Chapter 6 The Recursive Linear Inverse Problem

6.1 The Linear Inverse Problem

In the first three chapters of the course, we treated the linear inverse problem from the point of view of linear algebra. We can recover the techniques studied there in a statistical framework if we consider a model which is **linear** in the parameters. If the **noise** is assumed to be Gaussian distributed and if the **prior probability** of the parameters is (or may be assumed to be) Gaussian distributed, then posterior probability distribution of the parameters is also Gaussian distributed. If the covariance matrix of a Gaussian distribution is strictly positive definite, there can be only one peak and the mean is equal to the mode.

In the situation of non-linear fitting, it is unusual to have more than a few parameters due to the difficulty of characterizing a surface which has a complicated topography in a large space. The problems of multiple maxima and the possibility of finding unrepresentative peaks in the posterior probability can be very awkward. These problems are not present for linear fitting however, as the merit surface is a quadratic form whose matrix completely characterizes the way in which the probability density behaves away from the peak. It is thus quite usual to make models with large numbers of linear parameters.

6.1.1 The Likelihood Function

The linear inverse problem is characterized by the equation

$$\mathbf{y} = \mathbf{C}\mathbf{x} + \mathbf{n} \tag{6.1}$$

where $\mathbf{x} \in \mathbb{R}^M$, $\mathbf{y} \in \mathbb{R}^N$, $\mathbf{C} \in \mathbb{R}^{N \times M}$ and $\mathbf{n}$ is from an N dimensional Gaussian distributed random variable with mean $\mathbf{0}$ and covariance matrix Γ. The likelihood function is

$$p(\mathbf{y}|\mathbf{x}) = p(\mathbf{n} = \mathbf{y} - \mathbf{C}\mathbf{x}) = \frac{1}{(2\pi)^{N/2}\sqrt{\det \Gamma}} \exp\left[-\frac{1}{2}(\mathbf{y} - \mathbf{C}\mathbf{x})^t \Gamma^{-1} (\mathbf{y} - \mathbf{C}\mathbf{x})\right], \tag{6.2}$$

and so the misfit function is

$$\mathcal{E}(\mathbf{x};\mathbf{y}) = (\mathbf{y} - \mathbf{C}\mathbf{x})^t \Gamma^{-1} (\mathbf{y} - \mathbf{C}\mathbf{x}), \tag{6.3}$$

where the dependence on quantities other that $\mathbf{y}$ and $\mathbf{x}$ has been discarded as an unimportant constant. In the special case of independent identically-distributed noise samples, $\Gamma = R^2\mathbf{I}$ where R^2 is the variance of the noise. We prefer not to use the symbol σ^2 for the variance because σ will later be used to refer to the singular values of $\mathbf{C}$. In this case

$$\mathcal{E}(\mathbf{x};\mathbf{y}) = \frac{1}{R^2}\sum_{l=1}^{n}(y_l - \hat{y}_l)^2 \text{ where } \hat{y}_l = \sum_{k=1}^{m} a_{lk}x_k \tag{6.4}$$

$$= \frac{(\mathbf{y} - \mathbf{C}\mathbf{x})^t (\mathbf{y} - \mathbf{C}\mathbf{x})}{R^2}. \tag{6.5}$$

Recall that in the first part of this course, the problem of minimizing $\mathcal{E}$ was discussed in detail. This is most easily done in terms of the **singular value decomposition** of $\mathbf{C}$. If we write

$$\mathbf{C} = \mathbf{U}\mathbf{S}\mathbf{V}^t \tag{6.6}$$

where $\mathbf{U}$ is a unitary $N \times N$ matrix, $\mathbf{V}$ is a unitary $M \times M$ matrix and $\mathbf{S} \in \mathbb{R}^{N \times M}$ is a diagonal matrix of singular values $\mathbf{S} = \text{diag}(\sigma_1, \sigma_2, ..., \sigma_r)$, and where $r \leq \min(M, N)$ is the rank of $\mathbf{C}$, we know that in order to minimize $\mathcal{E}$, we set

$$\tilde{x}_k = \begin{cases} \tilde{y}_k/\sigma_k & \text{for } k = 1, ..., r \\ \text{arbitrary} & \text{for } k = r+1, ..., n \end{cases} \tag{6.7}$$

where $\tilde{\mathbf{y}} = \mathbf{U}^t\mathbf{y}$ and $\tilde{\mathbf{x}} = \mathbf{V}^t\mathbf{x}$ so that $\tilde{x}_k = \mathbf{v}_k^t\mathbf{x}$ and $\tilde{y}_k = \mathbf{u}_k^t\mathbf{y}$ are the projections of $\mathbf{x}$ and $\mathbf{y}$ along the appropriate singular vectors. The minimum value of $\mathcal{E}$ is

$$\mathcal{E}_{\min} = \frac{1}{R^2}\sum_{k=r+1}^{n}\tilde{y}_k^2. \tag{6.8}$$

The singular value decomposition thus explicitly gives **all** the solutions of the least-squares problem.

Exercise: If the above results seem unfamiliar, satisfy yourself that

$$\mathcal{E}(\mathbf{x};\mathbf{y}) = \frac{1}{R^2}\left[\sum_{k=1}^{r}\sigma_k^2\left(\tilde{x}_k - \frac{\tilde{y}_k}{\sigma_k}\right)^2 + \sum_{k=r+1}^{n}\tilde{y}_k^2\right]. \tag{6.9}$$

Exercise: Show that if the covariance of the noise is Γ rather than a multiple of the identity, we can carry out an analogue of the above analysis by using the singular value decomposition of $\Gamma^{-1/2}\mathbf{C}$, rather than of $\mathbf{C}$. Since Γ is a positive definite matrix, $\Gamma^{-1/2}$ always exists. The role of $\Gamma^{-1/2}$ is to effectively **decorrelate** the noise, and in signal processing is often called a **prewhitening filter**.

6.1.2 The role of the prior

As we saw in the first part of the course, the presence of zero or small singular values causes problems since the corresponding components $\tilde{x}_k$ are arbitrary or badly determined by the data. The solution introduced there was **regularization**, which we can now understand in terms of including a non-flat **prior probability.** If we choose a multivariate Gaussian prior with a non-zero mean

$$p(\mathbf{x}) \propto \exp\left[-\frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}})^t\mathbf{P}^{-1}(\mathbf{x}-\bar{\mathbf{x}})\right], \tag{6.10}$$

this has a peak at $\mathbf{x} = \bar{\mathbf{x}}$ and the covariance of the prior density is given by the (symmetric) matrix $\mathbf{P}$. The corresponding preference function $\mathcal{S}(\mathbf{x})$ is

$$\mathcal{S}(\mathbf{x}) = -(\mathbf{x}-\bar{\mathbf{x}})^t\mathbf{P}^{-1}(\mathbf{x}-\bar{\mathbf{x}}) \tag{6.11}$$

