CHAPTER

MEAN-SQUARE ERROR LINEAR ESTIMATION

4

CHAPTER OUTLINE

4.1 INTRODUCTION

Mean-square error linear estimation is a topic of fundamental importance for parameter estimation in statistical learning. Besides historical reasons, which take us back to the pioneering works of Kolmogorov, Wiener, and Kalman, who laid the foundations of the optimal estimation field, understanding

Machine Learning. http://dx.doi.org/10.1016/B978-0-12-801522-3.00004-5

mean-square error estimation is a must, prior to studying more recent techniques. One always has to grasp the basics and learn the classics prior to getting involved with new “adventures.” Many of the concepts to be discussed in this chapter are also used in the next chapters.

Optimizing via a loss function, which builds around the square of the error, has a number of advantages such as a single optimal value, which can be obtained via the solution of a linear set of equations; this is a very attractive feature in practice. Moreover, due to the relative simplicity of the resulting equations, the newcomer in the field can get a better feeling of the various notions associated with optimal parameter estimation. The elegant geometric interpretation of the mean-square error solution, via the orthogonality theorem, is presented and discussed. In the chapter, emphasis is also given to computational complexity issues while solving for the optimal solution. The essence behind these techniques remains exactly the same as that inspiring a number of computationally efficient schemes for online learning, to be discussed later in this book.

The development of the chapter is around real-valued variables, something that will be true for most of the book. However, complex-valued signals are particularly useful in a number of areas, with communications being a typical example, and the generalization from the real to the complex domain may not always be trivial. Although in most of the cases, the difference lies in changing matrix transpositions by Hermitian ones, this is not the whole story. This is the reason that we chose to deal with complex-valued data in separate sections, whenever the differences from the real data are not trivial and some subtle issues are involved.

4.2 MEAN-SQUARE ERROR LINEAR ESTIMATION: THE NORMAL EQUATIONS

The general estimation task has been introduced in Chapter 3. There, it was stated that given two dependent random vectors, $\mathbf{y}$ and $\mathbf{x}$, the goal of the estimation task is to obtain a function, g, so as, given a value $\boldsymbol{x}$ of $\mathbf{x}$, to be able to predict (estimate), in some optimal sense, the corresponding value $\boldsymbol{y}$ of $\mathbf{y}$, or $\hat{\boldsymbol{y}} = g(\boldsymbol{x})$. The mean-square error (MSE) estimation was also presented in Chapter 3 and it was shown that the optimal MSE estimate of $\mathbf{y}$ given the value $\mathbf{x} = \boldsymbol{x}$ is

$$\hat{\boldsymbol{y}} = \mathbb{E}[\mathbf{y}|\boldsymbol{x}].$$

In general, this is a nonlinear function. We now turn our attention to the case where g is *constrained* to be a linear function. For simplicity and in order to pay more attention to the concepts, we will restrict our discussion to the case of scalar dependent variables. The more general case will be discussed later on.

Let $(\mathrm{y}, \mathbf{x}) \in \mathbb{R} \times \mathbb{R}^l$ be two jointly distributed random entities of *zero mean values*. In case the mean values are not zero, they are subtracted. Our goal is to obtain an estimate of $\boldsymbol{\theta} \in \mathbb{R}^l$ in the linear estimator model,

$$\hat{\mathrm{y}} = \boldsymbol{\theta}^{\mathrm{T}}\mathbf{x}, \tag{4.1}$$

so that

$$J(\boldsymbol{\theta}) = \mathbb{E}[(\mathrm{y} - \hat{\mathrm{y}})^2], \tag{4.2}$$

is minimum, or

$$\boldsymbol{\theta}_* := \arg\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta}). \tag{4.3}$$

In other words, the optimal estimator is chosen so as to minimize the variance of the error random variable

$$\mathrm{e} = \mathrm{y} - \hat{\mathrm{y}}. \tag{4.4}$$

Minimizing the cost function $J(\boldsymbol{\theta})$ is equivalent with setting its gradient with respect to $\boldsymbol{\theta}$ equal to zero,

$$\begin{aligned}
\nabla J(\boldsymbol{\theta}) &= \nabla\, \mathbb{E}\left[\left(\mathrm{y} - \boldsymbol{\theta}^T \mathbf{x}\right)\left(\mathrm{y} - \mathbf{x}^T \boldsymbol{\theta}\right)\right] \\
&= \nabla \left\{ \mathbb{E}[\mathrm{y}^2] - 2\boldsymbol{\theta}^T\, \mathbb{E}[\mathbf{x}\mathrm{y}] + \boldsymbol{\theta}^T\, \mathbb{E}[\mathbf{x}\mathbf{x}^T]\boldsymbol{\theta} \right\} \\
&= -2\boldsymbol{p} + 2\Sigma_x \boldsymbol{\theta} = \mathbf{0}
\end{aligned}$$

or

$$\boxed{\Sigma_x \boldsymbol{\theta}_* = \boldsymbol{p}: \quad \text{Normal Equations,}} \tag{4.5}$$

where the input-output cross-correlation vector $\boldsymbol{p}$ is given by[1]

$$\boldsymbol{p} = \left[\mathbb{E}[\mathrm{x}_1 \mathrm{y}], \ldots, \mathbb{E}[\mathrm{x}_l \mathrm{y}]\right]^T = \mathbb{E}[\mathbf{x}\mathrm{y}], \tag{4.6}$$

and the respective covariance matrix is given by

$$\Sigma_x = \mathbb{E}\left[\mathbf{x}\mathbf{x}^T\right].$$

Thus, the weights of the optimal linear estimator are obtained via a linear system of equations, provided that the covariance matrix is *positive definite* and hence it can be inverted. Moreover, in this case, the solution is *unique*. On the contrary, if Σ_x is singular and hence cannot be inverted, there are infinitely many solutions (Problem 4.1).

4.2.1 THE COST FUNCTION SURFACE

Elaborating on the cost function, $J(\boldsymbol{\theta})$, as it is defined in (4.2), we get that

$$J(\boldsymbol{\theta}) = \sigma_y^2 - 2\boldsymbol{\theta}^T \boldsymbol{p} + \boldsymbol{\theta}^T \Sigma_x \boldsymbol{\theta}. \tag{4.7}$$

[1] The cross-correlation vector is usually denoted as $\boldsymbol{r}_{xy}$. Here we will use $\boldsymbol{p}$, in order to simplify the notation.

Adding and subtracting the term $\boldsymbol{\theta}_*^T \Sigma_x \boldsymbol{\theta}_*$ and taking into account the definition of $\boldsymbol{\theta}_*$ from (4.5), it is readily seen that

$$\boxed{J(\boldsymbol{\theta}) = J(\boldsymbol{\theta}_*) + (\boldsymbol{\theta} - \boldsymbol{\theta}_*)^T \Sigma_x (\boldsymbol{\theta} - \boldsymbol{\theta}_*),} \tag{4.8}$$

where

$$J(\boldsymbol{\theta}_*) = \sigma_y^2 - \boldsymbol{p}^T \Sigma_x^{-1} \boldsymbol{p} = \sigma_y^2 - \boldsymbol{\theta}_*^T \Sigma_x \boldsymbol{\theta}_* = \sigma_y^2 - \boldsymbol{p}^T \boldsymbol{\theta}_*, \tag{4.9}$$

is the minimum achieved at the optimal solution. From (4.8) and (4.9), the following remarks can be made.

Remarks 4.1.

- The cost at the optimal value $\boldsymbol{\theta}_*$ is always less than the variance $\mathbb{E}[\mathrm{y}^2]$ of the output variable. This is guaranteed by the positive definite nature of Σ_x or Σ_x^{-1}, which makes the second term on the right-hand side in (4.9) always positive, unless $\boldsymbol{p} = \boldsymbol{0}$; however, the cross-correlation vector will only be zero if **x** and y are uncorrelated. Well, in this case, one cannot say anything (make any prediction) about y by observing samples of **x**, at least as far as the MSE criterion is concerned, which turns out to involve information residing up to the second order statistics. In this case, the variance of the error, which coincides with $J(\boldsymbol{\theta}_*)$, will be equal to the variance σ_y^2; the latter is a measure of the “intrinsic” uncertainty of y around its (zero) mean value. On the contrary, if the input-output variables are correlated, then observing **x** removes part of the uncertainty associated with y.
- For any value $\boldsymbol{\theta}$, other than the optimal $\boldsymbol{\theta}_*$, the error variance increases as (4.8) suggests, due to the positive definite nature of Σ_x. Figure 4.1 shows the cost function (mean-square error) surface defined by $J(\boldsymbol{\theta})$ in (4.8). The corresponding isovalue contours are shown in Figure 4.2. In general, they are ellipses, whose axes are determined by the eigenstructure of Σ_x. For $\Sigma_x = \sigma^2 I$, where all eigenvalues are equal to σ^2, the contours are circles (Problem 4.3).

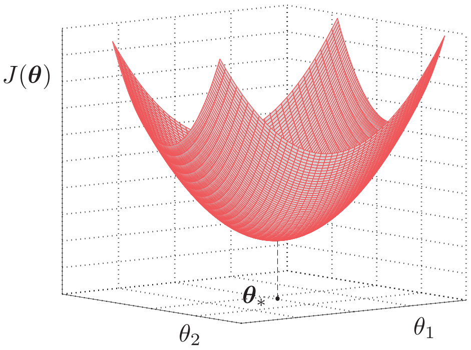


FIGURE 4.1

The MSE cost function has the form of a (hyper) paraboloid.

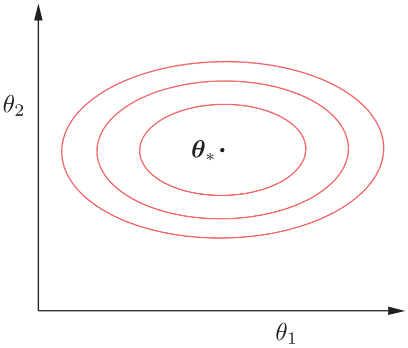


FIGURE 4.2

The isovalue contours for the cost function surface corresponding to Figure 4.1. They are ellipses; the major axis of each ellipse is determined by the maximum eigenvalue $\lambda_{\max}$ and the minor one by the smaller $\lambda_{\min}$ of the Σ of the input random variables. The largest the ratio $\frac{\lambda_{\max}}{\lambda_{\min}}$ is the more elongated the ellipse becomes. The ellipses become circles, if the covariance matrix has the special form of $\sigma^2 I$. That is, all variables are mutually uncorrelated and they have the same variance. By varying Σ, different shapes of the ellipses and different orientations result.

4.3 A GEOMETRIC VIEWPOINT: ORTHOGONALITY CONDITION

A very intuitive view of what we have said so far comes from the geometric interpretation of the random variables. The reader can easily check out that the set of random variables is a *vector space* over the field of real (and complex) numbers. Indeed, if x and y are any two random variables then $\mathrm{x} + \mathrm{y}$, as well as $\alpha\mathrm{x}$, are also random variables for every $\alpha \in \mathbb{R}$.[2] We can now equip this vector space with an inner product operation, which also implies a norm and makes it a *Euclidean space*. The reader can easily check out that the mean value operation has all the properties required for an operation to be called an inner product. Indeed, for any subset of random variables,

- $\mathbb{E}[\mathrm{xy}] = \mathbb{E}[\mathrm{yx}]$,
- $\mathbb{E}[(\alpha_1 \mathrm{x}_1 + \alpha_2 \mathrm{x}_2)\mathrm{y}] = \alpha_1\, \mathbb{E}[\mathrm{x}_1\mathrm{y}] + \alpha_2\, \mathbb{E}[\mathrm{x}_2\mathrm{y}]$,
- $\mathbb{E}[\mathrm{x}^2] \geq 0$, with equality if and only if $\mathrm{x} = 0$.

Thus, the norm induced by this inner product,

$$\|\mathrm{x}\| := \sqrt{\mathbb{E}[\mathrm{x}^2]},$$

coincides with the respective standard deviation (assuming $\mathbb{E}[\mathrm{x}] = 0$). From now on, given two uncorrelated random variables, x, y, or $\mathbb{E}[\mathrm{xy}] = 0$, we can call them *orthogonal*, because their inner product is zero. We are now free to apply to our task of interest any one of the theorems that have been derived for Euclidean spaces.

[2] These operations also satisfy all the properties required for a set to be a vector space, including associativity, commutativity, and so on (see [47] and Section 8.15).

Let us now rewrite (4.1) as

$$\hat{\mathrm{y}} = \theta_1 \mathrm{x}_1 + \cdots + \theta_l \mathrm{x}_l.$$

Thus, the random variable, $\hat{\mathrm{y}}$, which is now interpreted as a point in a vector space, results as a linear combination of l elements in this space. Thus, the estimate, $\hat{\mathrm{y}}$, will necessarily lie in the subspace spanned by these points. In contrast, the true variable, y, will not lie, in general, in this subspace. Because our goal is to obtain a $\hat{\mathrm{y}}$ that is a good approximation of y, we have to seek the specific linear combination that makes the norm of the error, $\mathrm{e} = \mathrm{y} - \hat{\mathrm{y}}$, minimum. This specific linear combination corresponds to the *orthogonal* projection of y onto the subspace spanned by the points $\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_l$. This is equivalent with requiring

$$\boxed{\mathbb{E}[\mathrm{e}\mathrm{x}_k] = 0, \quad k = 1, \ldots, l: \quad \text{Orthogonality Condition.}} \tag{4.10}$$

The error variable being orthogonal to every point x_k, $k = 1, 2, \ldots, l$, will be orthogonal to the respective subspace. This is illustrated in Figure 4.3. Such a choice guarantees that the resulting error will have the minimum norm; by the definition of the norm, this corresponds to the minimum MSE, or $\mathbb{E}[\mathrm{e}^2]$.

