with $U_{xy} = U_{yx}^T$ and $V_{xx_1} = V_{x_1x}^T$. Then, the summed form of Eqs. (13.13.3) read:

$$
\begin{aligned}
U_{yx} &= CU_{xx}\\
R &= U_{yy} - CU_{xy} - U_{yx}C^T + CU_{xx}C^T = U_{yy} - CU_{xy}\\
V_{x_1x} &= AV_{xx}\\
Q &= V_{x_1x_1} - AV_{xx_1} - V_{x_1x}A^T + AV_{xx}A^T = V_{x_1x_1} - AV_{xx_1}
\end{aligned}
$$

which may be solved for the model parameters:

$$
\boxed{
\begin{aligned}
C &= U_{yx}U_{xx}^{-1}\\
R &= U_{yy} - CU_{xy}\\
A &= V_{x_1x}V_{xx}^{-1}\\
Q &= V_{x_1x_1} - AV_{xx_1}
\end{aligned}
}
\tag{13.13.5}
$$

If A, C are known, then one can compute R, Q from the alternative quadratic expressions, which guarantee the (semi) positive-definiteness property of R, Q:

$$
\boxed{
\begin{aligned}
R &= U_{yy} - CU_{xy} - U_{yx}C^T + CU_{xx}C^T\\
Q &= V_{x_1x_1} - AV_{xx_1} - V_{x_1x}A^T + AV_{xx}A^T
\end{aligned}
}
\tag{13.13.6}
$$

The above can be turned into an iterative estimation algorithm as follows. We start with the set of observations, $Y = \{\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_N\}$, and an initial choice for the model parameters, say,

$$
\theta^{\text{old}} = \{A^{\text{old}}, C^{\text{old}}, Q^{\text{old}}, R^{\text{old}}, \bar{\mathbf{x}}_0^{\text{old}}, \Sigma_0^{\text{old}}\}
$$

The Kalman smoother is run on the data Y using this set of model parameters, and the estimated smoothed state vectors $\hat{\mathbf{x}}_{n/N}$ and corresponding mean-square error covariances are computed. The expectation values in Eq. (13.13.4) are then replaced by the conditional ones based on Y and θ^{old}, that is,

$$
\begin{aligned}
\hat{U}_{xx} &= \frac{1}{N+1}\sum_{n=0}^{N} E[\mathbf{x}_n\mathbf{x}_n^T\,|\,Y] & \hat{V}_{xx} &= \frac{1}{N}\sum_{n=0}^{N-1} E[\mathbf{x}_n\mathbf{x}_n^T\,|\,Y]\\
\hat{U}_{yx} &= \frac{1}{N+1}\sum_{n=0}^{N} E[\mathbf{y}_n\mathbf{x}_n^T\,|\,Y] & \hat{V}_{x_1x} &= \frac{1}{N}\sum_{n=0}^{N-1} E[\mathbf{x}_{n+1}\mathbf{x}_n^T\,|\,Y]\\
\hat{U}_{yy} &= \frac{1}{N+1}\sum_{n=0}^{N} E[\mathbf{y}_n\mathbf{y}_n^T\,|\,Y] & \hat{V}_{x_1x_1} &= \frac{1}{N}\sum_{n=0}^{N-1} E[\mathbf{x}_{n+1}\mathbf{x}_{n+1}^T\,|\,Y]
\end{aligned}
\tag{13.13.7}
$$

Using the orthogonal decomposition for the smoothed states, we have:

$$
\begin{aligned}
\mathbf{x}_n &= \hat{\mathbf{x}}_{n/N} + \mathbf{e}_{n/N}\,, & & E[\hat{\mathbf{x}}_{n/N}\mathbf{e}_{n/N}^T] = 0\\
\mathbf{x}_{n+1} &= \hat{\mathbf{x}}_{n+1/N} + \mathbf{e}_{n+1/N}\,, & & E[\hat{\mathbf{x}}_{n+1/N}\mathbf{e}_{n+1/N}^T] = 0\,,\quad E[\hat{\mathbf{x}}_{n/N}\mathbf{e}_{n+1/N}^T] = 0
\end{aligned}
$$

which give:

$$
\begin{aligned}
E[\mathbf{x}_n\mathbf{x}_n^T] &= E[\hat{\mathbf{x}}_{n/N}\hat{\mathbf{x}}_{n/N}^T] + E[\mathbf{e}_{n/N}\mathbf{e}_{n/N}^T]\\
E[\mathbf{x}_{n+1}\mathbf{x}_n^T] &= E[\hat{\mathbf{x}}_{n+1/N}\hat{\mathbf{x}}_{n/N}^T] + E[\mathbf{e}_{n+1/N}\mathbf{e}_{n/N}^T]\\
E[\mathbf{x}_{n+1}\mathbf{x}_{n+1}^T] &= E[\hat{\mathbf{x}}_{n+1/N}\hat{\mathbf{x}}_{n+1/N}^T] + E[\mathbf{e}_{n+1/N}\mathbf{e}_{n+1/N}^T]
\end{aligned}
$$