and the figure of merit function becomes

$$\mathcal{E}(\mathbf{x};\mathbf{y}) - \mathcal{S}(\mathbf{x}) = (\mathbf{y}-\mathbf{C}\mathbf{x})^t\Gamma^{-1}(\mathbf{y}-\mathbf{C}\mathbf{x}) + (\mathbf{x}-\bar{\mathbf{x}})^t\mathbf{P}^{-1}(\mathbf{x}-\bar{\mathbf{x}}), \tag{6.12}$$

which reduces to

$$\mathcal{E}(\mathbf{x};\mathbf{y}) - \mathcal{S}(\mathbf{x}) = \frac{1}{R^2}(\mathbf{y}-\mathbf{C}\mathbf{x})^t(\mathbf{y}-\mathbf{C}\mathbf{x}) + (\mathbf{x}-\bar{\mathbf{x}})^t\mathbf{P}^{-1}(\mathbf{x}-\bar{\mathbf{x}}), \tag{6.13}$$

if $\Gamma = R^2\mathbf{I}$. Comparing this with the procedure for Tikhonov regularization, in which we find the solution of

$$\hat{\mathbf{x}} = \arg\min\left\{\|\mathbf{y}-\mathbf{C}\mathbf{x}\|^2 + \lambda^2\|\mathbf{L}(\mathbf{x}-\mathbf{x}_\infty)\|^2\right\} \tag{6.14}$$

it is clear that these are equivalent when $\bar{\mathbf{x}} = \mathbf{x}_\infty$ the regularization parameter λ^2 is chosen to be $1/R^2$ and we set $\mathbf{P}^{-1} = \mathbf{L}\mathbf{L}^t$. Thus we see that

Tikhonov regularization $\equiv$ MAP estimator with linear model, Gaussian noise and Gaussian prior

If $\mathbf{L}$ is not simply the identity, the formal solution to the Tikhonov regularization problem is in terms of the **generalized singular value decomposition**. Note that we often allow $\mathbf{P}^{-1}$ to have some zero eigenvalues, although this would lead to an improper (un-normalizable) prior probability "density". This does not lead to any problems so long that the posterior probability is proper.

The posterior probability density is given by

$$\begin{aligned} p(\mathbf{x}|\mathbf{y}) &\propto \exp\left\{-\frac{1}{2}\left[\mathcal{E}(\mathbf{x};\mathbf{y}) - \mathcal{S}(\mathbf{x})\right]\right\} \\ &= \exp\left\{-\frac{1}{2}(\mathbf{y}-\mathbf{C}\mathbf{x})^t \Gamma^{-1}(\mathbf{y}-\mathbf{C}\mathbf{x}) + \frac{1}{2}(\mathbf{x}-\bar{\mathbf{x}})^t \mathbf{P}^{-1}(\mathbf{x}-\bar{\mathbf{x}})\right\} \end{aligned}$$

Since this is a Gaussian density in $\mathbf{x}$, we try to write the exponent explicitly as a quadratic form in $\mathbf{x}$. We see that

$$\begin{aligned} \mathcal{E}(\mathbf{x};\mathbf{y}) - \mathcal{S}(\mathbf{x}) &= (\mathbf{y}-\mathbf{C}\mathbf{x})^t \Gamma^{-1}(\mathbf{y}-\mathbf{C}\mathbf{x}) + (\mathbf{x}-\bar{\mathbf{x}})^t \mathbf{P}^{-1}(\mathbf{x}-\bar{\mathbf{x}}) \\ &= \mathbf{x}^t\left(\mathbf{C}^t\Gamma^{-1}\mathbf{C} + \mathbf{P}^{-1}\right)\mathbf{x} - \left(\mathbf{y}^t\Gamma^{-1}\mathbf{C} + \bar{\mathbf{x}}^t\mathbf{P}^{-1}\right)\mathbf{x} \qquad (6.15) \\ &\quad -\mathbf{x}^t\left(\mathbf{C}^t\Gamma^{-1}\mathbf{y} + \mathbf{P}^{-1}\bar{\mathbf{x}}\right) + \left(\mathbf{y}^t\Gamma^{-1}\mathbf{y} + \bar{\mathbf{x}}^t\mathbf{P}^{-1}\bar{\mathbf{x}}\right) \qquad (6.16) \end{aligned}$$

We may **complete the square** by using the identity

$$\mathbf{x}^t\mathbf{A}\mathbf{x} - \mathbf{B}^t\mathbf{x} - \mathbf{x}^t\mathbf{B} + C = \left(\mathbf{x}-\mathbf{A}^{-1}\mathbf{B}\right)^t \mathbf{A}\left(\mathbf{x}-\mathbf{A}^{-1}\mathbf{B}\right) + \left(C - \mathbf{B}^t\mathbf{A}^{-1}\mathbf{B}\right) \qquad (6.17)$$

where $\mathbf{A}$ is a symmetric non-singular $N \times N$ matrix, $\mathbf{B}$ is an $N \times 1$ vector and C is a scalar. This may readily be verified by expanding the right hand side. Thus the merit function may be rewritten as

$$\mathcal{E}(\mathbf{x};\mathbf{y}) - \mathcal{S}(\mathbf{x}) = \left[(\mathbf{x}-\tilde{\mathbf{x}})^t\left[\mathbf{C}^t\Gamma^{-1}\mathbf{C} + \mathbf{P}^{-1}\right](\mathbf{x}-\tilde{\mathbf{x}}) + \left(\mathbf{y}^t\Gamma^{-1}\mathbf{y} + \bar{\mathbf{x}}^t\mathbf{P}\bar{\mathbf{x}} - \tilde{\mathbf{x}}^t\left[\mathbf{C}^t\Gamma^{-1}\mathbf{C} + \mathbf{P}^{-1}\right]^{-1}\tilde{\mathbf{x}}\right)\right] \qquad (6.18)$$

where

$$\tilde{\mathbf{x}} = \left[\mathbf{C}^t\Gamma^{-1}\mathbf{C} + \mathbf{P}^{-1}\right]^{-1}\left(\mathbf{C}^t\Gamma^{-1}\mathbf{y} + \mathbf{P}^{-1}\bar{\mathbf{x}}\right).$$

This has a unique solution (i.e., the posterior probability is normalizable) if $\mathbf{C}^t\Gamma^{-1}\mathbf{C} + \mathbf{P}^{-1}$ is invertible, i.e., if the intersection of the null spaces of $\mathbf{P}^{-1}$ and of $\mathbf{C}$ only contains the zero vector. The system of equations which must be solved to find the maximum *à posteriori* solution $\tilde{\mathbf{x}}$ is

$$\left(\mathbf{C}^t\Gamma^{-1}\mathbf{C} + \mathbf{P}^{-1}\right)\tilde{\mathbf{x}} = \mathbf{C}^t\Gamma^{-1}\mathbf{y} + \mathbf{P}^{-1}\bar{\mathbf{x}} \qquad (6.19)$$

which can be done quite efficiently if $\mathbf{C}^t\Gamma^{-1}\mathbf{C} + \mathbf{P}^{-1}$ is a sparse matrix, or by using one of a variety of iterative methods as discussed above.