The set of equations in (4.10) can now be written as

$$\mathbb{E}\left[\left(\mathrm{y} - \sum_{i=1}^{l} \theta_i \mathrm{x}_i\right) \mathrm{x}_k\right] = 0, \quad k = 1, 2, \ldots, l,$$

or

$$\sum_{i=1}^{l} \mathbb{E}[\mathrm{x}_i \mathrm{x}_k] \theta_i = \mathbb{E}[\mathrm{x}_k \mathrm{y}], \quad k = 1, 2, \ldots, l, \tag{4.11}$$

which leads to the linear set of equations in (4.5).

This is the reason that this elegant set of equations is known as *normal equations*. Another name is *Wiener-Hopf equations*. Strictly speaking, the Wiener-Hopf equations were first derived for continuous time processes in the context of the causal estimation task [49, 50]; for a discussion see [16, 44].

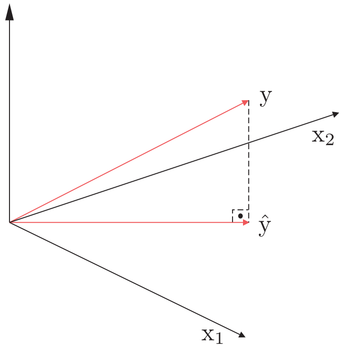


FIGURE 4.3

Projecting y on the subspace spanned by $\mathrm{x}_1, \mathrm{x}_2$ guarantees that the deviation between y and $\hat{\mathrm{y}}$ corresponds to the minimum MSE.

Nobert Wiener was a mathematician and philosopher. He was awarded a PhD at Harvard at the age of 17 in mathematical logic. During the Second World War, he laid the foundations of linear estimation theory in a classified work, independently of Kolmogorov. Later on, Wiener was involved in pioneering work embracing automation, artificial intelligence, and cognitive science. Being a pacifist, he was regarded with suspicion during the Cold War years.

The other pillar on which linear estimation theory is based is the pioneering work of Andrey Nikolaevich Kolmogorov (1903-1987) [24], who developed his theory independent of Wiener. Kolmogorov's contributions cover a wide range of topics in mathematics, including probability, computational complexity, and topology. He is the father of the modern axiomatic foundation of the notion of probability, see Chapter 2.

Remarks 4.2.

- So far, in our theoretical findings, we have assumed that **x** and y are jointly distributed (correlated) variables. If, in addition, we assume that they are linearly related according to the linear regression model,

$$\mathrm{y} = \boldsymbol{\theta}_o^{\mathrm{T}}\mathbf{x} + \eta, \quad \boldsymbol{\theta}_o \in \mathbb{R}^k, \tag{4.12}$$

  where η is a zero mean noise variable independent of **x**, then, if the dimension, k, of the true system, $\boldsymbol{\theta}_o$, is equal to the number of parameters, l, adopted for the model, so that the $k = l$, it turns out that (Problem 4.4)

$$\boldsymbol{\theta}_* = \boldsymbol{\theta}_o,$$

  and the optimal MSE is equal to the variance of the noise, σ_η^2.
- *Undermodeling*. If $k > l$, then the order of the model is less than that of the true system, which relates y and **x** in (4.12); this is known as *undermodeling*. It is easy to show that if the variables comprising **x** are uncorrelated then (Problem 4.5),

$$\boldsymbol{\theta}_* = \boldsymbol{\theta}_o^1,$$

  where

$$\boldsymbol{\theta}_o := \begin{bmatrix} \boldsymbol{\theta}_o^1 \\ \boldsymbol{\theta}_o^2 \end{bmatrix}, \quad \boldsymbol{\theta}_o^1 \in \mathbb{R}^l, \quad \boldsymbol{\theta}_o^2 \in \mathbb{R}^{k-l}.$$

  In other words, the MSE optimal estimator identifies the first l components of $\boldsymbol{\theta}_o$.

4.4 EXTENSION TO COMPLEX-VALUED VARIABLES

Everything that has been said so far can be extended to complex-valued signals. However, there are a few subtle points involved and this is the reason that we chose to treat this case separately. Complex-valued variables are very common in a number of applications, as for example in communications and fMRI [2, 41].

Given two real-valued variables, (x, y), one can consider them either as a vector quantity in the two-dimensional space, $[x, y]^{\mathrm{T}}$, or can describe them as a complex variable, $z = x + jy$, where $j^2 := -1$.

Adopting the latter approach offers the luxury of exploiting the operations available in the field $\mathbb{C}$ of complex numbers, in other words multiplication and division. The existence of such operations greatly facilitates the algebraic manipulations. Recall that such operations are not defined in vector spaces.[3]

Let us assume that we are given a complex-valued (output) random variable

$$\mathrm{y} := \mathrm{y}_r + j\mathrm{y}_i, \tag{4.13}$$

and a complex-valued (input) random vector

$$\mathbf{x} = \mathbf{x}_r + j\mathbf{x}_i. \tag{4.14}$$

The quantities $\mathrm{y}_r, \mathrm{y}_i, \mathbf{x}_r$, and $\mathbf{x}_i$ are real-valued random variables/vectors. The goal is to compute a linear estimator defined by a complex-valued parameter vector $\boldsymbol{\theta} = \boldsymbol{\theta}_r + j\boldsymbol{\theta}_i \in \mathbb{C}^l$, so as to minimize the respective mean-square error,

$$\mathbb{E}\left[|\mathrm{e}|^2\right] := \mathbb{E}\left[\mathrm{e}\mathrm{e}^*\right] = \mathbb{E}\left[|\mathrm{y} - \boldsymbol{\theta}^H\mathbf{x}|^2\right]. \tag{4.15}$$

Looking at (4.15), it is readily observed that in the case of complex variables the inner product operation between two complex-valued random variables should be defined as $\mathbb{E}[\mathrm{x}\mathrm{y}^*]$, so as to guarantee that the implied norm by the inner product, $\|\mathrm{x}\| = \sqrt{\mathbb{E}[\mathrm{x}\mathrm{x}^*]}$, is a valid quantity. Applying the orthogonality condition as before, we rederive the normal equations as in (4.11),

$$\Sigma_x\boldsymbol{\theta}_* = \boldsymbol{p}, \tag{4.16}$$

where now the covariance matrix and cross-correlation vector are given by

$$\Sigma_x = \mathbb{E}\left[\mathbf{x}\mathbf{x}^{\mathrm{H}}\right], \tag{4.17}$$

$$\boldsymbol{p} = \mathbb{E}\left[\mathbf{x}\mathrm{y}^*\right]. \tag{4.18}$$

Note that (4.16)-(4.18) can alternatively be obtained by minimizing (4.15) (Problem 4.6). Moreover, the counterpart of (4.9) is given by

$$J(\boldsymbol{\theta}_*) = \sigma_y^2 - \boldsymbol{p}^{\mathrm{H}}\Sigma_x^{-1}\boldsymbol{p} = \sigma_y^2 - \boldsymbol{p}^{\mathrm{H}}\boldsymbol{\theta}_*. \tag{4.19}$$

Using the definitions in (4.13) and (4.14), the cost in (4.15) is written as,

$$\begin{aligned} J(\boldsymbol{\theta}) &= \mathbb{E}[|\mathrm{e}|^2] = \mathbb{E}[|\mathrm{y} - \hat{\mathrm{y}}|^2] \\ &= \mathbb{E}[|\mathrm{y}_r - \hat{\mathrm{y}}_r|^2] + \mathbb{E}[|\mathrm{y}_i - \hat{\mathrm{y}}_i|^2], \end{aligned} \tag{4.20}$$

where

$$\boxed{\hat{\mathrm{y}} := \hat{\mathrm{y}}_r + j\hat{\mathrm{y}}_i = \boldsymbol{\theta}^{\mathrm{H}}\mathbf{x}: \quad \text{Complex Linear Estimator,}} \tag{4.21}$$

[3] Multiplication and division can also be defined for groups of four variables $(\mathrm{x}, \phi, \mathrm{z}, \mathrm{y})$ known as quaternions; the related algebra was introduced by Hamilton in 1843. The real and complex numbers as well as quaternions are all special cases of the so-called Clifford algebras [39].

or

$$\begin{aligned}\hat{y} &= (\boldsymbol{\theta}_r^{\mathrm{T}} - j\boldsymbol{\theta}_i^{\mathrm{T}})(\mathbf{x}_r + j\mathbf{x}_i)\\ &= (\boldsymbol{\theta}_r^{\mathrm{T}}\mathbf{x}_r + \boldsymbol{\theta}_i^{\mathrm{T}}\mathbf{x}_i) + j(\boldsymbol{\theta}_r^{\mathrm{T}}\mathbf{x}_i - \boldsymbol{\theta}_i^{\mathrm{T}}\mathbf{x}_r).\end{aligned} \tag{4.22}$$

Equation (4.22) reveals the true flavor behind the complex notation; that is, its *multichannel* nature. In multichannel estimation, we are given more than one set of input variables, namely $\mathbf{x}_r$ and $\mathbf{x}_i$, and we want to generate, jointly, more than one output variable, namely $\hat{y}_r$ and $\hat{y}_i$. Equation (4.22) can equivalently be written as

$$\begin{bmatrix}\hat{y}_r\\ \hat{y}_i\end{bmatrix} = \Theta \begin{bmatrix}\mathbf{x}_r\\ \mathbf{x}_i\end{bmatrix}, \tag{4.23}$$

where

$$\Theta := \begin{bmatrix}\boldsymbol{\theta}_r^{\mathrm{T}} & \boldsymbol{\theta}_i^{\mathrm{T}}\\ -\boldsymbol{\theta}_i^{\mathrm{T}} & \boldsymbol{\theta}_r^{\mathrm{T}}\end{bmatrix}. \tag{4.24}$$

Multichannel estimation can be generalized to more than two outputs and to more than two input sets of variables. We will come back to the more general multichannel estimation task toward the end of this chapter.

Looking at (4.23), we observe that starting from the direct generalization of the linear estimation task for real-valued signals, which led to the adoption of $\hat{y} = \boldsymbol{\theta}^{\mathrm{H}}\mathbf{x}$, resulted in a matrix, Θ, of a *very special structure*.

4.4.1 WIDELY LINEAR COMPLEX-VALUED ESTIMATION

Let us define the linear two-channel estimation task starting from the definition of a linear operation in vector spaces. The task is to generate a vector output, $\hat{\mathbf{y}} = [\hat{y}_r, \hat{y}_i]^{\mathrm{T}} \in \mathbb{R}^2$ from the input vector variables, $\mathbf{x} = [\mathbf{x}_r^T, \mathbf{x}_i^T]^T \in \mathbb{R}^{2l}$, via the linear operation,

$$\hat{\mathbf{y}} = \begin{bmatrix}\hat{y}_r\\ \hat{y}_i\end{bmatrix} = \Theta \begin{bmatrix}\mathbf{x}_r\\ \mathbf{x}_i\end{bmatrix}, \tag{4.25}$$

where

$$\Theta := \begin{bmatrix}\boldsymbol{\theta}_{11}^{\mathrm{T}} & \boldsymbol{\theta}_{12}^{\mathrm{T}}\\ \boldsymbol{\theta}_{21}^{\mathrm{T}} & \boldsymbol{\theta}_{22}^{\mathrm{T}}\end{bmatrix}, \tag{4.26}$$

and compute the matrix Θ so as to minimize the total error variance

$$\Theta_* := \arg\min_{\Theta} \left\{\mathbb{E}\left[(\mathrm{y}_r - \hat{\mathrm{y}}_r)^2\right] + \mathbb{E}\left[(\mathrm{y}_i - \hat{\mathrm{y}}_i)^2\right]\right\}. \tag{4.27}$$

Note that (4.27) can equivalently be written as

$$\Theta_* := \arg\min_{\Theta} \left\{\mathbb{E}[\mathbf{e}^{\mathrm{T}}\mathbf{e}]\right\} = \arg\min_{\Theta} \left\{\mathrm{trace}\{\mathbb{E}[\mathbf{e}\mathbf{e}^{\mathrm{T}}]\}\right\},$$

where

$$\mathbf{e} := \mathbf{y} - \hat{\mathbf{y}}.$$

Minimizing (4.27) is equivalent with minimizing the two terms individually; in other words, treating each channel separately (Problem 4.7). Thus, the task can be tackled by solving two sets of normal equations, namely

$$\Sigma_\varepsilon \begin{bmatrix} \boldsymbol{\theta}_{11} \\ \boldsymbol{\theta}_{12} \end{bmatrix} = \boldsymbol{p}_r, \quad \Sigma_\varepsilon \begin{bmatrix} \boldsymbol{\theta}_{21} \\ \boldsymbol{\theta}_{22} \end{bmatrix} = \boldsymbol{p}_i, \tag{4.28}$$

where

$$\begin{aligned} \Sigma_\varepsilon &:= \mathbb{E}\left[\begin{bmatrix} \mathbf{x}_r \\ \mathbf{x}_i \end{bmatrix} \begin{bmatrix} \mathbf{x}_r^T, & \mathbf{x}_i^T \end{bmatrix} \right] \\ &= \begin{bmatrix} \mathbb{E}[\mathbf{x}_r\mathbf{x}_r^T] & \mathbb{E}[\mathbf{x}_r\mathbf{x}_i^T] \\ \mathbb{E}[\mathbf{x}_i\mathbf{x}_r^T] & \mathbb{E}[\mathbf{x}_i\mathbf{x}_i^T] \end{bmatrix} := \begin{bmatrix} \Sigma_r & \Sigma_{ri} \\ \Sigma_{ir} & \Sigma_i \end{bmatrix}, \end{aligned} \tag{4.29}$$

and

$$\boldsymbol{p}_r := \mathbb{E}\begin{bmatrix} \mathbf{x}_r \mathrm{y}_r \\ \mathbf{x}_i \mathrm{y}_r \end{bmatrix}, \quad \boldsymbol{p}_i := \mathbb{E}\begin{bmatrix} \mathbf{x}_r \mathrm{y}_i \\ \mathbf{x}_i \mathrm{y}_i \end{bmatrix}. \tag{4.30}$$