Replacing these by the conditional expectations, we obtain:

$$
\begin{aligned}
&E[\mathbf{x}_n\mathbf{x}_n^T\,|\,Y]=E[\hat{\mathbf{x}}_{n/N}\hat{\mathbf{x}}_{n/N}^T\,|\,Y]+E[\mathbf{e}_{n/N}\mathbf{e}_{n/N}^T\,|\,Y]=\hat{\mathbf{x}}_{n/N}\hat{\mathbf{x}}_{n/N}^T+P_{n/N}\\
&E[\mathbf{x}_{n+1}\mathbf{x}_n^T\,|\,Y]=E[\hat{\mathbf{x}}_{n+1/N}\hat{\mathbf{x}}_{n/N}^T\,|\,Y]+E[\mathbf{e}_{n+1/N}\mathbf{e}_{n/N}^T\,|\,Y]=\hat{\mathbf{x}}_{n+1/N}\hat{\mathbf{x}}_{n/N}^T+V_{n+1,n}\\
&E[\mathbf{x}_{n+1}\mathbf{x}_{n+1}^T\,|\,Y]=E[\hat{\mathbf{x}}_{n+1/N}\hat{\mathbf{x}}_{n+1/N}^T\,|\,Y]+E[\mathbf{e}_{n+1/N}\mathbf{e}_{n+1/N}^T\,|\,Y]=\hat{\mathbf{x}}_{n+1/N}\hat{\mathbf{x}}_{n+1/N}^T+P_{n+1/N}
\end{aligned}
$$

where we set $V_{n+1,n}=E[\mathbf{e}_{n+1/N}\mathbf{e}_{n/N}^T\,|\,Y]$, given by Eq. (13.10.25). Similarly, we have:

$$
\begin{aligned}
E[\mathbf{y}_n\mathbf{x}_n^T\,|\,Y]&=\mathbf{y}_n\hat{\mathbf{x}}_{n/N}\\
E[\mathbf{y}_n\mathbf{y}_n^T\,|\,Y]&=\mathbf{y}_n\mathbf{y}_n^T
\end{aligned}
$$

Thus, we may replace Eqs. (13.13.4) by their estimated versions based on Y and θ^{old}:

$$
\boxed{\begin{aligned}
\hat{U}_{xx}&=\frac{1}{N+1}\sum_{n=0}^{N}\big[\hat{\mathbf{x}}_{n/N}\hat{\mathbf{x}}_{n/N}^T+P_{n/N}\big]\\
\hat{U}_{yx}&=\frac{1}{N+1}\sum_{n=0}^{N}\mathbf{y}_n\hat{\mathbf{x}}_{n/N}^T\\
\hat{U}_{yy}&=\frac{1}{N+1}\sum_{n=0}^{N}\mathbf{y}_n\mathbf{y}_n^T
\end{aligned}}
\quad
\boxed{\begin{aligned}
\hat{V}_{xx}&=\frac{1}{N}\sum_{n=0}^{N-1}\big[\hat{\mathbf{x}}_{n/N}\hat{\mathbf{x}}_{n/N}^T+P_{n/N}\big]\\
\hat{V}_{x_1x}&=\frac{1}{N}\sum_{n=0}^{N-1}\big[\hat{\mathbf{x}}_{n+1/N}\hat{\mathbf{x}}_{n/N}^T+V_{n+1,n}\big]\\
\hat{V}_{x_1x_1}&=\frac{1}{N}\sum_{n=0}^{N-1}\big[\hat{\mathbf{x}}_{n+1/N}\hat{\mathbf{x}}_{n+1/N}^T+P_{n+1/N}\big]
\end{aligned}}
\tag{13.13.8}
$$