The **covariance** of the posterior probability density is

$$\tilde{\mathbf{P}} = \left(\mathbf{P}^{-1} + \mathbf{C}^t\Gamma^{-1}\mathbf{C}\right)^{-1}. \qquad (6.20)$$

We may formally obtain the results for a flat prior (i.e., the maximum likelihood results) by setting $\mathbf{P}^{-1} = 0$ in the above.

6.2 The Recursive Linear Inverse Problem

In the above, we suppose that all the data $\mathbf{y}$ have been collected before we process them to find the extimate of $\mathbf{x}$ using a relation such as (6.19). The solution process involves the solution of a set of N simultaneous linear equations, where N is the number of unknown parameters. Often in practice, we do not have the

luxury of waiting for all the data to be collected before we are required to give an estimate of $\mathbf{x}$, this is the usual situation in navigation, radar tracking and automatic control. The aim of recursive algorithms is to provide a way of getting an estimate of $\mathbf{x}$ which is continually updated as the data arrive. . If there is enough computing power, it might be feasible to recompute and solve the matrix equations every time a new data point arrives but it is of course interesting to know that this is in fact not necessary.

We shall first consider the situation in which the parameter vector remains constant and we collect data which (hopefully) allow us to make successively more accurate estimates of the parameters. An example of this process is fitting the trajectory of an aircraft to a straight line as we get successive radar fixes of its position. At every stage, we need to know the best straight line through the data points collected so far and we assume that the parameters of the line do not change. The solution to this problem is the *recursive least squares* algorithm.

In the early 1960s, Kalman and Bucy made a very major advance when they realized that it is possible to do recursive least squares estimation even when the parameter vector itself is changing between timesteps due to the dynamical evolution of the system. The Kalman filter is now used in a very wide variety of applications – airplanes land on automatic control while a Kalman filter is fitting speed and position to the laws of motion, satellite navigation systems employ Kalman filters to take into account the motions of the satellite and the receiver, etc. The Kalman filter is discussed in the next section.

The key ideas behind the recursive least-squares algorithm are

- the sequential way in which Bayes' theorem may be used to take into account multiple observations,
- a Gaussian prior probability is converted into a Gaussian posterior probability when the likelihood function is Gaussian,
- the use of a small set of sufficient statistics which completely characterize our present state of knowledge of the parameters given the prior and all the data collected so far.

Let us suppose that we wish to make a least-squares estimate of the M component parameter vector $\mathbf{x}$. The k'th datum consists of a vector $\mathbf{y}_k$ related to $\mathbf{x}$ via a linear transformation of the form

$$\mathbf{y}_k = \mathbf{C}_k\mathbf{x} + \mathbf{n}_k \tag{6.21}$$

where $\mathbf{n}_k$ is from a zero-mean Gaussian noise process with covariance matrix Γ_k. Suppose that just before we measured $\mathbf{y}_k$, our state of knowledge of $\mathbf{x}$ is represented by a Gaussian prior probability with mean $\bar{\mathbf{x}}_k$ and covariance $\mathbf{P}_k$. We adopt the following procedure

1. On measuring the data $\mathbf{y}_k$, we use the analogues of equations (6.19) and (6.20) to calculate the mean $\tilde{\mathbf{x}}_k$ and covariance $\tilde{\mathbf{P}}_k$ of the posterior probability. These may be rewritten as

$$\tilde{\mathbf{P}}_k = \left[\mathbf{P}_k^{-1} + \mathbf{C}_k^t\Gamma_k^{-1}\mathbf{C}_k\right]^{-1} \tag{6.22}$$

$$\tilde{\mathbf{x}}_k = \tilde{\mathbf{P}}_k\left(\mathbf{C}_k^t\Gamma_k^{-1}\mathbf{y}_k + \mathbf{P}_k^{-1}\bar{\mathbf{x}}_k\right) \tag{6.23}$$

2. We use the posterior probability distribution for this data vector as the prior probability distribution for the next, thus we set $\bar{\mathbf{x}}_{k+1} = \tilde{\mathbf{x}}_k$ and $\mathbf{P}_{k+1} = \tilde{\mathbf{P}}_k$.
3. Repeat the above steps with the successive data vectors, obtaining a sequence of posterior probability distributions.

It should be apparent that the overall result of these successive modifications to our state of knowledge is the same as if we had processed all the data after they were collected. The advantage of the recursive algorithm is that at each stage we have a "best representation" of what we know about the parameters.

The above equations are not very convenient for computation, as they require us to invert a (large) matrix of size $M \times M$ each time a new data vector is measured. Fortunately, there is a way around this problem.

6.2.1 Block matrix inversion

The following lemma is useful in the subsequent theory. Suppose that $\mathbf{A}$ is an $m \times m$, $\mathbf{B}$ is an $m \times n$, $\mathbf{C}$ is a $n \times m$ and $\mathbf{D}$ is an $n \times n$ matrix. Assuming that the inverses in the following expressions exist, the inverse of the block matrix can be written as

$$\begin{aligned}\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix}^{-1} &= \begin{pmatrix} \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1} & -\mathbf{A}^{-1}\mathbf{B}(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \\ -(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1} & (\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \end{pmatrix} \\ &= \begin{pmatrix} (\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & -(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1}\mathbf{B}\mathbf{D}^{-1} \\ -\mathbf{D}^{-1}\mathbf{C}(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} & \mathbf{D}^{-1} + \mathbf{D}^{-1}\mathbf{C}(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1}\mathbf{B}\mathbf{D}^{-1} \end{pmatrix}\end{aligned}$$

Proof: These results follow from straightforward Gauss elimination using matrices. It is important to keep the order of the factors. For example, consider the proof of the first of the two equalities. The matrix is reduced to the identity via a sequence of elementary row operations which are also applied to the identity matrix. At the end of this sequence the identity is transformed into the desired inverse.

$$\left(\begin{array}{cc|cc} \mathbf{A} & \mathbf{B} & \mathbf{I} & \mathbf{0} \\ \mathbf{C} & \mathbf{D} & \mathbf{0} & \mathbf{I} \end{array}\right)$$

Adding $\mathbf{C}\mathbf{A}^{-1}$ times the first row to the second and then multiplying the first row by $\mathbf{A}^{-1}$ yields

$$\left(\begin{array}{cc|cc} \mathbf{I} & \mathbf{A}^{-1}\mathbf{B} & \mathbf{A}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B} & -\mathbf{C}\mathbf{A}^{-1} & \mathbf{I} \end{array}\right)$$

Multiplying the second row by $(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}$ gives

$$\left(\begin{array}{cc|cc} \mathbf{I} & \mathbf{A}^{-1}\mathbf{B} & \mathbf{A}^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & -(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1} & (\mathbf{D} - \mathbf{C}\mathbf{A}^{-\mathbf{1}}\mathbf{B})^{-\mathbf{1}} \end{array}\right)$$

Subtracting $\mathbf{A}^{-1}\mathbf{B}$ times the second row from the first,

$$\left(\begin{array}{cc|cc} \mathbf{I} & \mathbf{0} & \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1} & -\mathbf{A}^{-1}\mathbf{B}(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1} \\ \mathbf{0} & \mathbf{I} & -(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1} & (\mathbf{D} - \mathbf{C}\mathbf{A}^{-\mathbf{1}}\mathbf{B})^{-\mathbf{1}} \end{array}\right)$$

The matrix on the right hand side is the desired inverse. The other form for the inverse is proved similarly.