The obvious question that is now raised is whether we can tackle this more general task of the two-channel linear estimation task by employing complex-valued arithmetic. The answer is in the affirmative. Let us define

$$\boldsymbol{\theta} := \boldsymbol{\theta}_r + j\boldsymbol{\theta}_i, \quad \boldsymbol{v} := \boldsymbol{v}_r + j\boldsymbol{v}_i, \tag{4.31}$$

and

$$\mathbf{x} = \mathbf{x}_r + j\mathbf{x}_i.$$

Then define

$$\boldsymbol{\theta}_r := \frac{1}{2}(\boldsymbol{\theta}_{11} + \boldsymbol{\theta}_{22}), \quad \boldsymbol{\theta}_i := \frac{1}{2}(\boldsymbol{\theta}_{12} - \boldsymbol{\theta}_{21}), \tag{4.32}$$

and

$$\boldsymbol{v}_r := \frac{1}{2}(\boldsymbol{\theta}_{11} - \boldsymbol{\theta}_{22}), \quad \boldsymbol{v}_i := -\frac{1}{2}(\boldsymbol{\theta}_{12} + \boldsymbol{\theta}_{21}). \tag{4.33}$$

Under the previous definitions, it is a matter of simple algebra (Problem 4.8) to prove that the set of equations in (4.25) is equivalent to

$$\boxed{\hat{\mathrm{y}} := \hat{\mathrm{y}}_r + j\hat{\mathrm{y}}_i = \boldsymbol{\theta}^{\mathrm{H}}\mathbf{x} + \boldsymbol{v}^{\mathrm{H}}\mathbf{x}^* : \quad \text{Widely Linear Complex Estimator.}} \tag{4.34}$$

To distinguish from (4.21), this is known as *widely linear* complex-valued estimator. Note that in (4.34), **x** as well as its complex conjugate, $\mathbf{x}^*$, are *simultaneously* used in order to cover all possible solutions, as those are dictated by the vector space description, which led to the formulation in (4.25).

Circularity conditions

We now turn our attention into investigating conditions under which the widely linear formulation in (4.34) breaks down to (4.21); that is, the conditions for which the optimal widely linear estimator turns out to have $\boldsymbol{v} = \mathbf{0}$.

Let

$$\boldsymbol{\varphi} := \begin{bmatrix} \boldsymbol{\theta} \\ \boldsymbol{v} \end{bmatrix} \quad \text{and} \quad \tilde{\mathbf{x}} := \begin{bmatrix} \mathbf{x} \\ \mathbf{x}^* \end{bmatrix}. \tag{4.35}$$

Then the widely linear estimator is written as

$$\hat{\mathrm{y}} = \boldsymbol{\varphi}^{\mathrm{H}}\tilde{\mathbf{x}}.$$

Adopting the orthogonality condition in its complex formulation

$$\mathbb{E}\left[\tilde{\mathbf{x}}\mathrm{e}^*\right] = \mathbb{E}\left[\tilde{\mathbf{x}}\left(\mathrm{y}-\hat{\mathrm{y}}\right)^*\right] = \mathbf{0},$$

we obtain the following set of normal equations for the optimal $\boldsymbol{\varphi}_*$,

$$\mathbb{E}\left[\tilde{\mathbf{x}}\tilde{\mathbf{x}}^{\mathrm{H}}\right]\boldsymbol{\varphi}_* = \mathbb{E}\left[\tilde{\mathbf{x}}\tilde{\mathbf{x}}^{\mathrm{H}}\right]\begin{bmatrix}\boldsymbol{\theta}_*\\ \boldsymbol{v}_*\end{bmatrix} = \begin{bmatrix}\mathbb{E}[\mathbf{x}\mathrm{y}^*]\\ \mathbb{E}[\mathbf{x}^*\mathrm{y}^*]\end{bmatrix},$$

or

$$\begin{bmatrix}\Sigma_x & P_x\\ P_x^* & \Sigma_x^*\end{bmatrix}\begin{bmatrix}\boldsymbol{\theta}_*\\ \boldsymbol{v}_*\end{bmatrix} = \begin{bmatrix}\boldsymbol{p}\\ \boldsymbol{q}^*\end{bmatrix}, \tag{4.36}$$

where Σ_x and $\boldsymbol{p}$ have been defined in (4.17) and (4.18), respectively, and

$$P_x := \mathbb{E}[\mathbf{x}\mathbf{x}^{\mathrm{T}}], \quad \boldsymbol{q} := \mathbb{E}[\mathbf{x}\mathrm{y}]. \tag{4.37}$$

The matrix P_x is known as the *pseudo covariance/autocorrelation* matrix of $\mathbf{x}$. Note that (4.36) is the equivalent of (4.28); to obtain the widely linear estimator, one needs to solve one set of complex-valued equations whose number is double compared to that of the linear (complex) formulation.

Assume now that

$$\boxed{P_x = O \text{ and } \boldsymbol{q} = \mathbf{0}: \quad \text{Circularity Conditions.}} \tag{4.38}$$

We say that in this case, the input-output variables are *jointly circular* and the input variables in $\mathbf{x}$ obey the (second order) *circular* condition. It is readily observed that, under the previous circularity assumptions, (4.36) leads to $\boldsymbol{v}_* = \mathbf{0}$ and the optimal $\boldsymbol{\theta}_*$ is given by the set of normal equations (4.16)-(4.18), which govern the more restricted linear case. Thus, adopting the linear formulation leads to optimality only under certain conditions, which do not always hold true in practice; a typical such example of variables, which do not respect circularity, are met in fMRI imaging (see [1] and the references therein). It can be shown that the MSE achieved by a widely linear estimator is always less than or equal to that obtained via a linear one (Problem 4.9).

The notions of circularity and of the widely linear estimation were treated in a series of fundamental papers [35, 36]. A stronger condition for circularity is based on the pdf of a complex random variable: A random variable x is circular (or strictly circular) if x and $\mathrm{x}e^{j\phi}$ are distributed according to the same pdf; that is, the pdf is *rotationally invariant* [35]. Figure 4.4a shows the scatter plot of points generated by a circularly distributed variable and Figure 4.4b corresponds to a noncircular one. Strict circularity implies the second-order circularity, but the converse is not always true. For more on complex random variables, the interested reader may consult [3, 37]. In Ref. [28], it is pointed out that the full second-order statistics of the error, without doubling the dimension, can be achieved if instead of the MSE one employs the Gaussian entropy criterion.

Finally, note that substituting in (4.29) the second-order circularity conditions, given in (4.38), one obtains (Problem 4.10),

$$\Sigma_r = \Sigma_i, \quad \Sigma_{ri} = -\Sigma_{ir}, \quad \mathbb{E}[\mathbf{x}_r\mathrm{y}_r] = \mathbb{E}[\mathbf{x}_i\mathrm{y}_i], \quad \mathbb{E}[\mathbf{x}_i\mathrm{y}_r] = -\,\mathbb{E}[\mathbf{x}_r\mathrm{y}_i], \tag{4.39}$$

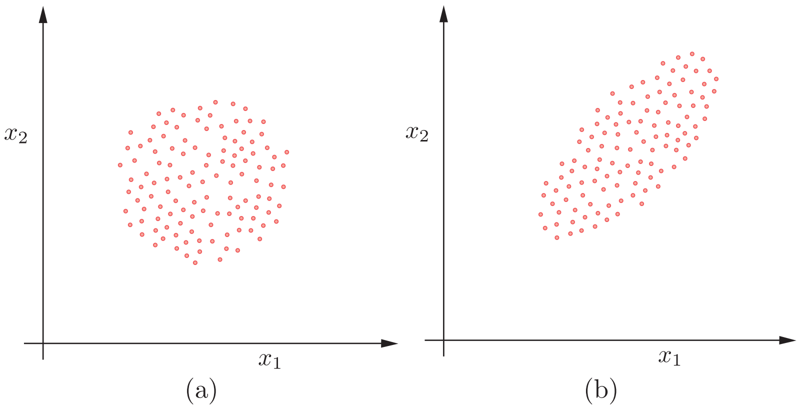


FIGURE 4.4

Scatter plots of points corresponding to (a) a circular process and (b) a noncircular one, in the two-dimensional space.

which then implies that $\boldsymbol{\theta}_{11} = \boldsymbol{\theta}_{22}$, and $\boldsymbol{\theta}_{12} = -\boldsymbol{\theta}_{21}$; in this case, (4.33) verifies that $\boldsymbol{v} = \mathbf{0}$, and that the optimal in the MSE sense solution has the special structure of (4.23) and (4.24).

4.4.2 OPTIMIZING WITH RESPECT TO COMPLEX-VALUED VARIABLES: WIRTINGER CALCULUS

So far, in order to derive the estimates of the parameters, for both the linear as well as the widely linear estimators, the orthogonality condition was mobilized. For the complex linear estimation case, the normal equations were derived in Problem 4.6, by direct minimization of the cost function in (4.20). Those who got involved with solving the problem have experienced a procedure that was more cumbersome compared to the real-valued linear estimation. This is because one has to use the real and imaginary parts of all the involved complex variables and express the cost function in terms of the equivalent real-valued quantities *only*; then the required gradients for the optimization have to be performed. Recall that any complex function $f : \mathbb{C} \rightarrow \mathbb{R}$ is not differentiable with respect to its complex argument, because the Cauchy-Riemann conditions are violated (Problem 4.11). The previously stated procedure of splitting the involved variables into their real and imaginary parts can become cumbersome with respect to algebraic manipulations. Wirtinger calculus provides an equivalent formulation that is based on simple rules and principles, which bear a great resemblance to the rules of standard complex differentiation.

Let $f : \mathbb{C} \longmapsto \mathbb{C}$ be a complex function defined on $\mathbb{C}$. Obviously, such a function can be regarded as either defined on $\mathbb{R}^2$ or $\mathbb{C}$ (i.e., $f(z) = f(x + jy) = f(x, y)$). Furthermore, it may be regarded as either complex-valued, $f(x, y) = f_r(x, y) + jf_i(x, y)$ or as vector-valued $f(x, y) = (f_r(x, y), f_i(x, y))$. We say that f is *differentiable in the real sense* if both f_r and f_i are differentiable. Wirtinger's calculus considers the complex structure of f and the real derivatives are described using an equivalent formulation that greatly simplifies calculations; moreover, this formulation bears a surprising similarity with the complex derivatives.

Definition 4.1. The *Wirtinger derivative* or *W-derivative* of a complex function f at a point $z_0 \in \mathbb{C}$ is defined as

$$\boxed{\frac{\partial f}{\partial z}(z_0) = \frac{1}{2}\left(\frac{\partial f_r}{\partial x}(z_0) + \frac{\partial f_i}{\partial y}(z_0)\right) + \frac{j}{2}\left(\frac{\partial f_i}{\partial x}(z_0) - \frac{\partial f_r}{\partial y}(z_0)\right) : \text{ W-derivative.}}$$

The *Conjugate Wirtinger's derivative* or *CW-derivative* of f at z_0 is defined as

$$\boxed{\frac{\partial f}{\partial z^*}(z_0) = \frac{1}{2}\left(\frac{\partial f_r}{\partial x}(z_0) - \frac{\partial f_i}{\partial y}(z_0)\right) + \frac{j}{2}\left(\frac{\partial f_i}{\partial x}(z_0) + \frac{\partial f_r}{\partial y}(z_0)\right) : \text{ CW-derivative.}}$$

For some of the properties and the related proofs regarding Wirtinger's derivatives see Appendix A.3. An important property for us is that if f is real-valued (i.e., $\mathbb{C} \longmapsto \mathbb{R}$) and z_0 is a (local) optimal point of f, it turns out that

$$\boxed{\frac{\partial f}{\partial z}(z_0) = \frac{\partial f}{\partial z^*}(z_0) = 0 : \quad \text{Optimality Conditions.}} \qquad (4.40)$$

In order to apply Wirtinger's derivatives, the following simple *tricks* are adopted:

- express function f in terms of z *and* z^*;
- to compute W-derivative apply the usual differentiation rule, treating z^* as a constant;
- to compute CW-derivative apply the usual differentiation rule, treating z as a constant.