Eqs. (13.13.5) can be used now to compute a new set of model parameters:

$$
\boxed{\begin{aligned}
C^{\text{new}}&=\hat{U}_{yx}\hat{U}_{xx}^{-1}\\
R^{\text{new}}&=\hat{U}_{yy}-C^{\text{new}}\hat{U}_{xy}\\
A^{\text{new}}&=\hat{V}_{x_1x}\hat{V}_{xx}^{-1}\\
Q^{\text{new}}&=\hat{V}_{x_1x_1}-A^{\text{new}}\hat{V}_{xx_1}
\end{aligned}}
\tag{13.13.9}
$$

or, if A,C are known, only Q,R are updated:

$$
\boxed{\begin{aligned}
R^{\text{new}}&=\hat{U}_{yy}-C\hat{U}_{xy}-\hat{U}_{yx}C^T+C\hat{U}_{xx}C^T\\
Q^{\text{new}}&=\hat{V}_{x_1x_1}-A\hat{V}_{xx_1}-\hat{V}_{x_1x}A^T+A\hat{V}_{xx}A^T
\end{aligned}}
\tag{13.13.10}
$$

The initial values $\bar{\mathbf{x}}_0,\Sigma_0$ are also estimated in the same way:

$$
\begin{aligned}
\bar{\mathbf{x}}_0&=E[\mathbf{x}_0]\\
\Sigma_0&=E[(\mathbf{x}_0-\bar{\mathbf{x}}_0)(\mathbf{x}_0-\bar{\mathbf{x}}_0)^T]
\end{aligned}
\quad\Rightarrow\quad
\begin{aligned}
\bar{\mathbf{x}}_0^{\text{new}}&=E[\mathbf{x}_0\,|\,Y]=\hat{\mathbf{x}}_{0/N}\\
\hat{\Sigma}_0^{\text{new}}&=E[(\mathbf{x}_0-\bar{\mathbf{x}}_0)(\mathbf{x}_0-\bar{\mathbf{x}}_0)^T\,|\,Y]=P_{0/N}\,,\quad\text{or,}
\end{aligned}
$$

$$
\boxed{\bar{\mathbf{x}}_0^{\text{new}}=\hat{\mathbf{x}}_{0/N}\,,\quad\hat{\Sigma}_0^{\text{new}}=P_{0/N}}
\tag{13.13.11}
$$

Thus, Eqs. (13.13.9)–(13.13.11) define a new set of model parameters:

$$
\theta^{\text{new}}=\{A^{\text{new}},C^{\text{new}},Q^{\text{new}},R^{\text{new}},\bar{\mathbf{x}}_0^{\text{new}},\Sigma_0^{\text{new}}\}
$$

and the iteration can be repeated until convergence by monitoring the value of the likelihood function $L_\theta(Y)$. Of course, all or any subset of the model parameters θ may be iterated through this algorithm.

The outputs `Xs`, `Ps`, `V` of the MATLAB function `ksmooth` and can be used to efficiently calculate the quantities of Eq. (13.13.8). For example, we have in MATLAB notation:

$$
\begin{aligned}
\hat{U}_{xx}&=\frac{1}{N+1}\sum_{n=0}^{N}\big[\hat{\mathbf{x}}_{n/N}\hat{\mathbf{x}}_{n/N}^T+P_{n/N}\big]=\frac{1}{N+1}\big(X_s*X_s'+\text{sum}(P_s,3)\big)\\
\hat{U}_{yx}&=\frac{1}{N+1}\sum_{n=0}^{N}\mathbf{y}_n\hat{\mathbf{x}}_{n/N}^T=\frac{1}{N+1}(Y*X_s')\\
\hat{U}_{yy}&=\frac{1}{N+1}\sum_{n=0}^{N}\mathbf{y}_n\mathbf{y}_n^T=\frac{1}{N+1}(Y*Y')\\
\hat{V}_{xx}&=\frac{1}{N}\sum_{n=0}^{N-1}\big[\hat{\mathbf{x}}_{n/N}\hat{\mathbf{x}}_{n/N}^T+P_{n/N}\big]=\frac{1}{N}\big(X_{s0}*X_{s0}'+\text{sum}(P_{s0},3)\big)\\
\hat{V}_{x_1x}&=\frac{1}{N}\sum_{n=0}^{N-1}\big[\hat{\mathbf{x}}_{n+1/N}\hat{\mathbf{x}}_{n/N}^T+V_{n+1,n}\big]=\frac{1}{N}\big(X_{s1}*X_{s0}'+\text{sum}(V_0,3)\big)\\
\hat{V}_{x_1x_1}&=\frac{1}{N}\sum_{n=0}^{N-1}\big[\hat{\mathbf{x}}_{n+1/N}\hat{\mathbf{x}}_{n+1/N}^T+P_{n+1/N}\big]=\frac{1}{N}\big(X_{s1}*X_{s1}'+\text{sum}(P_{s1},3)\big)
\end{aligned}
$$

where we used the definitions:

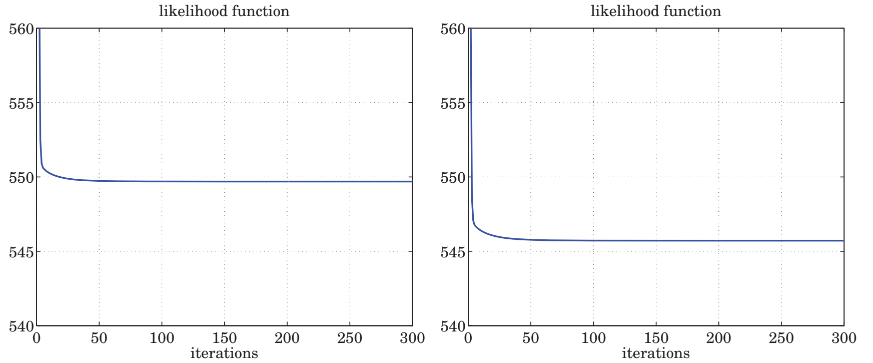
$$
\begin{aligned}
Y&=[\mathbf{y}_0,\mathbf{y}_1,\dots,\mathbf{y}_N]\\
X_s&=[\hat{\mathbf{x}}_{0/N},\,\hat{\mathbf{x}}_{1/N},\,\dots,\,\hat{\mathbf{x}}_{N-1/N},\,\hat{\mathbf{x}}_{N/N}]=[X_{s0},\,\hat{\mathbf{x}}_{N/N}]=[\hat{\mathbf{x}}_{0/N},\,X_{s1}]\\
X_{s0}&=[\hat{\mathbf{x}}_{0/N},\,\hat{\mathbf{x}}_{1/N},\,\dots,\,\hat{\mathbf{x}}_{N-1/N}]\\
X_{s1}&=[\hat{\mathbf{x}}_{1/N},\,\dots,\,\hat{\mathbf{x}}_{N-1/N},\,\hat{\mathbf{x}}_{N/N}]
\end{aligned}
\tag{13.13.12}
$$

and P_{s0},P_{s1},V_0 are sub-arrays of P_s and V matching the indexing of (13.13.12).

Example 13.13.1: We apply the estimation algorithm to the Nile River data of Example 13.12.1 by keeping A,C fixed, and only estimating $Q,R,\bar{x}_0,\Sigma_0$. We allow 300 iterations, with starting values chosen to be the same as those of Example 13.12.1. The following MATLAB code illustrates the implementation of the algorithm:

```
K = 300; N = length(y)-1;              % use same y as in Example 15.12.1
x0 = 0; S0 = 1e7; Q = 1; R = 1;        % initial values for the EM iteration
```

```
for i=1:K,                                      % EM iteration
   [L(i),Xs,Ps,V] = ksmooth(A,C,Q,R,y,x0,S0);   % E-step of the algorithm

   Uxx = (Xs*Xs' + sum(Ps,3))/(N+1);            % construct U,V matrices
   Uyx = (y*Xs')/(N+1);
   Uyy = (y*y')/(N+1);
   Vxx = (Xs(:,1:end-1)*Xs(:,1:end-1)' + sum(Ps(:,:,1:end-1),3))/N;
   V1x = (Xs(:,2:end)*Xs(:,1:end-1)' + sum(V(:,:,1:end-1),3))/N;
   V11 = (Xs(:,2:end)*Xs(:,2:end)' + sum(Ps(:,:,2:end),3))/N;

   R = Uyy - C*Uyx' - Uyx*C' + C*Uxx*C';        % M-step of the algorithm
   Q = V11 - A*V1x' - V1x*A' + A*Vxx*A';

   % x0 = Xs(:,1);        % uncomment to also estimate x0,S0
   % S0 = Ps(:,:,1);
end

k = 0:K-1; figure; plot(k,L);                   % plot likelihood function
```



The left graph shows the negative-log-likelihood function of Eq. (13.12.2) plotted versus iteration number for the case when only Q, R are being estimated, with $A, C, \bar{x}_0, \Sigma_0$ held fixed. The right graphs shows the case when $Q, R, \bar{x}_0, \Sigma_0$ are estimated. The estimated quantities at the last iteration are:

$$Q = 1468.5\,, \quad R = 15099.0$$

$$Q = 1294.7\,, \quad R = 15252.4\,, \quad \bar{x}_0 = 1118.4\,, \quad \Sigma_0 = 0.6$$

The results for the first case are to be compared with those of Example 13.12.1, that is, $Q = 1468.5$, $R = 15099.7$. The second case is different because four model parameters are being iterated, which lead to a smaller value of the likelihood function.