Some interesting corollaries of the above result come from looking at the corresponding elements of the inverse matrix. For example,

$$(\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C})^{-1} = \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}(\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B})^{-1}\mathbf{C}\mathbf{A}^{-1} \tag{6.24}$$

We may regard this as an expression for inverting a matrix which is close to a given matrix $\mathbf{A}$. This result is the **Woodbury formula**. As a special case, suppose that $\mathbf{A}$ is an $m \times m$ matrix, $\mathbf{B} = \mathbf{u}$ is a $m \times 1$ column vector, $\mathbf{C} = \mathbf{v}^t$ is a $1 \times m$ row vector and $\mathbf{D}$ is the 1×1 matrix 1. Then

$$\begin{aligned}(\mathbf{A} - \mathbf{u}\mathbf{v}^t)^{-1} &= \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{u}(1 - \mathbf{v}^t\mathbf{A}^{-1}\mathbf{u})^{-1}\mathbf{v}^t\mathbf{A}^{-1} \\ &= \mathbf{A}^{-1} + \frac{(\mathbf{A}^{-1}\mathbf{u})(\mathbf{v}^t\mathbf{A}^{-1})}{1 - \mathbf{v}^t\mathbf{A}^{-1}\mathbf{u}}\end{aligned} \tag{6.25}$$

where the second equality arises since $\mathbf{v}^t\mathbf{A}^{-1}\mathbf{u}$ is a scalar. This is known as the **Sherman-Morrison** formula.

The utility of the Sherman-Morrison formula is apparent if m is large so that inverting $\mathbf{A}$ is computationally expensive. Suppose that we have previously calculated $\mathbf{A}^{-1}$ and now want to make a "small" change in $\mathbf{A}$ such as changing one element a_{ij}, or a few elements, or one row, or one column. Using the Sherman-Morrison formula it is possible to calculate the corresponding change in $\mathbf{A}^{-1}$ with far less computational effort than reinverting the new matrix provided that the change is of the form $\mathbf{u}\mathbf{v}^t$. Instead of $O(m^3/3)$ multiplications to repeat the inversion, the Sherman-Morrisson formula only takes about $3m^2$ multiplications (two matrix-vector multiplications and a vector dot product).

Similar comments apply to the Woodbury formula given above which is the matrix form of the Sherman-Morrisson formula. This allows more general perturbations to the matrix $\mathbf{A}$. If $n < m$, the Woodbury formula allows the inverse of the new $m \times m$ matrix $\mathbf{A} - \mathbf{B}\mathbf{D}^{-1}\mathbf{C}$ to be calculated via an $n \times n$ matrix inversion of $\mathbf{D} - \mathbf{C}\mathbf{A}^{-1}\mathbf{B}$ together with several matrix multiplications.

We now return to the recursive least-squares problem. The update equation for the covariance may be rewritten using the Woodbury formula with $\mathbf{A} = \mathbf{P}_k^{-1}$, $\mathbf{B} = \mathbf{C}_k^t$, $\mathbf{C} = -\mathbf{C}_k$ and $\mathbf{D} = \Gamma_1$. We find

$$\tilde{\mathbf{P}}_k = \left[\mathbf{P}_k^{-1} + \mathbf{C}_k^t \Gamma_k^{-1} \mathbf{C}_k\right]^{-1} = \mathbf{P}_k - \mathbf{P}_k \mathbf{C}_k^t \left[\Gamma_k + \mathbf{C}_k \mathbf{P}_k \mathbf{C}_k^t\right]^{-1} \mathbf{C}_k \mathbf{P}_k \tag{6.26}$$

This update formula for $\mathbf{P}$ works without requiring the calculation of the inverse of $\mathbf{P}$. The inversion of $\Gamma_k + \mathbf{C}_k \mathbf{P}_k \mathbf{C}_k^t$ involves a matrix of size determined by the data vector rather than the parameter vector. In particular if the data at each step is a scalar, the inversion degenerates into a simple division.

Note that since $\mathbf{P}_k$, $\tilde{\mathbf{P}}_k$ and $\mathbf{P}_k \mathbf{C}_k^t (\Gamma_k + \mathbf{C}_k \mathbf{P}_k \mathbf{C}_k^t)^{-1} \mathbf{C}_k \mathbf{P}_k$ are all positive definite, the effect of including the datum $\mathbf{y}_k$ is to reduce the variances of the parameter estimates (which form the diagonal of $\mathbf{P}$).

The update formula for the mean may be rewritten as

$$\begin{aligned} \tilde{\mathbf{x}}_k &= \tilde{\mathbf{P}}_k \left(\left[\tilde{\mathbf{P}}_k^{-1} - \mathbf{C}_k^t \Gamma_k^{-1} \mathbf{C}_k \right] \overline{\mathbf{x}}_k + \mathbf{C}_k^t \Gamma_k^{-1} \mathbf{y}_k \right) \\ &= \overline{\mathbf{x}}_k + \tilde{\mathbf{P}}_k \mathbf{C}_k^t \Gamma_k^{-1} \left(\mathbf{y}_k - \mathbf{C}_k \overline{\mathbf{x}}_k \right) \\ &= \overline{\mathbf{x}}_k + \mathbf{K}_k \left(\mathbf{y}_k - \mathbf{C}_k \overline{\mathbf{x}}_k \right) \end{aligned} \tag{6.27}$$

where $\mathbf{K}_k = \tilde{\mathbf{P}}_k \mathbf{C}_k^t \Gamma_k^{-1}$ is called the **gain matrix.**

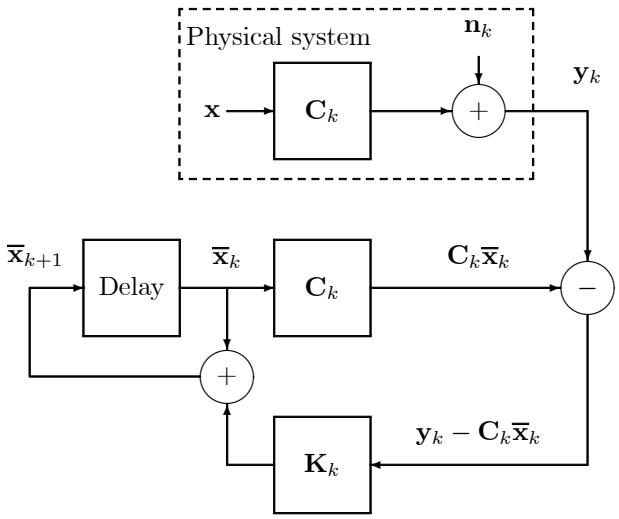
Exercise: Show that the gain matrix $\mathbf{K}_k$ can be rewritten as

$$\mathbf{K}_k = \mathbf{P}_k \mathbf{C}_k^t \left(\Gamma_k + \mathbf{C}_k \mathbf{P}_k \mathbf{C}_k^t \right)^{-1} \tag{6.28}$$

and the covariance update as

$$\tilde{\mathbf{P}}_1 = \left(\mathbf{I} - \mathbf{K}_k \mathbf{C}_k \right) \mathbf{P}_k \tag{6.29}$$

The algorithm can be represented by the block diagram below. This structure suggests interpreting the recursive least squares algorithm as a system with feedback which takes the "error signal" $\mathbf{r}_k = \mathbf{y}_k - \mathbf{C}_k \mathbf{x}_k$ based on the current estimate and processes it using $\mathbf{K}_k$ to produce an update which is added to $\mathbf{x}_k$ to give $\tilde{\mathbf{x}}_k$.