It should be emphasized that all these statements must be regarded as useful computational tricks rather than rigorous mathematical rules. Analogous definitions and properties carry on for complex vectors $\boldsymbol{z}$, and the W-gradient and CW-gradients

$$\nabla_{\boldsymbol{z}} f(\boldsymbol{z}_0), \quad \nabla_{\boldsymbol{z}^*} f(\boldsymbol{z}_0),$$

result from the respective definitions if partial derivatives are replaced by partial gradients, $\nabla_{\boldsymbol{x}}, \nabla_{\boldsymbol{y}}$.

Although Wirtinger's calculus has been known since 1927 [51], its use in applications has a rather recent history [7] and its revival was ignited by the widely linear filtering concept [27]. The interested reader may obtain more on this issue from [2, 25, 30]. Extensions of Wirtinger's derivative to general Hilbert (infinite dimensional) spaces was done more recently in [6] and to the subgradient notion in [46].

Application in Linear Estimation. The cost function in this case is

$$J(\boldsymbol{\theta}, \boldsymbol{\theta}^*) = \mathbb{E}\left[|\mathrm{y} - \boldsymbol{\theta}^{\mathrm{H}}\mathbf{x}|^2\right] = \mathbb{E}\left[\left(\mathrm{y} - \boldsymbol{\theta}^{\mathrm{H}}\mathbf{x}\right)\left(\mathrm{y}^* - \boldsymbol{\theta}^{\mathrm{T}}\mathbf{x}^*\right)\right].$$

Thus, treating $\boldsymbol{\theta}$ as a constant, the optimal occurs at

$$\nabla_{\boldsymbol{\theta}^*} J = \mathbb{E}[\mathbf{x}\mathrm{e}^*] = \mathbf{0},$$

which is the orthogonality condition leading to the normal equations (4.16)-(4.18).

Application in Widely Linear Estimation. The cost function is now (see notation in (4.35))

$$J(\boldsymbol{\varphi}, \boldsymbol{\varphi}^*) = \mathbb{E}\left[\left(\mathrm{y} - \boldsymbol{\varphi}^{\mathrm{H}}\tilde{\mathbf{x}}\right)\left(\mathrm{y}^* - \boldsymbol{\varphi}^{\mathrm{T}}\tilde{\boldsymbol{x}}^*\right)\right],$$

and treating $\boldsymbol{\varphi}$ as a constant,

$$\nabla_{\boldsymbol{\varphi}^*} J = \mathbb{E}[\tilde{\mathbf{x}}\mathrm{e}^*] = \mathbb{E}\begin{bmatrix} \mathbf{x}\mathrm{e}^* \\ \mathbf{x}^*\mathrm{e}^* \end{bmatrix} = \mathbf{0},$$

which leads to the set derived in (4.36).

Wirtinger's calculus will prove very useful in subsequent chapters for deriving gradient operations in the context of online/adaptive estimation in Euclidean as well as in reproducing kernel Hilbert spaces.

4.5 LINEAR FILTERING

Linear statistical filtering is an instance of the general estimation task, when the notion of time evolution needs to be taken into consideration and estimates are obtained at each time instant. There are three major types of problems that emerge:

- *Filtering*, where the estimate at time instant n is based on all previously received (measured) input information *up to and including* the current time index, n.
- *Smoothing*, where data over a time interval, $[0, N]$, are first collected and an estimate is obtained at each time instant $n \leq N$, using *all* the available information in the interval $[0, N]$.
- *Prediction*, where estimates at times $n + \tau,\ \tau > 0$ are to be obtained based on the information up to and including time instant n.

To fit in the above definitions more with what has been said so far in the chapter, take for example a time-varying case, where the output variable at time instant n is y_n and its value depends on observations included in the corresponding input vector $\mathbf{x}_n$. In filtering, the latter can include measurements received only at time instants $n, n-1, \ldots, 0$. This restriction in the index set is directly related to *causality*. In contrast, in smoothing, we can also include future time instants $n+2, n+1, n, n-1$.

Most of the effort in this book will be spent on filtering whenever time information enters into the picture. The reason is that this is the most commonly encountered task and, also, the techniques used for smoothing and prediction are similar in nature with that of filtering, with usually minor modifications.

In signal processing, the term filtering is usually used in a more specific context, and it refers to the operation of a *filter*, which acts on an input random process/signal (u_n), to transform it into another one (d_n), see Section 2.4.3. Note that we have switched into the notation, introduced in Chapter 2, used to denote random processes. We prefer to keep different notation for processes and random variables, because in the case of random processes, the filtering task obtains a special structure and properties, as we will soon see. Moreover, although the mathematical formulation of the involved equations, for both cases, may end up to be the same, we feel that it is good for the reader to keep in mind that there is a different underlying mechanism for generating the data.

The task in statistical linear filtering is to compute the coefficients (impulse response) of the filter so that the output process of the filter, $\hat{\mathrm{d}}_n$, when the filter is excited by the input random process, u_n, to be as close as possible to a *desired* response process, d_n. In other words, the goal is to minimize, in some sense, the corresponding error processes, see Figure 4.5. Assuming that the unknown filter is of

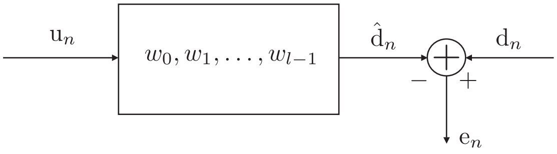


FIGURE 4.5

In statistical filtering, the impulse response coefficients are estimated so as to minimize the error between the output and the desired response processes. In MSE linear filtering, the cost function is $\mathbb{E}[\mathrm{e}_n^2]$.

a finite impulse response (FIR) (see Section 2.4.3 for related definitions), denoted as $w_0, w_1, \ldots, w_{l-1}$, the output $\hat{\mathrm{d}}_n$ of the filter is given as

$$\boxed{\hat{\mathrm{d}}_n = \sum_{i=0}^{l-1} w_i \mathrm{u}_{n-i} = \boldsymbol{w}^T \mathbf{u}_n : \quad \text{Convolution Sum,}} \tag{4.41}$$

where

$$\boldsymbol{w} = [w_0, w_1, \ldots, w_{l-1}]^T, \quad \text{and} \quad \mathbf{u}_n = [\mathrm{u}_n, \mathrm{u}_{n-1}, \ldots, \mathrm{u}_{n-l+1}]^T. \tag{4.42}$$

Figure 4.6 illustrates the convolution operation of the linear filter, when the input is excited by a realization u_n of the input processes to provide in the output the signal/sequence $\hat{d}_n$.

Alternatively, (4.41) can be viewed as the *linear* estimator function; given the jointly distributed variables, at time instant n, $(\mathrm{d}_n, \mathbf{u}_n)$, (4.41) provides the estimator, $\hat{\mathrm{d}}_n$, given the values of $\mathbf{u}_n$. In order to obtain the coefficients, $\boldsymbol{w}$, the mean-square error criterion will be adopted. Furthermore, we will assume that

- The processes, u_n, d_n are *wide-sense stationary* real random processes.
- Their mean values are equal to zero, in other words, $\mathbb{E}[\mathrm{u}_n] = \mathbb{E}[\mathrm{d}_n] = 0, \ \forall n$. If this is not the case, we can subtract the respective mean values from the processes, u_n and d_n, during a preprocessing stage. Due to this assumption, the autocorrelation and covariance matrices of $\mathbf{u}_n$ coincide, so that

$$R_u = \Sigma_u.$$

The normal equations in (4.5) now take the form

$$\Sigma_u \boldsymbol{w} = \boldsymbol{p},$$

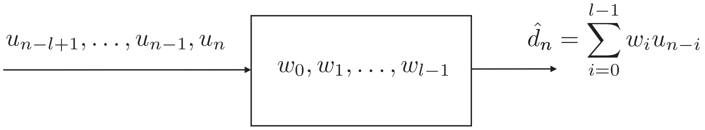

$u_{n-l+1}, \ldots, u_{n-1}, u_n$
$w_0, w_1, \ldots, w_{l-1}$
$\hat{d}_n = \sum_{i=0}^{l-1} w_i u_{n-i}$

FIGURE 4.6

The linear filter is excited by a realization of an input process. The output signal is the convolution between the input sequence and the filter's impulse response.

where

$$\boldsymbol{p} = \left[\mathbb{E}[\mathrm{u}_n \mathrm{d}_n], \ldots, \mathbb{E}[\mathrm{u}_{n-l+1} \mathrm{d}_n]\right]^{\mathrm{T}},$$

and the respective covariance/autocorrelation matrix, of order l, of the input process is given by

$$\Sigma_u := \mathbb{E}[\mathbf{u}_n \mathbf{u}_n^{\mathrm{T}}] = \begin{bmatrix} r(0) & r(1) & \ldots & r(l-1) \\ r(1) & r(0) & \ldots & r(l-2) \\ \vdots & & \ddots & \\ r(l-1) & r(l-2) & \ldots & r(0) \end{bmatrix}, \tag{4.43}$$

where $r(k)$ is the autocorrelation sequence of the input process. Because we have assumed that the involved processes are wide-sense stationary, we have that

$$r(n, n-k) := \mathbb{E}[\mathrm{u}_n \mathrm{u}_{n-k}] = r(k).$$

Also, recall that, for real wide-sense stationary processes, the autocorrelation sequence is symmetric, or $r(k) = r(-k)$ (Section 2.4.3). Observe that in this case, where the input vector results from a random process, the covariance matrix has a special structure, which will be exploited later on to derive efficient schemes for the solution of the normal equations.

For the complex linear filtering case, the only differences are

- The output is given as, $\hat{\mathrm{d}}_n = \boldsymbol{w}^{\mathrm{H}} \mathbf{u}_n$,
- $\boldsymbol{p} = \mathbb{E}[\mathbf{u}_n \mathrm{d}_n^*]$,
- $\Sigma_u = \mathbb{E}[\mathbf{u}_n \mathbf{u}_n^{\mathrm{H}}]$,
- $r(-k) = r^*(k)$.

4.6 MSE LINEAR FILTERING: A FREQUENCY DOMAIN POINT OF VIEW

Let us now turn our attention to the more general case, and assume that our filter is of *infinite impulse response* (IIR). Then, (4.41) now becomes

$$\hat{\mathrm{d}}_n = \sum_{i=-\infty}^{+\infty} w_i \mathrm{u}_{n-i}. \tag{4.44}$$

Moreover, we have allowed the filter to be *noncausal*.[4] Following similar arguments as those used to prove the MSE optimality of $\mathbb{E}[y|\boldsymbol{x}]$ in Section 3.15, it turns out that the optimal filter coefficients must satisfy the following condition, (Problem 4.12),

$$\boxed{\mathbb{E}\left[(\mathrm{d}_n - \sum_{i=-\infty}^{+\infty} w_i \mathrm{u}_{n-i}) \mathrm{u}_{n-j}\right] = 0, \quad j \in \mathbb{Z}.} \tag{4.45}$$

[4] A system is called *causal* if the output, $\hat{d}_n$, does depend *only* on input values u_m, $m \leq n$. A necessary and sufficient condition for causality is that the impulse response is zero for negative time instants, meaning that $w_n = 0$, $n < 0$. This can easily be checked out; try it.

Observe that this is a generalization (involving an infinite number of terms) of the orthogonality condition stated in (4.10). A rearrangement of the terms in (4.45) results in

$$\sum_{i=-\infty}^{+\infty} w_i \, \mathbb{E}[\mathrm{u}_{n-i}\mathrm{u}_{n-j}] = \mathbb{E}[\mathrm{d}_n \mathrm{u}_{n-j}], \ j \in \mathbb{Z}, \tag{4.46}$$

and finally to

$$\sum_{i=-\infty}^{+\infty} w_i r(j-i) = r_{du}(j), \ j \in \mathbb{Z}. \tag{4.47}$$

Equation (4.47) can be considered as the generalization of (4.5) to the case of random processes. The problem now is how one can solve (4.47). The way out is to cross into the frequency domain. Equation (4.47) can be seen as the convolution of the unknown sequence with the autocorrelation sequence of the input process, which gives rise to the cross-correlation sequence. However, we know that convolution of two sequences corresponds to the product of the respective Fourier transforms (e.g., [42]). Thus, we can now write that

$$\boxed{W(\omega) S_u(\omega) = S_{du}(\omega),} \tag{4.48}$$

where $W(\omega)$ is the Fourier transform of the sequence of the unknown parameters, and $S_u(\omega)$ is the *power spectral density* of the input process, defined in Section 2.4.3. In analogy, the Fourier transform $S_{du}(\omega)$ of the cross-correlation sequence is known as the *cross-spectral density*. If the latter two quantities are available, then once $W(\omega)$ has been computed, the unknown parameters can be obtained via the inverse Fourier transform.