We note that in both cases, the likelihood function converges very fast initially, and then reaches a slowly-decreasing plateau. One could perhaps stop the iteration as soon the plateau is reached and use the slightly suboptimal estimates. □

The above estimation method can be extended to handle the case of a rank-defective noise covariance matrix Q. This arises for the following type of state-space model:

$$\begin{aligned} \mathbf{x}_{n+1} &= A\mathbf{x}_n + B\mathbf{w}_n \\ \mathbf{y}_n &= C\mathbf{x}_n + \mathbf{v}_n \end{aligned} \tag{13.13.13}$$

where $\mathbf{x}_n$ is p-dimensional and B is a $p\times q$ matrix with $q < p$, assumed to have full rank (i.e., q), and $\mathbf{w}_n$ is a q-dimensional white-noise signal with non-singular $q\times q$ covariance matrix Q_w. The corresponding Q that appears in the Kalman filtering algorithm is then $Q = BQ_wB^T$, and has rank q.

Assuming a known B, the iterative algorithm can be modified to estimate Q_w. Given an initial Q_w^{old}, we calculate $Q^{\text{old}} = BQ_w^{\text{old}}B^T$, and then carry out the usual iteration step to determine Q^{new} from Eq. (13.13.9) or (13.13.10). To find Q_w^{new}, we solve the equation $BQ_w^{\text{new}}B^T = Q^{\text{new}}$ using the pseudoinverse of B, that is, $B^+ = (B^TB)^{-1}B^T$,

$$Q_w^{\text{new}} = B^+Q^{\text{new}}B^{+T} \tag{13.13.14}$$

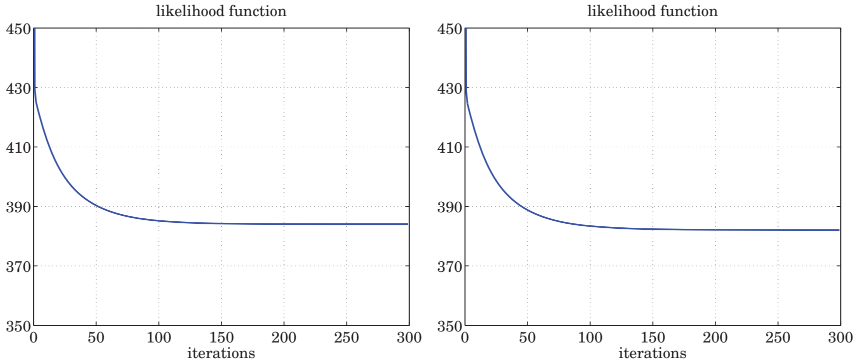
Example 13.13.2: Consider the α–β tracking model discussed in Example 13.12.2 and defined by the state-space model:

$$\begin{bmatrix} x_{n+1} \\ \dot{x}_{n+1} \end{bmatrix} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_n \\ \dot{x}_n \end{bmatrix} + \begin{bmatrix} T^2/2 \\ T \end{bmatrix} a_n\,, \quad y_n = [1, 0] \begin{bmatrix} x_n \\ \dot{x}_n \end{bmatrix} + v_n$$

with model matrices:

$$A = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \quad C = [1, 0]\,, \quad B = \begin{bmatrix} T^2/2 \\ T \end{bmatrix}, \quad Q_w = \sigma_a^2\,, \quad Q = BQ_wB^T\,, \quad R = \sigma_v^2$$

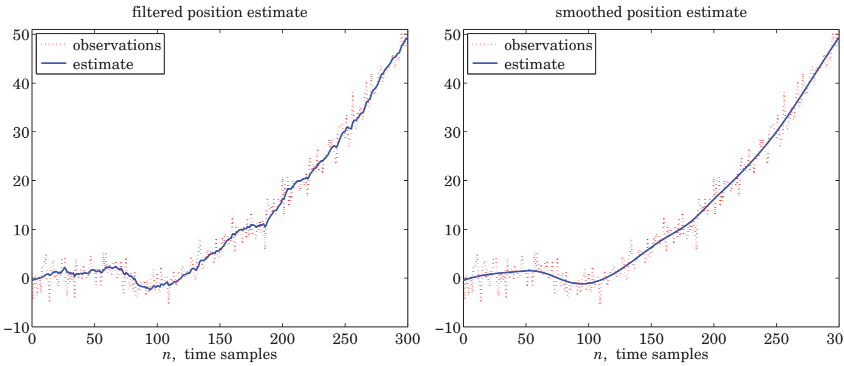
The following figure shows the likelihood as a function of iteration number for the two cases of estimating only Q_w, R (left graph), or, $Q_w, R, \bar{\mathbf{x}}_0, \Sigma_0$ (right graph), using 300 iterations. The data y_n are generated by exactly the same code (not repeated here) as in Example 13.12.2. We note again the rapid initial decrease, followed by a plateau.