Update equations

$$\mathbf{K}_k = \mathbf{P}_k \mathbf{C}_k^t (\Gamma_k + \mathbf{C}_k \mathbf{P}_k \mathbf{C}_k^t)^{-1}$$

$$\mathbf{P}_{k+1} = (\mathbf{I} - \mathbf{K}_k \mathbf{C}_k) \mathbf{P}_k$$

or

$$\mathbf{P}_{k+1} = \mathbf{P}_k - \mathbf{P}_k \mathbf{C}_k^t (\Gamma_k + \mathbf{C}_k \mathbf{P}_k \mathbf{C}_k^t)^{-1} \mathbf{C}_k \mathbf{P}_k$$

$$\mathbf{K}_k = \mathbf{P}_{k+1} \mathbf{C}_k^t \Gamma_k$$

Example: As the simplest example of the recursive least squares algorithm, suppose that the data are $y_k = x + n_k$ so that the scalar-valued samples y_k are simply noise-corrupted versions of the scalar-valued

parameter x. If the variance in each noise sample is σ^2, we see that $C_k = 1$ and $\Gamma_k = \sigma^2$. We thus find

$$K_k = \frac{P_k}{\sigma^2 + P_k} \tag{6.30}$$

$$P_{k+1} = \tilde{P}_k = \frac{\sigma^2 P_k}{\sigma^2 + P_k} \tag{6.31}$$

$$\overline{x}_{k+1} = \tilde{x}_k = \overline{x}_k + \frac{P_k}{\sigma^2 + P_k}(y_k - \overline{x}_k) \tag{6.32}$$

If we start with $P_1 = \infty$, we find that $K_1 = 1$, $P_2 = \sigma^2$ and $\overline{x}_2 = y_1$. Subsequently,

$$K_2 = \frac{1}{2}, \quad P_3 = \frac{\sigma^2}{2}, \quad \overline{x}_3 = \overline{x}_2 + \frac{1}{2}(y_2 - \overline{x}_2) = \frac{1}{2}(y_1 + y_2)$$

and

$$K_3 = \frac{1}{3}, \quad P_4 = \frac{\sigma^2}{3}, \quad \overline{x}_4 = \overline{x}_3 + \frac{1}{3}(y_3 - \overline{x}_3) = \frac{1}{3}(y_1 + y_2 + y_3)$$

we see that $\overline{x}_{n+1} = \tilde{x}_n$ is just the mean of the n data points $y_1, y_2, ..., y_n$. The recursive least-squares algorithm allows us to compute this as the data are collected.

6.3 The Kalman Filter

The Kalman filter applies to the situation when the parameters of the system are evolving between samples and we wish to calculate the best estimate of this dynamically changing state. By including knowledge of the dynamics of the system into the observation process, we can improve the performance of the estimator. Since the state is now evolving, instead of a single "true" parameter vector $\mathbf{x}$, the parameter vector is $\mathbf{x}_k$. In the basic discrete-time Kalman filter which we shall consider, the inter-sample time evolution is modelled by an inhomogeneous linear transformation of the form

$$\mathbf{x}_{k+1} = \Phi_k \mathbf{x}_k + \mathbf{b}_k \tag{6.33}$$

where Φ_k is the time-evolution matrix between samples and $\mathbf{b}_k$ is a known vector. At each sample instant, we measure a data vector

$$\mathbf{y}_k = \mathbf{C}_k \mathbf{x}_k + \mathbf{n}_k \tag{6.34}$$

which is corrupted by noise $\mathbf{n}_k$ with covariance matrix Γ_k.

Before measuring any data, we start with a Gaussian prior probability density for $\mathbf{x}_1$, the parameter vector at the first sample time. We shall denote the mean and covariance of this prior probability by $\overline{\mathbf{x}}_1$ and $\mathbf{P}_1$. As a result of measuring $\mathbf{y}_1$, we update our state of knowledge of $\mathbf{x}_1$. The posterior probability $p(\mathbf{x}_1|\mathbf{y}_1)$ is also Gaussian and we shall denote its mean and covariance by $\tilde{\mathbf{x}}_1$ and $\tilde{\mathbf{P}}_1$. Since the dynamical evolution of the system is known, at this stage we can compute a (predicted) prior probability density for $\mathbf{x}_2$ based on the data so far. We denote the mean and covariance of this prediction by $\overline{\mathbf{x}}_2$ and $\mathbf{P}_2$. Continuing in this way, at each sample time we obtain a posterior probability for the state at that time as well as a prediction for the state at the next time.

Example: Consider a body falling under gravity. The equation of motion for the height h is

$$\ddot{h} = -g \tag{6.35}$$

We can rewrite this second-order differential equation in terms of a first-order vector differential equation in the usual way. This yields

$$\frac{d}{dt}\begin{pmatrix} h \\ v \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} h \\ v \end{pmatrix} + \begin{pmatrix} 0 \\ -g \end{pmatrix} \tag{6.36}$$

Suppose that we sample this system at regularly spaced times T apart. The differential equation must be converted into discrete form, expressing h and v at time $(k+1)T$ in terms of their values at time kT. In this example, since the differential equation is shift invariant, Laplace transform techniques can be used. The result is

$$v[(k+1)T] = v[kT] - gT \tag{6.37}$$

$$h[(k+1)T] = h[kT] + v[kT]T - \frac{1}{2}gT^2 \tag{6.38}$$

This can be written in matrix form as

$$\begin{pmatrix} h_{k+1} \\ v_{k+1} \end{pmatrix} = \begin{pmatrix} 1 & T \\ 0 & 1 \end{pmatrix} \begin{pmatrix} h_k \\ v_k \end{pmatrix} + \begin{pmatrix} -\frac{1}{2}gT^2 \\ -gT \end{pmatrix} \tag{6.39}$$

We see that a possible state vector for this problem is $\mathbf{x} = \begin{pmatrix} h \\ v \end{pmatrix}$. Comparing equations (6.39) and (6.33), we readily identify the transition matrix Φ and inhomogeneous term $\mathbf{b}$ for this problem.