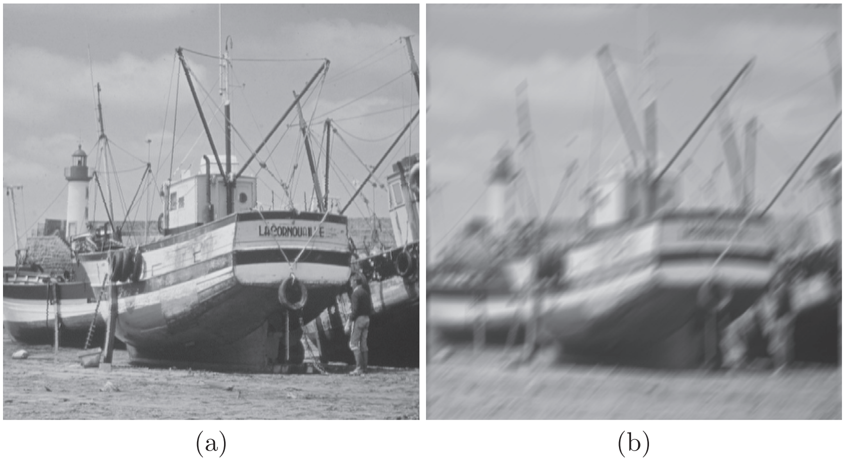
Deconvolution: image deblurring

We will now consider an important application in order to demonstrate the power of MSE linear estimation. Image deblurring is a typical *deconvolution* task. An image is degraded due to its transmission via a nonideal system; the task of deconvolution is to optimally recover (in the MSE sense in our case), the original undegraded image. Figure 4.7a shows the original image and 4.7b a blurred version (e.g., taken by a nonsteady camera) with some small additive noise.

At this point, it is interesting to recall that deconvolution is a process that our human brain performs all the time. The human (and not only) vision system is one of the most complex and highly developed biological systems that has been formed over millions years of a continuous evolution process. Any raw image that falls on the retina of the eye is *severely blurred*. Thus, one of the main early processing activities of our visual system is to deblur it (see, e.g., [29] and the references therein for a related discussion).

Before we proceed any further, the following assumptions are adopted:

- The image is a *wide-sense stationary* two-dimensional random process. Two-dimensional random processes are also known as *random fields*, see Chapter 15.
- The image is of an infinite extent; this can be justified for the case of large images. This assumption will grant us the "permission" to use (4.48). The fact that an image is a two-dimensional process does not change anything in the theoretical analysis; the only difference is that now the Fourier transforms involve two frequency variables, ω_1, ω_2, one for each of the two dimensions.



(a) (b)

FIGURE 4.7

(a) The original image and (b) its blurred and noisy version.

A gray image is represented as a two-dimensional array. To stay close to the notation used so far, let $d(n,m),\ n,m \in \mathbb{Z}$ be the original undegraded image (which for us is now the desired response), and $u(n,m),\ n,m \in \mathbb{Z}$ be the degraded one, obtained as

$$u(n,m) = \sum_{i=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} h(i,j)d(n-i,m-j) + \eta(n,m), \tag{4.49}$$

where $\eta(n,m)$ is the realization of a noise field, which is assumed to be zero mean and independent of the input (undegraded) image. The sequence $h(i,j)$ is the *point spread sequence* (impulse response) of the system (e.g., camera). We will assume that this is known and it has, somehow, been measured.[5]

Our task now is to estimate a two-dimensional filter, $w(n,m)$, which is applied to the degraded image to optimally reconstruct (in the MSE sense) the original undegraded image. In the current context, Eq. (4.48) is written as

$$W(\omega_1,\omega_2)S_u(\omega_1,\omega_2) = S_{du}(\omega_1,\omega_2).$$

Following similar arguments as those used to derive Eq. (2.130) of Chapter 2, it is shown that (Problem 4.13)

$$S_{du}(\omega_1,\omega_2) = H^*(\omega_1,\omega_2)S_d(\omega_1,\omega_2), \tag{4.50}$$

and

$$S_u(\omega_1,\omega_2) = |H(\omega_1,\omega_2)|^2 S_d(\omega_1,\omega_2) + S_\eta(\omega_1,\omega_2), \tag{4.51}$$

[5] Note that this is not always the case.

where "*" denotes complex conjugation and S_η is the power spectral density of the noise field. Thus, we finally obtain that

$$\boxed{W(\omega_1,\omega_2) = \frac{1}{H(\omega_1,\omega_2)} \frac{|H(\omega_1,\omega_2)|^2}{|H(\omega_1,\omega_2)|^2 + \frac{S_\eta(\omega_1,\omega_2)}{S_d(\omega_1,\omega_2)}}.} \tag{4.52}$$

Once $W(\omega_1,\omega_2)$ has been computed, the unknown parameters could be obtained via an inverse (two-dimensional) Fourier transform. The deblurred image then results as

$$\hat{d}(n,m) = \sum_{i=-\infty}^{+\infty} \sum_{j=-\infty}^{+\infty} w(i,j)u(n-i,m-j). \tag{4.53}$$

In practice, because we are not really interested in obtaining the weights of the deconvolution filter, we implement (4.53) in the frequency domain

$$\hat{D}(\omega_1,\omega_2) = W(\omega_1,\omega_2)U(\omega_1,\omega_2),$$

and then obtain the inverse Fourier transform. Thus, all the processing is efficiently performed in the frequency domain. Software packages to perform Fourier transforms (via the Fast Fourier Transform, FFT) of an image array are "omnipresent" on the internet.

Another important issue is that in practice we do not know $S_d(\omega_1,\omega_2)$. An approximation, which is usually adopted that renders sensible results, is to assume that $\frac{S_\eta(\omega_1,\omega_2)}{S_d(\omega_1,\omega_2)}$ is a constant, C, and try different values of it. Figure 4.8 shows the deblurred image for $C = 2.3 \times 10^{-6}$. The quality of the end result depends a lot on the choice of this value (MATLAB exercise 4.25). Other, more advanced,

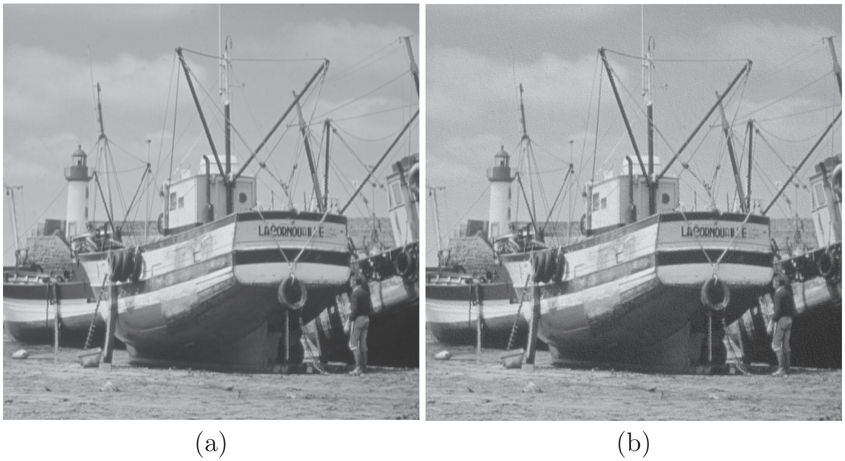


FIGURE 4.8

(a) The original image and (b) the deblurred one for $C = 2.3 \times 10^{-6}$. Observe that in spite of the simplicity of the method, the reconstruction is pretty good. The differences become more obvious to the eye when the images are enlarged.

techniques, have also been proposed. For example, one can get a better estimate of $S_d(\omega_1, \omega_2)$ by using information from $S_\eta(\omega_1, \omega_2)$ and $S_u(\omega_1, \omega_2)$. The interested reader can obtain more on the image deconvolution/restoration task from Refs. [14, 34].

4.7 SOME TYPICAL APPLICATIONS

Optimal linear estimation/filtering has been applied in a wide range of diverse applications of statistical learning, such as regression modeling, communications, control, biomedical signal processing, seismic signal processing, image processing. In the sequel, we present some typical applications in order for the reader to grasp the main rationale of how the previously stated theory can find its way in solving practical problems. In all cases, wide-sense stationarity of the involved random processes is assumed.

4.7.1 INTERFERENCE CANCELLATION

In interference cancellation, we have access to a mixture of two signals expressed as $d_n = y_n + s_n$. Ideally, we would like to remove one of them, say y_n. We will consider them as realizations of respective random processes/signals, or d_n, y_n and s_n. To achieve this goal, the only available information is another signal, say u_n, that is statistically related to the unwanted signal, y_n. For example, y_n may be a filtered version of u_n. This is illustrated in Figure 4.9, where the corresponding realizations of the involved random processes are shown.

Process y_n is the output of an unknown system H, whose input is excited by u_n. The task is to model H by obtaining estimates of its impulse response (assuming that it is LTI and of known order). Then, the output of the model will be an approximation of y_n, when this is activated by the same input, u_n. We will use d_n as the desired response process. The optimal estimates of $w_0, \ldots, w_{l-1}$ (assuming the order of the unknown system H to be l) are provided by the normal equations

$$\Sigma_u \boldsymbol{w}_* = \boldsymbol{p}.$$

However,

$$\begin{aligned}\boldsymbol{p} &= \mathbb{E}\left[\mathbf{u}_n \mathrm{d}_n\right] = \mathbb{E}\left[\mathbf{u}_n\left(\mathrm{y}_n + \mathrm{s}_n\right)\right] \\ &= \mathbb{E}\left[\mathbf{u}_n \mathrm{y}_n\right],\end{aligned} \tag{4.54}$$

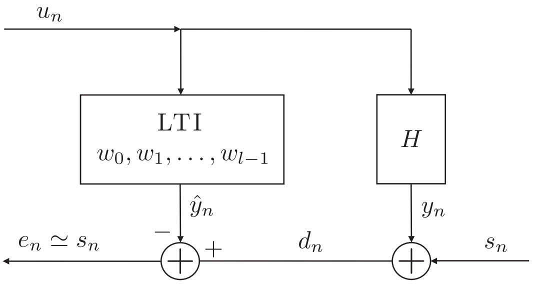


FIGURE 4.9

A basic block diagram illustrating the interference cancellation task.

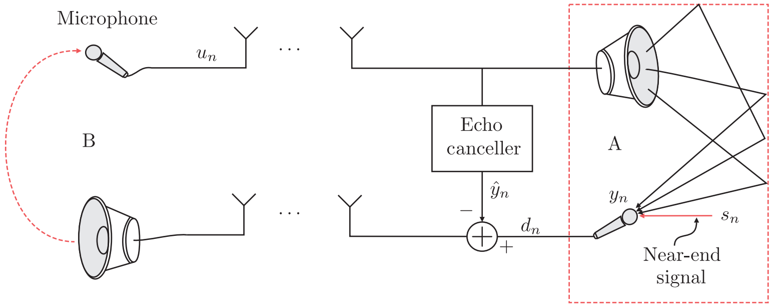


FIGURE 4.10

The echo canceller is optimally designed to remove the part of the far-end signal, u_n, that interferes with the near-end signal, s_n.

because the respective input vector $\mathbf{u}_n$ and s_n are considered statistically independent. That is, the previous formulation of the problem leads to the same normal equations as if the desired response was the signal y_n, which we want to remove! Hence, the output of our model will be an approximation (in the MSE sense), $\hat{\mathrm{y}}_n$, of y_n, and if subtracted from d_n the resulting (error) signal, e_n, will be an approximation to s_n. How good this approximation is depends on whether l is a good "estimate" of the true order of H. The cross-correlation in the right-hand side of (4.54) can be approximated by computing the respective sample mean values, in particular over periods where s_n is absent. In practical systems, online/adaptive versions of this implementation are usually employed, as we will see Chapter 5.

Interference cancellation schemes have been widely used in many systems such as noise cancellation, echo cancellation in telephone networks and video conferencing, and in biomedical applications; for example, in order to cancel the maternal interference in a fetal electrocardiograph.

Figure 4.10 illustrates the echo cancellation task in a video conference application. The same set up applies to the hands-free telephone service in a car. The *far-end* speech signal is considered to be a realization u_n of a random process, u_n; through the loudspeakers, it is broadcasted in room A (car) and it is reflected in the interior of the room. Part of it is absorbed and part of it enters the microphone; this is denoted as y_n. The equivalent response of the room (reflections) on u_n can be represented by a filter, H, as in Figure 4.9. Signal y_n returns back and the speaker in location B listens to her or his own voice, together with the *near-end* speech signal, s_n of the speaker in A. In certain cases, this feedback path from the loudspeakers to the microphone can cause instabilities giving rise to a "howling" sound effect. The goal of the echo canceller is to optimally remove y_n.

4.7.2 SYSTEM IDENTIFICATION

System identification is similar in nature to the interference cancellation task. Note that in Figure 4.9, one basically models the unknown system. However, the focus there was on replicating the output y_n and not on the system's impulse response.

In system identification, the aim is to model the impulse response of an unknown plant. To this end, we have access to its input signal as well as to a *noisy* version of its output. The task is to design a model whose impulse response approximates that of the unknown plant. To achieve this, we optimally design

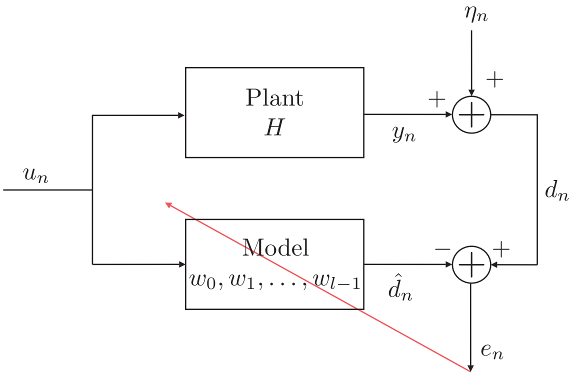


FIGURE 4.11

In system identification, the impulse response of the model is optimally estimated so that the output is close, in the MSE, to that of the unknown plant. The red line indicates that the error is used for the optimal estimation of the unknown parameters of the filter.

a linear filter whose input is the same signal as the one that activates the plant and its desired response is the noisy output of the plant, see Figure 4.11. The associated normal equations are

$$\Sigma_u \boldsymbol{w}_* = \mathbb{E}[\mathbf{u}_n \mathrm{d}_n] = \mathbb{E}[\mathbf{u}_n \mathrm{y}_n] + 0,$$

assuming the noise η_n is statistically independent of $\mathbf{u}_n$. Thus, once more, the resulting normal equations are the same as if we had provided the model with a desired response equal to the noiseless output of the unknown plant, expressed as $\mathrm{d}_n = \mathrm{y}_n$. Hence, the impulse response of the model is estimated so that its output is close, in the MSE, to the true (noiseless) output of the unknown plant. System identification is of major importance in a number of applications. In control, it is used for driving the associated controllers. In data communications, for estimating the transmission channel in order to build up maximum likelihood estimators of the transmitted data. In many practical systems, adaptive versions of the system identification scheme are implemented, as we will discuss in following chapters.