The MATLAB code used to generate the right graph is shown below.

```
K = 300; N = length(y)-1; t =0:N;         % data y generated as in Ex. 15.12.2

B = [T^2/2; T]; Binv = pinv(B);

Qw = 0.1; R = 1;                          % initial values for the iteration
x0 = [0; 0.1]; S0 = 0.1 * eye(2);


for i=1:K,
   Q = B*Qw*B';                                   % construct Q_old
   [L(i),Xs,Ps,V] = ksmooth(A,C,Q,R,y,x0,S0);

   Uxx = (Xs*Xs' + sum(Ps,3))/(N+1);              % compute U,V matrices
   Uyx = (y*Xs')/(N+1);
   Uyy = (y*y')/(N+1);
   Vxx = (Xs(:,1:end-1)*Xs(:,1:end-1)' + sum(Ps(:,:,1:end-1),3))/N;
   V1x = (Xs(:,2:end)*Xs(:,1:end-1)' + sum(V(:,:,1:end-1),3))/N;
   V11 = (Xs(:,2:end)*Xs(:,2:end)' + sum(Ps(:,:,2:end),3))/N;

   R = Uyy - C*Uyx' - Uyx*C' + C*Uxx*C';          % construct R_new
   Q = V11 - A*V1x' - V1x*A' + A*Vxx*A';          % construct Q_new
   Qw = Binv * Q * Binv';                         % construct Q_w_new

   x0 = Xs(:,1);             % comment out to skip estimation
   S0 = Ps(:,:,1);
end

k = 0:K-1; figure; plot(k,L);

[Lmin,X,P,Xf,Pf] = kfilt(A,C,Q,R,y,x0,S0);       % use estimated Q,R,x0,S0
[Lmin,Xs,Ps] = ksmooth(A,C,Q,R,y,x0,S0);

figure; plot(t,y,':', t,Xf(1,:),'-');
figure; plot(t,y,':', t,Xs(1,:),'-');
```

Once the parameters $Q_w, R, \bar{\mathbf{x}}_0, \Sigma_0$ have been estimated, the Kalman filter and smoother are run to determine the filtered and smoothed state estimates, $\hat{\mathbf{x}}_{n/n}$ and $\hat{\mathbf{x}}_{n/N}$. Their position components are shown in the figure below.


filtered position estimate

smoothed position estimate

The estimated parameters in Example 13.12.2 were $Q_w = (0.0199)^2$ and $R = (2.0237)^2$, compared with the theoretical values that were simulated $Q_w = (0.02)^2$ and $R = (2.0)^2$. By comparison, the present method gives the estimates: $Q_w = (0.0201)^2$ and $R = (2.0235)^2$ if only Q_w, R are estimated, and $Q_w = (0.0193)^2$ and $R = (2.0221)^2$, if all four parameters $Q_w, R, \bar{\mathbf{x}}_0, \Sigma_0$ are estimated. In the latter case, the estimated initial values were:

$$\bar{\mathbf{x}}_0 = \begin{bmatrix} -0.4468 \\ 0.0791 \end{bmatrix}, \quad \Sigma_0 = \begin{bmatrix} 0.00199 & -0.00015 \\ -0.00015 & 0.00003 \end{bmatrix}$$

Connection to the EM Algorithm

Consider a realization of the states and observations of the state-space model (13.13.1),

$$X = [\mathbf{x}_0, \mathbf{x}_1, \dots, \mathbf{x}_N], \quad Y = [\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_N] \tag{13.13.15}$$

Ideally, the maximum likelihood method would maximize the joint probability density $p_\theta(X,Y)$, or equivalently the log density, $\ln p_\theta(X,Y)$, with respect to the model parameters θ. However, in practice only the observations Y are available and not the states X, and one chooses to maximize instead the marginal density $p_\theta(Y)$:

$$p_\theta(Y) = \int p_\theta(X,Y)\,dX \tag{13.13.16}$$

The EM algorithm for maximizing (13.13.16) is based on the following inequality:

$$\int p(X|Y,\theta^{\text{old}})\,\ln\left[\frac{p_\theta(X,Y)}{p(X|Y,\theta^{\text{old}})}\right]dX \le \ln p_\theta(Y) \tag{13.13.17}$$

where $p(X|Y,\theta^{\text{old}})$ is the conditional density of X given Y and the parameter set θ^{old}. Eq. (13.13.17) is a consequence of Jensen's inequality, which implies:

$$\begin{aligned}\int p(X|Y,\theta^{\text{old}})\,\ln\left[\frac{p_\theta(X,Y)}{p(X|Y,\theta^{\text{old}})}\right]dX &\le \ln\left[\int p(X|Y,\theta^{\text{old}})\,\frac{p_\theta(X,Y)}{p(X|Y,\theta^{\text{old}})}\,dX\right]\\ &= \ln\left[\int p_\theta(X,Y)\,dX\right] = \ln p_\theta(Y)\end{aligned}$$