If at each sample time we observe a noisy version of h, the observation matrix $\mathbf{C}$ is $\begin{pmatrix} 1 & 0 \end{pmatrix}$. Notice that in this example, the matrices Φ and $\mathbf{C}$ happen to be constant.

The actual dynamical evolution of the real system is usually only approximated by the time-evolution equations that we formulate. There may be additional forces which we do not take into account within our model (e.g. air friction in the case of the falling body) and our model may not be totally accurate. If we use our approximate equations of motion to predict the subsequent state of the system based on the current state, we may become incorrectly overconfident of the accuracy of the estimate. In order to accommodate the inadequacies in the model and the presence of unknown or stochastically variable forces, the time evolution equation (6.33) is modified to

$$\mathbf{x}_{k+1} = \Phi_k \mathbf{x}_k + \mathbf{b}_k + \mathbf{e}_k \tag{6.40}$$

where $\mathbf{e}_k$ is taken to be from a zero-mean Gaussian process with covariance matrix $\mathbf{S}_k$. By assigning a large covariance to $\mathbf{e}_k$, we are admitting that our dynamical model is likely to be inadequate. Of course if the covariance is made too large, the information about the dynamics is of little benefit in the estimation process.

Let us analyze the step at which the first datum $\mathbf{y}_1$ is measured. The modification of prior probability for $\mathbf{x}_1$ to its posterior probability is exactly the same as before and the relationships (6.26) through (6.27) still apply. Equation (6.27) gives our MAP estimate of $\mathbf{x}_1$ after measuring $\mathbf{y}_1$ and is called the *Kalman filtered* state vector.

We now consider the evolution from $\mathbf{x}_1$ to $\mathbf{x}_2$. Since

$$\mathbf{x}_2 = \Phi_1 \mathbf{x}_1 + \mathbf{b}_1 + \mathbf{e}_1 \tag{6.41}$$

where $\mathbf{e}_1$ is a zero-mean Gaussian distributed process, a Gaussian probability density for $\mathbf{x}_1$ leads to a Gaussian probability density for $\mathbf{x}_2$. The mean is

$$\mathrm{E}[\mathbf{x}_2] = \Phi_1 \mathrm{E}[x_1] + \mathbf{b}_1 \tag{6.42}$$

and the covariance is

$$\begin{aligned} \mathrm{E}[(\mathbf{x}_2 - \mathrm{E}[\mathbf{x}_2])(\mathbf{x}_2 - \mathrm{E}[\mathbf{x}_2])^t] &= \mathrm{E}[(\Phi_1(\mathbf{x}_1 - \mathrm{E}[\mathbf{x}_1]) + \mathbf{e}_1)(\Phi_1(\mathbf{x}_1 - \mathrm{E}[\mathbf{x}_1]) + \mathbf{e}_1)^t] \\ &= \Phi_1 \mathrm{E}[(\mathbf{x}_1 - \mathrm{E}[\mathbf{x}_1])(\mathbf{x}_1 - \mathrm{E}[\mathbf{x}_1])^t] \Phi_1^t + \mathrm{E}[\mathbf{e}_1 \mathbf{e}_1^t] \end{aligned} \tag{6.43}$$

Notice that these are the mean and covariance for our state of knowledge of $\mathbf{x}_2$ based on our prior, on the measurement $\mathbf{y}_1$ and on the equation of evolution. As such it is a *prediction* of $\mathbf{x}_2$ and gives the *prior* probability for $\mathbf{x}_2$ *before* we measure $\mathbf{y}_2$. We thus denote this mean and covariance of $\mathbf{x}_2$ by $\overline{\mathbf{x}}_2$ and $\mathbf{P}_2$. The mean and covariance of $\mathbf{x}_1$ on the right hand sides are those *after* measuring $\mathbf{y}_1$ and so are $\tilde{\mathbf{x}}_1$ and $\tilde{\mathbf{P}}_1$. Hence,

$$\overline{\mathbf{x}}_2 = \Phi_1 \tilde{\mathbf{x}}_1 + \mathbf{b}_1 \tag{6.44}$$

$$\mathbf{P}_2 = \Phi_1 \tilde{\mathbf{P}}_1^t \Phi_1 + \mathbf{S}_1 \tag{6.45}$$

$\overline{\mathbf{x}}_2$ is termed the *Kalman prediction* of the new state.

Based on these results, we can summarize the action of one step of the Kalman filter algorithm as follows. Just before measuring the datum $\mathbf{y}_k$, our state of knowledge is described by $\overline{\mathbf{x}}_k$ and $\mathbf{P}_k$ the prior mean and covariance for the state vector $\mathbf{x}_k$. As a result of measuring $\mathbf{y}_k$, we compute in sequence:

1. The covariance of the posterior probability of $\mathbf{x}_k$:
$$\tilde{\mathbf{P}}_k = \mathbf{P}_k - \mathbf{P}_k\mathbf{C}_k^t(\Gamma_k + \mathbf{C}_k\mathbf{P}_k\mathbf{C}_k^t)^{-1}\mathbf{C}_k\mathbf{P}_k \tag{6.46}$$
2. The matrix $\mathbf{K}_k$
$$\mathbf{K}_k = \tilde{\mathbf{P}}_k\mathbf{C}_k^t\Gamma_k^{-1} \tag{6.47}$$
This is called the *Kalman gain matrix*
3. The mean of the posterior probability of $\mathbf{x}_k$, i.e., the Kalman filtered state:
$$\tilde{\mathbf{x}}_k = \overline{\mathbf{x}}_k + \mathbf{K}_k(\mathbf{y}_k - \mathbf{C}_k\overline{\mathbf{x}}_k) \tag{6.48}$$
4. The covariance of the prior probability of $\mathbf{x}_{k+1}$:
$$\mathbf{P}_{k+1} = \Phi_k\tilde{\mathbf{P}}_k^t\Phi_k + \mathbf{S}_k \tag{6.49}$$
5. The mean of the prior probability of $\mathbf{x}_{k+1}$, i.e., the Kalman predicted new state
$$\overline{\mathbf{x}}_{k+1} = \Phi_k\tilde{\mathbf{x}}_k + \mathbf{b}_k \tag{6.50}$$

Just as in the recursive least squares estimator, it is possible to reverse the order of computation of the first two steps. These lead to

$$\mathbf{K}_k = \mathbf{P}_k\mathbf{C}_k^t\left(\Gamma_k + \mathbf{C}_k\mathbf{P}_k\mathbf{C}_k^t\right)^{-1} \tag{6.51}$$
$$\tilde{\mathbf{P}}_k = \left(\mathbf{I} - \mathbf{K}_k\mathbf{C}_k\right)\mathbf{P}_k \tag{6.52}$$

The Kalman filter can be represented by the block diagram shown below. We see that the filter is essentially a copy of the model of the physical system which is driven by $\mathbf{K}_k(\mathbf{y}_k - \mathbf{C}_k\overline{\mathbf{x}}_k)$, a filtered version of the error signal. This structure is referred to in control theory as an **observer**. We note that the observer structure was forced upon us as a consequence of the mathematics and was not introduced in the formulation of the problem. An alternative derivation of the Kalman filter can be given by regarding it as the optimal observer of a physical system rather than in terms of estimation theory. In this alternative formulation, the Kalman gain $\mathbf{K}_k$ is chosen to minimize the expected difference between the observer state $\overline{\mathbf{x}}_k$ and the true state $\mathbf{x}_k$.