4.7.3 DECONVOLUTION: CHANNEL EQUALIZATION

Note that in the cancellation task the goal was to "remove" the (filtered version) of the input signal (u_n) to the unknown system H. In system identification, the focus was on the (unknown) system itself. In *deconvolution*, the emphasis is on the input of the unknown system. That is, our goal now is to recover, in the MSE optimal sense, the (delayed) input signal, u_{n-L}, where L is the delay in units of the sampling period, T. The task is also called *inverse system identification*. The term *equalization* or *channel equalization* is used in communications. The deconvolution task was introduced in the context of image deblurring in Section 4.6. There, the required information about the *unknown* input process was obtained via an approximation. In the current framework, this can be approached via the transmission of a training sequence.

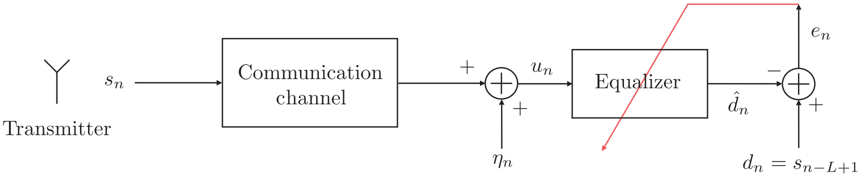


FIGURE 4.12

The task of an equalizer is to optimally recover the originally transmitted information sequence, s_n, delayed by L time lags.

The goal of an *equalizer* is to recover the transmitted information symbols, by mitigating the so-called *intersymbol interference* (ISI) that any (imperfect) dispersive communication channel imposes on the transmitted signal; besides ISI, additive noise is also present in the transmitted information bits (see Example 4.2). Equalizers are "omnipresent" in these days; in our mobile phones, in our modems, etc. Figure 4.12 presents the basic scheme for an equalizer. The equalizer is trained so that its output is as close as possible to the transmitted data bits delayed by some time lag L; the delay is used in order to account for the overall delayed imposed by the channel-equalizer system. Deconvolution/channel equalization is at the heart of a number of applications besides communications, such as acoustics, optics, seismic signal processing, and control. The channel equalization task will also be discussed in the next chapter in the context of online learning via the decision feedback equalization mode of operation.

Example 4.1 (Noise Cancellation)**.** The noise cancellation application is illustrated in Figure 4.13. The signal of interest is a realization of a process, s_n, that is contaminated by the noise sequence $\mathrm{v}_1(n)$. For example, s_n may be the speech signal of the pilot in the cockpit and $\mathrm{v}_1(n)$ the aircraft noise at the location of the microphone. We assume that $\mathrm{v}_1(n)$ is an AR process of order one, expressed as

$$\mathrm{v}_1(n) = a_1 \mathrm{v}_1(n-1) + \eta_n.$$

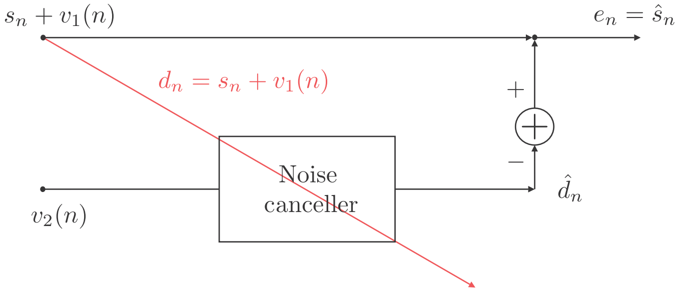


FIGURE 4.13

A block diagram for a noise canceller. Using as desired response the contaminated signal, the output of the optimal filter is an estimate of the noise component.

The signal $\mathrm{v}_2(n)$ is a noise sequence,[6] which is related to $\mathrm{v}_1(n)$, but it is statistically independent of s_n. For example, it may be the noise picked up from another microphone positioned at a nearby location. This is also assumed to be an AR process of the first order,

$$\mathrm{v}_2(n) = a_2\mathrm{v}_2(n-1) + \eta_n.$$

Note that both $\mathrm{v}_1(n)$ and $\mathrm{v}_2(n)$ are generated by the same noise source, η_n, that is assumed to be white of variance σ_η^2. For example, in an aircraft it can be assumed that the noise at different points is due to a "common" source, especially for nearby locations.

The goal of the example is to compute estimates of the weights of the noise canceller, in order to optimally remove (in the MSE sense) the noise $\mathrm{v}_1(n)$ from the mixture $\mathrm{s}_n + \mathrm{v}_1(n)$. Assume the canceller to be of order two.

The input to the canceller is $\mathrm{v}_2(n)$ and as desired response the mixture signal, $\mathrm{d}_n = \mathrm{s}_n + \mathrm{v}_1(n)$, will be used. To establish the normal equations, we need to compute the covariance matrix, Σ_2, of $\mathrm{v}_2(n)$ and the cross-correlation vector, $\boldsymbol{p}_2$, between the input random vector, $\mathbf{v}_2(n)$, and d_n.

Because $\mathrm{v}_2(n)$ is an AR process of the first order, recall from Section 2.4.4 that, the autocorrelation sequence is given by

$$r_2(k) = \frac{a_2^k\sigma_\eta^2}{1-a_2^2}, \quad k = 0, 1, \ldots \tag{4.55}$$

Hence,

$$\Sigma_2 = \begin{bmatrix} r_2(0) & r_2(1) \\ r_2(1) & r_2(0) \end{bmatrix} = \begin{bmatrix} \dfrac{\sigma_\eta^2}{1-a_2^2} & \dfrac{a_2\sigma_\eta^2}{1-a_2^2} \\ \dfrac{a_2\sigma_\eta^2}{1-a_2^2} & \dfrac{\sigma_\eta^2}{1-a_2^2} \end{bmatrix}.$$

Next, we are going to compute the cross-correlation vector.

$$\begin{aligned} p_2(0) :&= \mathbb{E}[\mathrm{v}_2(n)\mathrm{d}_n] = \mathbb{E}\left[\mathrm{v}_2(n)\left(\mathrm{s}_n + \mathrm{v}_1(n)\right)\right] \\ &= \mathbb{E}[\mathrm{v}_2(n)\mathrm{v}_1(n)] + 0 = \mathbb{E}\left[\left(a_2\mathrm{v}_2(n-1) + \eta_n\right)\left(a_1\mathrm{v}_1(n-1) + \eta_n\right)\right] \\ &= a_2a_1p_2(0) + \sigma_\eta^2, \end{aligned}$$

or

$$p_2(0) = \frac{\sigma_\eta^2}{1-a_2a_1}. \tag{4.56}$$

We used the fact that $\mathbb{E}[\mathrm{v}_2(n-1)\eta_n] = \mathbb{E}[\mathrm{v}_1(n-1)\eta_n] = 0$, because $\mathrm{v}_2(n-1)$ and $\mathrm{v}_2(n-1)$ depend recursively on previous values, i.e., $\eta(n-1), \eta(n-2), \ldots$, and also η_n is a white noise sequence, hence the respective correlation values are zero. Also, due to stationarity, $\mathbb{E}[\mathrm{v}_2(n)\mathrm{v}_1(n)] = \mathbb{E}[\mathrm{v}_2(n-1)\mathrm{v}_1\ (n-1)]$.

[6] We use the index n in parenthesis to unclutter notation due to the presence of a second subscript.

For the other value of the cross-correlation vector we have

$$\begin{aligned} p_2(1) &= \mathbb{E}[\mathrm{v}_2(n-1)\mathrm{d}_n] = \mathbb{E}\left[\mathrm{v}_2(n-1)\left(\mathrm{v}_1(n)+\eta_n\right)\right] \\ &= \mathbb{E}[\mathrm{v}_2(n-1)\mathrm{v}_1(n)] + 0 = \mathbb{E}\left[\mathrm{v}_2(n-1)\left(a_1\mathrm{v}_1(n-1)+\eta_n\right)\right] \\ &= a_1 p_2(0) = \frac{a_1\sigma_\eta^2}{1-a_1a_2}. \end{aligned}$$

In general, it is easy to show that

$$p_2(k) = \frac{a_1^k\sigma_\eta^2}{1-a_2a_1}, \quad k = 0, 1, \ldots. \tag{4.57}$$

Recall that because the processes are real-valued, the covariance matrix is symmetric, meaning $r_2(k) = r_2(-k)$. Also, for (4.55) to make sense, $(r_2(0) > 0)$, $|a_2| < 1$. The same holds true for $|a_1|$, following similar arguments for the autocorrelation process of $\mathrm{v}_1(n)$.

Thus, the optimal weights of the noise canceller are given by the following set of normal equations,

$$\begin{bmatrix} \frac{\sigma_\eta^2}{1-a_2^2} & \frac{a_2\sigma_\eta^2}{1-a_2^2} \\ \frac{a_2\sigma_\eta^2}{1-a_2^2} & \frac{\sigma_\eta^2}{1-a_2^2} \end{bmatrix} \boldsymbol{w} = \begin{bmatrix} \frac{\sigma_\eta^2}{1-a_1a_2} \\ \frac{a_1\sigma_\eta^2}{1-a_1a_2} \end{bmatrix}.$$

Note that the canceller optimally "removes" from the mixture, $\mathrm{s}_n + \mathrm{v}_1(n)$, the component that is correlated to the input, $\mathrm{v}_2(n)$; observe that $\mathrm{v}_1(n)$ basically acts as the desired response.

Figure 4.14a shows a realization of the signal $d_n = s_n + v_1(n)$, where $s_n = \cos(\omega_0 n)$ with $\omega_0 = 2 * 10^{-3} * \pi.$, $a_1 = 0.8$, and $\sigma_\eta^2 = 0.05$. Figure 4.14b is the respective realization of the signal $\mathrm{s}_n + \mathrm{v}_1(n) - \hat{\mathrm{d}}(n)$ for $a_2 = 0.75$. The corresponding weights for the canceller are $\boldsymbol{w}_* = [1, 0.125]^\mathrm{T}$. Figure 4.14c corresponds to $a_2 = 0.5$. Observe that the higher the cross-correlation between $\mathrm{v}_1(n)$ and $\mathrm{v}_2(n)$ the better the obtained result becomes.

Example 4.2 (Channel Equalization)**.** Consider the channel equalization set up in Figure 4.12, where the output of the channel, which is sensed by the receiver, is given by

$$\mathrm{u}_n = 0.5\mathrm{s}_n + \mathrm{s}_{n-1} + \eta_n. \tag{4.58}$$

The goal is to design an equalizer comprising three taps, $\boldsymbol{w} = [w_0, w_1, w_2]^\mathrm{T}$, so that

$$\hat{\mathrm{d}}_n = \boldsymbol{w}^\mathrm{T}\mathbf{u}_n,$$

and estimate the unknown taps using as a desired response sequence $\mathrm{d}_n = \mathrm{s}_{n-1}$. We are given that $\mathbb{E}[s_n] = \mathbb{E}[\eta_n] = 0$ and

$$\Sigma_s = \sigma_s^2 I, \quad \Sigma_\eta = \sigma_\eta^2 I.$$

Note that for the desired response we have used a delay $L = 1$. In order to better understand the reason that a delay is used and without going into many details (for the more experienced reader, note that the channel is nonminimum phase, e.g., [41]) observe that at time n, most of the contribution to u_n in (4.58) comes from the symbol s_{n-1}, which is weighted by one, while the sample s_n is weighted

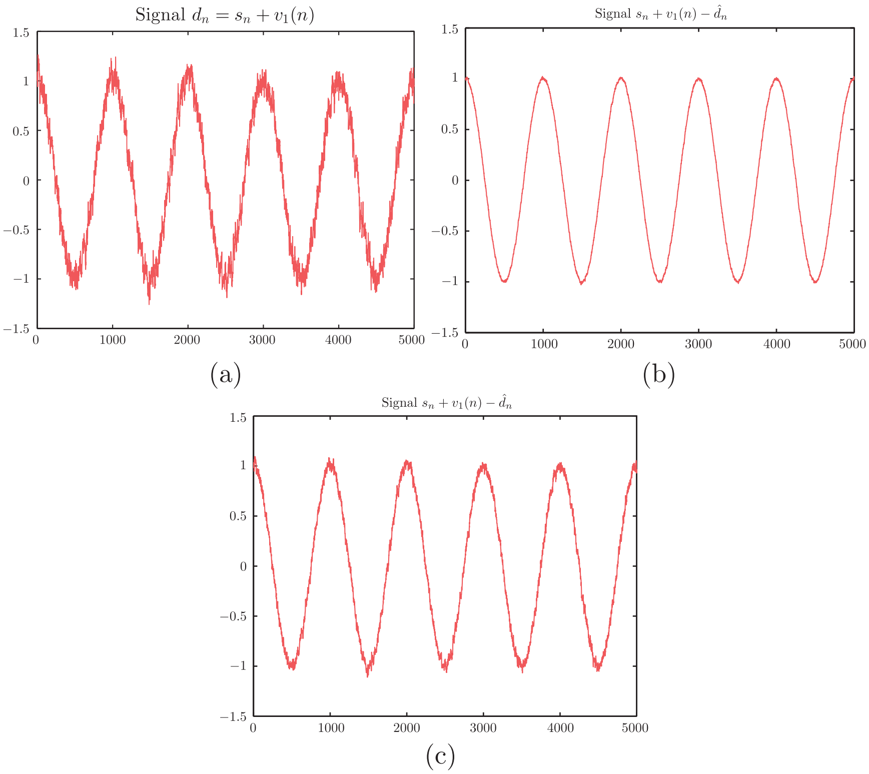


FIGURE 4.14

(a) The noisy sinusoid signal of Example 4.1. (b) The de-noised signal for strongly correlated noise sources, v_1 and v_2. (c) The obtained de-noised signal for less correlated noise sources.

by 0.5; hence, it is most natural from an intuitive point of view, at time n, having received u_n, to try to obtain an estimate for s_{n-1}. This justifies the use of the delay.