It follows from Bayes's rule, $p_\theta(X,Y) = p_\theta(X|Y)p_\theta(Y)$, that (13.13.17) becomes an equality at $\theta = \theta^{\text{old}}$. Eq. (13.13.17) can now be re-written in the form:

$$\mathcal{Q}(\theta,\theta^{\text{old}}) + I(\theta^{\text{old}}) \le \ln p_\theta(Y) \tag{13.13.18}$$

where we defined the following conditional expectation and entropy with respect to the conditional density $p(X|Y,\theta^{\text{old}})$:

$$\begin{aligned}\mathcal{Q}(\theta,\theta^{\text{old}}) &= \int p(X|Y,\theta^{\text{old}})\,\ln p_\theta(X,Y)\,dX \equiv E\big[\ln p_\theta(X,Y)\,|\,Y,\theta^{\text{old}}\big]\\ I(\theta^{\text{old}}) &= -\int p(X|Y,\theta^{\text{old}})\,\ln p(X|Y,\theta^{\text{old}})\,dX\end{aligned} \tag{13.13.19}$$

We note that $I(\theta^{\text{old}})$ is a positive quantity. The EM algorithm chooses that θ that maximizes the left bound in (13.13.17). More precisely, the EM algorithm consists of the repetition the the following two steps:

a. The expectation step (E-step), in which the conditional expectation $\mathcal{Q}(\theta,\theta^{\text{old}})$ is calculated.
b. The maximization step (M-step), in which $\mathcal{Q}(\theta,\theta^{\text{old}})$ is maximized resulting into a new θ:

$$\theta^{\text{new}} = \arg\max_{\theta} \mathcal{Q}(\theta,\theta^{\text{old}})$$

The convergence properties of the EM algorithm have been discussed in the literature [905,906], and its application to state-space models, in [909–911] and others.

The advantage of the EM algorithm is that, although $p_\theta(Y)$ is hard to calculate and maximize, the joint density $p_\theta(X,Y)$ is straightforward to calculate before the E-step is carried out. Indeed, successively applying Bayes's rule we have:

$$p_\theta(X,Y) = p(\mathbf{x}_0)\prod_{n=0}^{N-1} p(\mathbf{x}_{n+1}|\mathbf{x}_n)\prod_{n=0}^{N} p(\mathbf{y}_n|\mathbf{x}_n)$$

Assuming Gaussian statistics and full-rank covariance matrices Q, R, Σ_0, we find for the negative-log-likelihood function up to a constant:

$$\begin{aligned}
-\ln p_\theta(X,Y) &= \frac{1}{2}\Big[\ln\det\Sigma_0 + (\mathbf{x}_0-\bar{\mathbf{x}}_0)^T\Sigma_0^{-1}(\mathbf{x}_0-\bar{\mathbf{x}}_0)\Big]\\
&+ \frac{1}{2}\sum_{n=0}^{N-1}\Big[\ln\det Q + (\mathbf{x}_{n+1}-A\mathbf{x}_n)^T Q^{-1}(\mathbf{x}_{n+1}-A\mathbf{x}_n)\Big]\\
&+ \frac{1}{2}\sum_{n=0}^{N}\Big[\ln\det R + (\mathbf{y}_n-C\mathbf{x}_n)^T R^{-1}(\mathbf{y}_n-C\mathbf{x}_n)\Big]
\end{aligned}\tag{13.13.20}$$

and using the identity, $\ln\det Q = \operatorname{tr}\ln Q$, we obtain:

$$\begin{aligned}
-\ln p_\theta(X,Y) &= \frac{1}{2}\operatorname{tr}\Big[\ln\Sigma_0 + \Sigma_0^{-1}(\mathbf{x}_0-\bar{\mathbf{x}}_0)(\mathbf{x}_0-\bar{\mathbf{x}}_0)^T\Big]\\
&+ \frac{1}{2}\operatorname{tr}\Big[\ln Q + Q^{-1}\frac{1}{N}\sum_{n=0}^{N-1}(\mathbf{x}_{n+1}-A\mathbf{x}_n)(\mathbf{x}_{n+1}-A\mathbf{x}_n)^T\Big]\\
&+ \frac{1}{2}\operatorname{tr}\Big[\ln R + R^{-1}\frac{1}{N+1}\sum_{n=0}^{N}(\mathbf{y}_n-C\mathbf{x}_n)(\mathbf{y}_n-C\mathbf{x}_n)^T\Big]
\end{aligned}\tag{13.13.21}$$