Example: Let us again consider the problem which we solved using the recursive least squares algorithm. The data are $y_k = x_k + n_k$ so that the scalar-valued samples y_k are simply noise-corrupted versions of the scalar-valued parameter x_k. If the variance in each noise sample is σ^2, we see that $C_k = 1$ and $\Gamma_k = \sigma^2$. For the dynamical evolution of the system we take $x_{k+1} = x_k + e_k$ where e_k is from a zero-mean Gaussian process of variance $S_k = \epsilon^2$. Thus in this example, $\Phi_k = 1$ and $b_k = 0$. The Kalman filter equations become

$$\tilde{P}_k = \frac{P_k\sigma^2}{\sigma^2 + P_k}, \quad K_k = \frac{P_k}{\sigma^2 + P_k}$$

$$\tilde{x}_k = \overline{x}_k + K_k(y_k - \overline{x}_k), \quad P_{k+1} = \epsilon^2 + \tilde{P}_k \quad \overline{x}_{k+1} = \tilde{x}_k$$

Starting with $P_1 = \infty$ and writing $\rho = (\epsilon/\sigma)^2$ we find

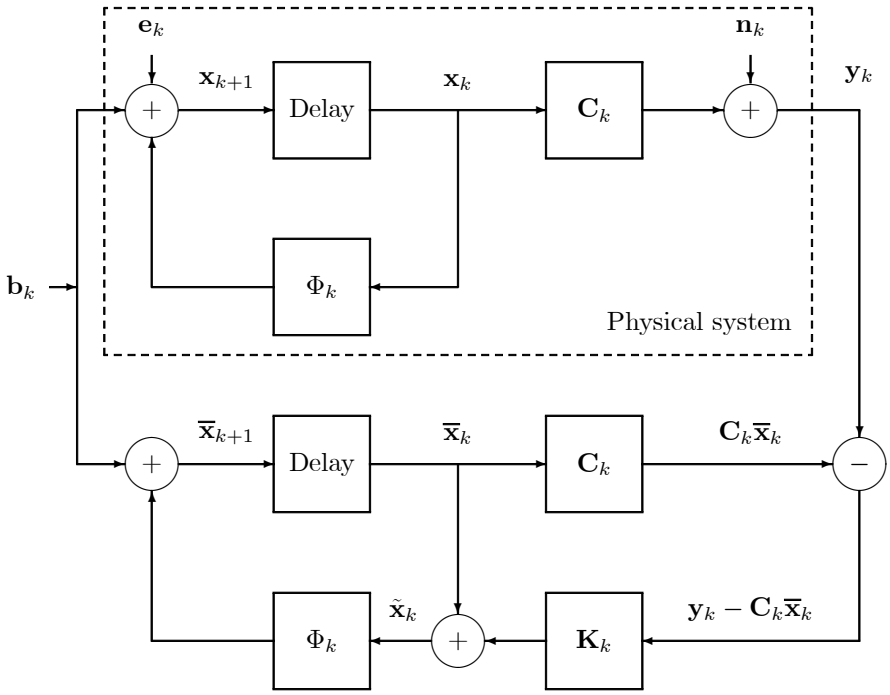
$$P_1 = \infty, \quad \tilde{P}_1 = \sigma^2, \quad K_1 = 1, \quad \overline{x}_2 = \tilde{x}_1 = y_1$$

$$P_2 = (1+\rho)\sigma^2, \quad \tilde{P}_2 = \left(\frac{1+\rho}{2+\rho}\right)\sigma^2, \quad K_2 = \frac{1+\rho}{2+\rho}, \quad \overline{x}_3 = \tilde{x}_2 = \left(\frac{1+\rho}{2+\rho}\right)y_2 + \left(\frac{1}{2+\rho}\right)y_1$$

$$P_3 = \left(\frac{1+3\rho+\rho^2}{2+\rho}\right)\sigma^2, \quad \tilde{P}_3 = \left(\frac{1+3\rho+\rho^2}{3+4\rho+\rho^2}\right)\sigma^2, \quad K_3 = \frac{1+3\rho+\rho^2}{3+4\rho+\rho^2},$$

$$\overline{x}_4 = \tilde{x}_3 = \left(\frac{1+3\rho+\rho^2}{3+4\rho+\rho^2}\right)y_3 + \left(\frac{1+\rho}{3+4\rho+\rho^2}\right)y_2 + \left(\frac{1}{3+4\rho+\rho^2}\right)y_1$$

We see that $\overline{x}_k$ is always a weighted average of the data. These results reduce to those of the recursive least-squares estimator if $\epsilon \to 0$ so that $\rho \to 0$. In this limit, we are certain that x_k remains constant and so the optimal processing involves an equal-weight average of all the data points which have been collected. On the other hand, as ρ is increased, the optimal average weights the most recent point heavily and the weights decrease for the previous points. This is reasonable since a non-zero ϵ means that the x_k can drift and so the optimal filter must be able to track this. In the limit of $\rho \to \infty$, the dynamical evolution is ignored completely and the best estimate $\tilde{x}_k$ is simply the most recent datum y_k. This is appropriate either if there is no error in the observation $\sigma = 0$ or if the innovation variance ϵ is very large.



6.4 The Extended Kalman Filter

We briefly discuss the situation of nonlinear dynamical evolution and observation processes since these commonly occur in practice. Of course such nonlinearities mean that the likelihood function fails to be Gaussian in the parameters even when the noise is additive and Gaussian. If the nonlinearities are not too severe and the fluctuations are small, we may approximate the posterior probability functions by Gaussians and proceed using linearized models.

Non-linear dynamical evolution

The dynamical evolution of the system enters into the Kalman filter algorithm in two ways. In the linear case, this evolution is specified by the matrix Φ and the vector $\mathbf{b}$ and they are used for updating the covariance

matrix via

$$\mathbf{P}_{k+1} = \mathbf{S}_k + \Phi_k \tilde{\mathbf{P}}_k \Phi_k^t \tag{6.53}$$

and for updating the state vector via

$$\overline{\mathbf{x}}_{k+1} = \Phi_k \tilde{\mathbf{x}}_k + \mathbf{b}_k \tag{6.54}$$

For general nonlinear evolution, we have

$$\mathbf{x}_{k+1} = \mathbf{F}(k, \mathbf{x}_k) + \mathbf{e}_k \tag{6.55}$$

for some function $\mathbf{F}$. Suppose that after measuring $\mathbf{y}_k$, we represent our state of knowledge of $\mathbf{x}_k$ by a Gaussian with mean $\tilde{\mathbf{x}}_k$ and covariance $\tilde{\mathbf{P}}_k$. Linearizing $\mathbf{F}$ about $\tilde{\mathbf{x}}_k$, we find

$$\mathbf{F}(k, \mathbf{x}_k) \approx \mathbf{F}(k, \tilde{\mathbf{x}}_k) + J\mathbf{F}(k, \tilde{\mathbf{x}}_k)(\mathbf{x}_k - \tilde{\mathbf{x}}_k) \tag{6.56}$$

where $J\mathbf{F}$ is the Jacobian matrix of $\mathbf{F}$ at $\tilde{\mathbf{x}}_k$, i.e., the ij'th component of $J\mathbf{F}$ is $\partial F_i/\partial x_{kj}$ and x_{kj} is the j'th component of $\mathbf{x}_k$.