Figure 4.15a shows a realization of the input information sequence s_n. It consists of equiprobable ±1 samples, randomly generated. The effect of the channel is (a) to combine successive information samples together (ISI) and (b) to add noise; the purpose of the equalizer is to optimally remove both of them. Figure 4.15b shows the respective realization sequence of u_n, which is received at the receiver's front end. Observe that, by looking at it, one cannot recognize in it the original sequence; the noise together with the ISI have really changed its "look."

Following a similar procedure as in the previous example, we obtain (Problem 4.14)

$$\Sigma_u = \begin{bmatrix} 1.25\sigma_s^2 + \sigma_\eta^2 & 0.5\sigma_s^2 & 0 \\ 0.5\sigma_s^2 & 1.25\sigma_s^2 + \sigma_\eta^2 & 0.5\sigma_s^2 \\ 0 & 0.5\sigma_s^2 & 1.25\sigma_s^2 + \sigma_\eta^2 \end{bmatrix}, \quad \boldsymbol{p} = \begin{bmatrix} \sigma_s^2 \\ 0.5\sigma_s^2 \\ 0 \end{bmatrix}.$$

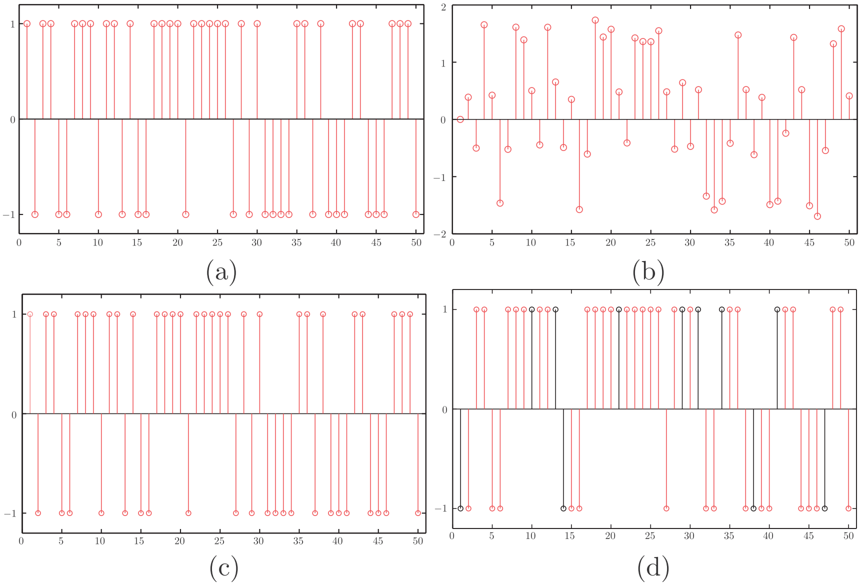


FIGURE 4.15

(a) A realization of the information sequence comprising equiprobable, randomly generated, ±1 samples of Example 4.2. (b) The received at the receiver-end corresponding sequence. (c) The sequence at the output of the equalizer for a low channel noise case. The original sequence is fully recovered with no errors. (d) The output of the equalizer for high channel noise. The samples in gray are in error and of opposite polarity compared to the originally transmitted samples.

Solving the normal equations,

$$\Sigma_u \boldsymbol{w}_* = \boldsymbol{p},$$

for $\sigma_s^2 = 1$ and $\sigma_\eta^2 = 0.01$, results in

$$\boldsymbol{w}_* = [0.7462, 0.1195, -0.0474]^{\mathrm{T}}.$$

Figure 4.15c shows the recovered sequence by the equalizer ($\boldsymbol{w}_*^{\mathrm{T}} \boldsymbol{u}_n$). It is exactly the same with the transmitted one; no errors. Figure 4.15d shows the recovered sequence for the case where the variance of the noise was increased to $\sigma_\eta^2 = 1$. The corresponding MSE optimal equalizer is equal to

$$\boldsymbol{w}_* = [0.4132, 0.1369, -0.0304]^{\mathrm{T}}.$$

This time, the sequence reconstructed by the equalizer has errors with respect to the transmitted one (gray lines).

A slightly alternative formulation for obtaining Σ_u, instead of computing each one of its elements individually, is the following. Verify that the input vector to the equalizer (with tree taps) at time, n, is given by

$$\mathbf{u}_n = \begin{bmatrix} 0.5 & 1 & 0 & 0 \\ 0 & 0.5 & 1 & 0 \\ 0 & 0 & 0.5 & 1 \end{bmatrix} \begin{bmatrix} \mathrm{s}_n \\ \mathrm{s}_{n-1} \\ \mathrm{s}_{n-2} \\ \mathrm{s}_{n-3} \end{bmatrix} + \begin{bmatrix} \eta_n \\ \eta_{n-1} \\ \eta_{n-2} \\ \eta_{n-3} \end{bmatrix}$$

$$:= H\mathbf{s}_n + \boldsymbol{\eta}_n, \tag{4.59}$$

which results in

$$\begin{aligned} \Sigma_{\mathrm{u}} = \mathbb{E}[\mathbf{u}_n\mathbf{u}_n^{\mathrm{T}}] &= H\sigma_s^2 H^{\mathrm{T}} + \Sigma_\eta \\ &= \sigma_s^2 HH^{\mathrm{T}} + \sigma_\eta^2 I. \end{aligned}$$

The reader can easily verify that this is the same as before. Note, however, that (4.59) reminds us of the linear regression model. Moreover, note the special structure of the matrix H. Such matrices are also known as *convolution* matrices. This structure is imposed by the fact that the elements of $\mathbf{u}_n$ are time-shifted versions of the first element, because the input vector corresponds to a random process. This is exactly the property that will be exploited next to derive efficient schemes for the solution of the normal equations.

4.8 ALGORITHMIC ASPECTS: THE LEVINSON AND THE LATTICE-LADDER ALGORITHMS

The goal of this section is to present algorithmic schemes for the efficient solution of the normal equations in (4.16). The filtering case where the input and output entities are random processes will be considered. In this case, we have already pointed out that the input covariance matrix has a special structure. The main concepts to be presented here have a generality that goes beyond the specific form of the normal equations. A vast literature concerning efficient (fast) algorithms for the least-squares task as well as a number of its online/adaptive versions have their roots to the schemes to be presented here. At the heart of all these schemes lies the specific structure of the input vector, whose elements are *time-shifted versions* of its first element, u_n.

Recall from linear algebra that in order to solve a general linear system of l equations with l unknowns, one requires $O(l^3)$ operations (multiplications-additions (MADs)). Exploiting the rich structure of the autocorrelation/covariance matrix, associated with random processes, an algorithm with $O(l^2)$ operations will be derived. The more general complex-valued case will be considered.

The autocorrelation/covariance matrix of the input random vector has been defined in (4.17). That is, it is Hermitian as well as semipositive definite. From now on, we will assume that it is positive definite. The autocorrelation/covariance matrix in $\mathbb{C}^{m\times m}$, associated with a complex wide-sense stationary process, is given by

$$\Sigma_m = \begin{bmatrix} r(0) & r(1) & \cdots & r(m-1) \\ r(-1) & r(0) & \cdots & r(m-2) \\ \vdots & \vdots & \ddots & \vdots \\ r(-m+1) & r(-m+2) & \cdots & r(0) \end{bmatrix}$$

$$= \begin{bmatrix} r(0) & r(1) & \cdots & r(m-1) \\ r^*(1) & r(0) & \cdots & r(m-2) \\ \vdots & \vdots & \ddots & \vdots \\ r^*(m-1) & r^*(m-2) & \cdots & r(0) \end{bmatrix},$$

where the property

$$r(i) := \mathbb{E}[\mathrm{u}_n \mathrm{u}_{n-i}^*] = \mathbb{E}\left[\left(\mathrm{u}_{n-i}\mathrm{u}_n^*\right)^*\right] := r^*(-i)$$

has been used. We have relaxed the notational dependence of Σ on u and we have instead explicitly indicated the order of the matrix, because this will be a very useful index from now on.

We will follow a recursive approach, and our aim will be to express the optimal filter solution of order m, denoted from now on as $\boldsymbol{w}_m$, in terms of the optimal one, $\boldsymbol{w}_{m-1}$, of order $m-1$.

The covariance matrix of a wide-sense stationary process is a *Toeplitz* matrix; all the elements along *any* of its diagonals are equal. This property together with its Hermitian nature give rise to the following *nested* structure,

$$\Sigma_m = \begin{bmatrix} \Sigma_{m-1} & J_{m-1}\boldsymbol{r}_{m-1} \\ \boldsymbol{r}_{m-1}^{\mathrm{H}} J_{m-1} & r(0) \end{bmatrix} \tag{4.60}$$

$$= \begin{bmatrix} r(0) & \boldsymbol{r}_{m-1}^{\mathrm{T}} \\ \boldsymbol{r}_{m-1}^* & \Sigma_{m-1} \end{bmatrix}, \tag{4.61}$$

where

$$\boldsymbol{r}_{m-1} := \begin{bmatrix} r(1) \\ r(2) \\ \vdots \\ r(m-1) \end{bmatrix}, \tag{4.62}$$

and J_{m-1} is the antidiagonal matrix of dimension $(m-1) \times (m-1)$, defined as

$$J_{m-1} := \begin{bmatrix} 0 & 0 & \cdots & 1 \\ 0 & 0 & \iddots & 0 \\ 0 & 1 & \cdots & 0 \\ 1 & 0 & \cdots & 0 \end{bmatrix}.$$

Note that right-multiplication of any matrix by J_{m-1} has as an effect to reverse the order of its columns, while multiplying it from the left reverses the order of the rows as follows

$$\boldsymbol{r}_{m-1}^{\mathrm{H}} J_{m-1} = \left[r^*(m-1) \;\; r^*(m-2) \;\; \cdots \;\; r^*(1)\right],$$

and

$$J_{m-1}\boldsymbol{r}_{m-1} = \left[r(m-1) \;\; r(m-2) \;\; \cdots \;\; r(1)\right]^{\mathrm{T}}.$$

Applying the matrix inversion lemma from Appendix A.1 for the upper partition in (4.60), we obtain

$$\Sigma_m^{-1} = \begin{bmatrix} \Sigma_{m-1}^{-1} & \mathbf{0} \\ \mathbf{0}^{\mathrm{T}} & 0 \end{bmatrix} + \begin{bmatrix} -\Sigma_{m-1}^{-1} J_{m-1} \boldsymbol{r}_{m-1} \\ 1 \end{bmatrix} \frac{1}{\alpha_{m-1}^{b}} \begin{bmatrix} -\boldsymbol{r}_{m-1}^{\mathrm{H}} J_{m-1} \Sigma_{m-1}^{-1} & 1 \end{bmatrix}, \tag{4.63}$$

where for this case the so-called *Schur complement* is the scalar

$$\alpha_{m-1}^{b} = r(0) - \boldsymbol{r}_{m-1}^{\mathrm{H}} J_{m-1} \Sigma_{m-1}^{-1} J_{m-1} \boldsymbol{r}_{m-1}. \tag{4.64}$$

The cross-correlation vector of order m, $\boldsymbol{p}_m$, admits the following partition,

$$\boldsymbol{p}_m = \begin{bmatrix} \mathbb{E}[\mathrm{u}_n \mathrm{d}_n^*] \\ \vdots \\ \mathbb{E}[\mathrm{u}_{n-m+2} \mathrm{d}_n^*] \\ \mathbb{E}[\mathrm{u}_{n-m+1} \mathrm{d}_n^*] \end{bmatrix} = \begin{bmatrix} \boldsymbol{p}_{m-1} \\ p_{m-1} \end{bmatrix}, \quad \text{where} \quad p_{m-1} := \mathbb{E}[\mathrm{u}_{n-m+1} \mathrm{d}_n^*]. \tag{4.65}$$

Combining (4.63) and (4.65), the following elegant relation results:

$$\boldsymbol{w}_m := \Sigma_m^{-1} \boldsymbol{p}_m = \begin{bmatrix} \boldsymbol{w}_{m-1} \\ 0 \end{bmatrix} + \begin{bmatrix} -\boldsymbol{b}_{m-1} \\ 1 \end{bmatrix} k_{m-1}^{w}, \tag{4.66}$$

where

$$\boldsymbol{w}_{m-1} = \Sigma_{m-1}^{-1} \boldsymbol{p}_{m-1}, \quad \boldsymbol{b}_{m-1} := \Sigma_{m-1}^{-1} J_{m-1} \boldsymbol{r}_{m-1},$$

and

$$k_{m-1}^{w} := \frac{p_{m-1} - \boldsymbol{r}_{m-1}^{\mathrm{H}} J_{m-1} \boldsymbol{w}_{m-1}}{\alpha_{m-1}^{b}}. \tag{4.67}$$

Equation (4.66) is an order recursion that relates the optimal solution $\boldsymbol{w}_m$ with $\boldsymbol{w}_{m-1}$. In order to obtain a complete recursive scheme, all one needs is a recursion for updating $\boldsymbol{b}_m$.