Taking conditional expectations with respect to $p(X|Y,\theta^{\text{old}})$, we have:

$$\begin{aligned}
-\mathcal{Q}(\theta,\theta^{\text{old}}) &= \frac{1}{2}\operatorname{tr}\Big[\ln\Sigma_0 + \Sigma_0^{-1}E\big[(\mathbf{x}_0-\bar{\mathbf{x}}_0)(\mathbf{x}_0-\bar{\mathbf{x}}_0)^T\,|\,Y,\theta^{\text{old}}\big]\Big]\\
&+ \frac{1}{2}\operatorname{tr}\Big[\ln Q + Q^{-1}\frac{1}{N}\sum_{n=0}^{N-1}E\big[(\mathbf{x}_{n+1}-A\mathbf{x}_n)(\mathbf{x}_{n+1}-A\mathbf{x}_n)^T\,|\,Y,\theta^{\text{old}}\big]\Big]\\
&+ \frac{1}{2}\operatorname{tr}\Big[\ln R + R^{-1}\frac{1}{N+1}\sum_{n=0}^{N}E\big[(\mathbf{y}_n-C\mathbf{x}_n)(\mathbf{y}_n-C\mathbf{x}_n)^T\,|\,Y,\theta^{\text{old}}\big]\Big]
\end{aligned}$$

which can be written in terms of the definitions (13.13.8),

$$\begin{aligned}
-\mathcal{Q}(\theta,\theta^{\text{old}}) &= \frac{1}{2}\operatorname{tr}\Big[\ln\Sigma_0 + \Sigma_0^{-1}\big[(\bar{\mathbf{x}}_0-\hat{\mathbf{x}}_{0/N})(\bar{\mathbf{x}}_0-\hat{\mathbf{x}}_{0/N})^T + P_{0/N}\big]\Big]\\
&+ \frac{1}{2}\operatorname{tr}\Big[\ln Q + Q^{-1}\big[\hat{V}_{x_1x_1} - A\hat{V}_{xx_1} - \hat{V}_{x_1x}A^T + A\hat{V}_{xx}A^T\big]\Big]\\
&+ \frac{1}{2}\operatorname{tr}\Big[\ln R + R^{-1}\big[\hat{U}_{yy} - C\hat{U}_{xy} - \hat{U}_{yx}C^T + C\hat{U}_{xx}C^T\big]\Big]
\end{aligned}\tag{13.13.22}$$

This represents the E-step of the algorithm. The M-step leads to the same equations as (13.13.9)–(13.13.11). Indeed, the minimization of $-\mathcal{Q}(\theta,\theta^{\text{old}})$ with respect to $\bar{\mathbf{x}}_0$ gives $\bar{\mathbf{x}}_0 = \hat{\mathbf{x}}_{0/N}$. Similarly, the first-order differentials with respect to A and C are:

$$\begin{aligned}
-d\mathcal{Q} &= \frac{1}{2}\operatorname{tr}\Big[Q^{-1}\big[(A\hat{V}_{xx}-\hat{V}_{x_1x})dA^T + dA(\hat{V}_{xx}A^T-\hat{V}_{xx_1})\big]\Big]\\
-d\mathcal{Q} &= \frac{1}{2}\operatorname{tr}\Big[R^{-1}\big[(C\hat{U}_{xx}-\hat{U}_{yx})dC^T + dC(\hat{U}_{xx}-\hat{U}_{xy})\big]\Big]
\end{aligned}$$

Since the variations dA, dC are arbitrary, the vanishing of their coefficients leads to the solutions given in Eq. (13.13.9). The minimizations of $-\mathcal{Q}(\theta,\theta^{\text{old}})$ with respect to Q, R, Σ_0 are similar and have the generic form of finding R that minimizes:

$$F = \operatorname{tr}\big[\ln R + R^{-1}U\big]$$

If all entries of R are treated as independent variables, then the vanishing of the first-order differential of F gives:

$$dF = \operatorname{tr}\big[R^{-1}dR - R^{-1}dR\,R^{-1}U\big] = 0 \quad\Rightarrow\quad R = U$$

where we used $d(R^{-1}) = -R^{-1}dR\,R^{-1}$. In our case, Q, R, Σ_0 are symmetric matrices and only their lower (or upper) triangular parts may be regarded as independent variables. However, the answer turns out to be the same as if all matrix elements were treated as independent. Thus, the minimization of $-\mathcal{Q}(\theta,\theta^{\text{old}})$ gives the same answers as those of Eqs. (13.13.9)–(13.13.11).