Taking the expectation value of (6.55) yields the mean for our prior of $\mathbf{x}_{k+1}$. Using the linear approximation for $\mathbf{F}$ and realizing that the mean of $\mathbf{x}_k - \tilde{\mathbf{x}}_k$ is zero,

$$\overline{\mathbf{x}}_{k+1} = \mathbf{F}(k, \tilde{\mathbf{x}}_k) \tag{6.57}$$

Similarly, we can calculate the covariance $\mathbf{P}_{k+1}$ from the expectation value of $(\mathbf{x}_{k+1} - \overline{\mathbf{x}}_{k+1})(\mathbf{x}_{k+1} - \overline{\mathbf{x}}_{k+1})^t$. This yields

$$\mathbf{P}_{k+1} = \mathbf{S}_k + (J\mathbf{F}(k, \tilde{\mathbf{x}}_k))\tilde{\mathbf{P}}_k(J\mathbf{F}(k, \tilde{\mathbf{x}}_k))^t \tag{6.58}$$

From these it is clear that the Jacobian of $\mathbf{F}$ is used instead of Φ_k in (6.53) for updating the covariance matrix whereas the state update (6.54) is performed using the full non-linear evolution $\mathbf{F}$.

As an example, consider the following dynamical system which defines an oscillator with unknown angular frequency $\sqrt{\gamma}$. The non-linear equations of motion are

$$\dot{x} = v \tag{6.59}$$
$$\dot{v} = -\gamma x \tag{6.60}$$
$$\dot{\gamma} = e(t) \tag{6.61}$$

where x and v are the position and velocity of the oscillator and $e(t)$ represents a stochastic innovation process. We first need to convert these into equations in discrete time. One way of doing this is to use the midpoint method to integrate the differential equations approximately from time kT to time $(k+1)T$. This involves the calculation of the half-step variables

$$\hat{x}\left(\left[k+\frac{1}{2}\right]T\right) = x(kT) + \frac{1}{2}Tv(kT) \tag{6.62}$$
$$\hat{v}\left(\left[k+\frac{1}{2}\right]T\right) = v(kT) - \frac{1}{2}T\gamma(kT)x(kT) \tag{6.63}$$
$$\hat{\gamma}\left(\left[k+\frac{1}{2}\right]T\right) = \gamma(kT) + \frac{1}{2}Te(kT) \tag{6.64}$$

These are then used to approximate the full-step integral

$$x((k+1)T) = x(kT) + T\hat{v}\left(\left[k+\frac{1}{2}\right]T\right) = x(kT) + T\left[v(kT) - \frac{1}{2}T\gamma(kT)x(kT)\right] \tag{6.65}$$

$$\begin{aligned} v((k+1)T) &= v(kT) - T\hat{\gamma}\left(\left[k+\frac{1}{2}\right]T\right)\hat{x}\left(\left[k+\frac{1}{2}\right]T\right) \\ &= v(kT) - T\left[\gamma(kT) + \frac{1}{2}Te(kT)\right]\left[x(kT) + \frac{1}{2}Tv(kT)\right] \end{aligned} \tag{6.66}$$

$$\gamma((k+1)T) = \gamma(kT) + Te(kT) \tag{6.67}$$

These equations are used to carry out the single-step time-evolution. The Jacobian of the transformation is

$$\begin{pmatrix} \frac{\partial x([k+1]T)}{\partial x(kT)} & \frac{\partial x([k+1]T)}{\partial v(kT)} & \frac{\partial x([k+1]T)}{\partial \gamma(kT)} \\ \frac{\partial v([k+1]T)}{\partial x(kT)} & \frac{\partial v([k+1]T)}{\partial v(kT)} & \frac{\partial v([k+1]T)}{\partial \gamma(kT)} \\ \frac{\partial \gamma([k+1]T)}{\partial x(kT)} & \frac{\partial \gamma([k+1]T)}{\partial v(kT)} & \frac{\partial \gamma([k+1]T)}{\partial \gamma(kT)} \end{pmatrix} = \begin{pmatrix} 1-\frac{1}{2}T^2\gamma & T & -\frac{1}{2}T^2 x \\ -T\gamma & 1-\frac{1}{2}T^2\gamma & -Tx-\frac{1}{2}vT^2 \\ 0 & 0 & 1 \end{pmatrix} \tag{6.68}$$

The linearized form of the innovation process involves the matrix $\mathbf{E}_k$ where

$$\mathbf{E}_k = \begin{pmatrix} \frac{\partial x([k+1]T)}{\partial e(kT)} \\ \frac{\partial v([k+1]T)}{\partial e(kT)} \\ \frac{\partial \gamma([k+1]T)}{\partial e(kT)} \end{pmatrix} = \begin{pmatrix} 0 \\ -\frac{1}{2}T^2 x + \frac{1}{4}T^3 v \\ T \end{pmatrix} \tag{6.69}$$

where all terms on the right-hand sides are evaluated at kT. The innovation covariance matrix $\mathbf{S}_k$ is then $\mathbf{E}_k \epsilon^2 \mathbf{E}_k^t$ where ϵ^2 is the variance in $e(kT)$.

Non-linear observation process

The observation process which is characterized by $\mathbf{C}_k$ in the linear case also enters into the Kalman filter algorithm in two ways. Firstly, it is used to calculate the Kalman gain and update the covariance

$$\tilde{\mathbf{P}}_k = \mathbf{P}_k - \mathbf{P}_k \mathbf{C}_k^t (\Gamma_k + \mathbf{C}_k \mathbf{P}_k \mathbf{C}_k^t)^{-1} \mathbf{C}_k \mathbf{P}_k \tag{6.70}$$

$$\mathbf{K}_k = \tilde{\mathbf{P}}_k \mathbf{C}_k^t \Gamma_k^{-1} \tag{6.71}$$

For a nonlinear observation process, we replace $\mathbf{C}_k$ by the Jacobian of the process.

Secondly, the residual $\mathbf{y}_k - \mathbf{C}_k \overline{\mathbf{x}}_k$ is used to update $\overline{\mathbf{x}}_k$ to obtain $\tilde{\mathbf{x}}_k$. For a nonlinear process, we calculate the residual using the actual nonlinear function in place of $\mathbf{C}_k \overline{\mathbf{x}}_k$.

It should be realized that the extended Kalman filter is only an approximation which often works for problems in which the nonlinearities are not too severe. It is not difficult to construct situations in which it fails or gives strange results.