Forward and backward MSE optimal predictors

Backward Prediction: The vector $\boldsymbol{b}_m = \Sigma_m^{-1} J_m \boldsymbol{r}_m$ has an interesting physical interpretation: it is the MSE-optimal backward predictor of order m. That is, it is the linear filter, which optimally estimates/predicts the value of u_{n-m} given the values of $\mathrm{u}_{n-m+1}, \mathrm{u}_{n-m+2}, \ldots, \mathrm{u}_n$. Thus, in order to design the optimal backward predictor of order m, the desired response must be

$$\mathrm{d}_n = \mathrm{u}_{n-m},$$

and from the respective normal equations we get

$$\boldsymbol{b}_m = \Sigma_m^{-1} \begin{bmatrix} \mathbb{E}[\mathrm{u}_n \mathrm{u}_{n-m}^*] \\ \mathbb{E}[\mathrm{u}_{n-1} \mathrm{u}_{n-m}^*] \\ \vdots \\ \mathbb{E}[\mathrm{u}_{n-m+1} \mathrm{u}_{n-m}^*] \end{bmatrix} = \Sigma_m^{-1} J_m \boldsymbol{r}_m. \tag{4.68}$$

Hence, the MSE-optimal backward predictor coincides with $\boldsymbol{b}_m$, i.e.,

$$\boxed{\boldsymbol{b}_m = \Sigma_m^{-1} J_m \boldsymbol{r}_m : \quad \text{MSE-Optimal Backward Predictor.}}$$

Moreover, the corresponding minimum mean-square error, adapting (4.19) to our current needs, is equal to

$$J(\boldsymbol{b}_m) = r(0) - \boldsymbol{r}_m^{\mathrm{H}} J_m \Sigma_m^{-1} J_m \boldsymbol{r}_m = \alpha_m^b.$$

That is, the Schur complement in (4.64) is equal to the respective optimal mean-square error!

Forward Prediction: The goal of the forward prediction task is to predict the value u_{n+1}, given the values $\mathrm{u}_n, \mathrm{u}_{n-1}, \ldots, \mathrm{u}_{n-m+1}$. Thus, the MSE-optimal forward predictor of order m, $\boldsymbol{a}_m$, is obtained by selecting the desired response $\mathrm{d}_n = \mathrm{u}_{n+1}$, and the respective normal equations become

$$\boldsymbol{a}_m = \Sigma_m^{-1} \begin{bmatrix} \mathbb{E}[\mathrm{u}_n \mathrm{u}_{n+1}^*] \\ \mathbb{E}[\mathrm{u}_{n-1} \mathrm{u}_{n+1}^*] \\ \vdots \\ \mathbb{E}[\mathrm{u}_{n-m+1} \mathrm{u}_{n+1}^*] \end{bmatrix} = \Sigma_m^{-1} \begin{bmatrix} r^*(1) \\ r^*(2) \\ \vdots \\ r^*(m) \end{bmatrix} \tag{4.69}$$

or

$$\boxed{\boldsymbol{a}_m = \Sigma_m^{-1} \boldsymbol{r}_m^* : \quad \text{MSE-Optimal Forward Predictor.}} \tag{4.70}$$

From (4.70), it is not difficult to show (Problem 4.16) that (recall that $J_m J_m = I_m$)

$$\boldsymbol{a}_m = J_m \boldsymbol{b}_m^* \Rightarrow \boldsymbol{b}_m = J_m \boldsymbol{a}_m^*, \tag{4.71}$$

and that the optimal mean-square error for the forward prediction, $J(\boldsymbol{a_m}) := \alpha_m^f$, is equal to that for the backward, i.e.,

$$J(\boldsymbol{a_m}) = \alpha_m^f = \alpha_m^b = J(\boldsymbol{b}_m).$$

Figure 4.16 depicts the two prediction tasks. In other words, the optimal forward predictor is the conjugate reverse of the backward one, so that

$$\boldsymbol{a}_m := \begin{bmatrix} a_m(0) \\ \vdots \\ a_m(m-1) \end{bmatrix} = J_m \boldsymbol{b}_m^* := \begin{bmatrix} b_m^*(m-1) \\ \vdots \\ b_m^*(0) \end{bmatrix}.$$

This property is due to the stationarity of the involved process. Because the statistical properties only depend on the difference of the time instants, forward and backward predictions are not much different; in both cases, given a set of samples, $u_{n-m+1}, \ldots, u_n$, we predict *one* sample ahead in the future (u_{n+1} in the forward prediction) or *one* sample back in the past (u_{n-m} in the backward prediction).

Having established the relationship between $\boldsymbol{a}_m$ and $\boldsymbol{b}_m$ in (4.71), we are ready to complete the missing step in (4.66); that is, to complete an order recursive step for the update of $\boldsymbol{b}_m$. Since (4.66)

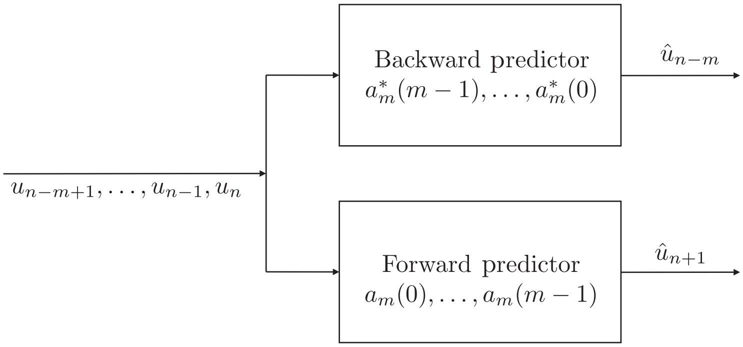

Backward predictor
$a_m^*(m-1), \ldots, a_m^*(0)$
$\hat{u}_{n-m}$
$u_{n-m+1}, \ldots, u_{n-1}, u_n$
Forward predictor
$a_m(0), \ldots, a_m(m-1)$
$\hat{u}_{n+1}$

FIGURE 4.16

The impulse response of the backward predictor is the conjugate reverse of that of the forward predictor.

holds true for any desired response, d_n, it also applies for the special case where the optimal filter to be designed is the forward predictor $\boldsymbol{a}_m$; in this case, $\mathrm{d}_n = \mathrm{u}_{n+1}$. Replacing in (4.66) $\boldsymbol{w}_m$ ($\boldsymbol{w}_{m-1}$) with $\boldsymbol{a}_m$ ($\boldsymbol{a}_{m-1}$) results in

$$\boldsymbol{a}_m = \begin{bmatrix} \boldsymbol{a}_{m-1} \\ 0 \end{bmatrix} + \begin{bmatrix} -J_{m-1}\boldsymbol{a}_{m-1}^* \\ 1 \end{bmatrix} k_{m-1}, \tag{4.72}$$

where (4.71) has been used and

$$k_{m-1} = \frac{r^*(m) - \boldsymbol{r}_{m-1}^{\mathrm{H}} J_{m-1}\boldsymbol{a}_{m-1}}{\alpha_{m-1}^b}. \tag{4.73}$$

Combining (4.66), (4.67), (4.71), (4.72), and (4.73) the following algorithm, known as *Levinson's* algorithm, for the solution of the normal equations results:

Algorithm 4.1 (Levinson's algorithm).

- Input
  - $r(0), r(1), \ldots, r(l)$
  - $p_k = \mathbb{E}[\mathrm{u}_{n-k}\mathrm{d}_n^*], \; k = 0, 1, \ldots, l-1$
- Initialize
  - $w_1 = \frac{p_0}{r(0)}, \; a_1 = \frac{r^*(1)}{r(0)}, \; \alpha_1^b = r(0) - \frac{|r(1)|^2}{r(0)}$
  - $k_1^w = \frac{p_1 - r^*(1)w_1}{\alpha_1^b}, \; k_1 = \frac{r^*(2) - r^*(1)a_1}{\alpha_1^b}$
- **For** $m = 2, \ldots, l-1$, **Do**
  - $\boldsymbol{w}_m = \begin{bmatrix} \boldsymbol{w}_{m-1} \\ 0 \end{bmatrix} + \begin{bmatrix} -J_{m-1}\boldsymbol{a}_{m-1}^* \\ 1 \end{bmatrix} k_{m-1}^w$
  - $\boldsymbol{a}_m = \begin{bmatrix} \boldsymbol{a}_{m-1} \\ 0 \end{bmatrix} + \begin{bmatrix} -J_{m-1}\boldsymbol{a}_{m-1}^* \\ 1 \end{bmatrix} k_{m-1}$
  - $\alpha_m^b = \alpha_{m-1}^b(1 - |k_{m-1}|^2)$

  - $k_m^w = \frac{p_m - \boldsymbol{r}_m^{\mathrm{H}} J_m \boldsymbol{w}_m}{\alpha_m^b}$
  - $k_m = \frac{r^*(m+1) - \boldsymbol{r}_m^{\mathrm{H}} J_m \boldsymbol{a}_m}{\alpha_m^b}$
- **End For**

Note that the update for α_m^b is a direct consequence of its definition in (4.64) and (4.72) (Problem 4.17). Also note that $\alpha_m^b \geq 0$ implies that $|k_m| \leq 1$.

Remarks 4.3.

- The complexity per order recursion is $4m$ MADS, hence for a system with l equations this amounts to $2l^2$ MADS. This computational saving is substantial compared to the $O(l^3)$ MADS, required by adopting a general purpose scheme. The previous very elegant scheme was proposed in 1947 by Levinson, [26]. A formulation of the algorithm was also independently proposed by Durbin, [12] and the algorithm is usually called the *Levinson-Durbin* algorithm. In [11], it was shown that Levinson's algorithm is redundant in its prediction part and the *split Levinson* algorithm was developed, whose recursions evolve around symmetric vector quantities leading to further computational savings.

4.8.1 THE LATTICE-LADDER SCHEME

So far, we have been involved with the so called *transversal* implementation of an LTI FIR filter; in other words, the output is expressed as a convolution between the impulse response and the input of the linear structure. Levinson's algorithm provided a computationally efficient scheme for obtaining the MSE-optimal estimate $\boldsymbol{w}_*$. We now turn our attention to an equivalent implementation of the corresponding linear filter, which comes as a direct consequence of Levinson's algorithm.

Define the error signals associated with the mth order optimal forward and backward predictors, at time instant n, as

$$\mathrm{e}_m^f(n) := \mathrm{u}_n - \boldsymbol{a}_m^{\mathrm{H}} \mathbf{u}_m(n-1), \tag{4.74}$$

where $\mathbf{u}_m(n)$ is the input random vector of the mth order filter, and the order of the filter has been explicitly brought into the notation.[7] The backward error is given by

$$\begin{aligned} \mathrm{e}_m^b(n) :&= \mathrm{u}_{n-m} - \boldsymbol{b}_m^{\mathrm{H}} \mathbf{u}_m(n) \\ &= \mathrm{u}_{n-m} - \boldsymbol{a}_m^{\mathrm{T}} J_m \mathbf{u}_m(n). \end{aligned} \tag{4.75}$$

Employing in (4.74), (4.75), the order recursion in (4.72) and the partitioning of $\mathbf{u}_m(n)$, which is represented by

$$\mathbf{u}_m(n) = [\mathbf{u}_{m-1}^T(n), \mathrm{u}_{n-m+1}]^T = [\mathrm{u}_n, \mathbf{u}_{m-1}^T(n-1)]^T, \tag{4.76}$$

we readily obtain

$$\mathrm{e}_m^f(n) = \mathrm{e}_{m-1}^f(n) - \mathrm{e}_{m-1}^b(n-1)k_{m-1}^*, \quad m = 1, 2, \ldots, l, \tag{4.77}$$

$$\mathrm{e}_m^b(n) = \mathrm{e}_{m-1}^b(n-1) - \mathrm{e}_{m-1}^f(n)k_{m-1}, \quad m = 1, 2, \ldots, l, \tag{4.78}$$

[7] The time index is now given in parentheses, to avoid having double subscripts.

with $\mathrm{e}_0^f(n) = \mathrm{e}_0^b(n) = \mathrm{u}_n$, and $k_0 = \frac{r^*(1)}{r(0)}$. This pair of recursions is known as *lattice* recursions. Let us focus a bit more on this set of equations.

Orthogonality of the optimal backward errors

From the vector space interpretation of random signals, it is apparent that $\mathrm{e}_m^b(n)$ lies in the subspace spanned by $\mathrm{u}_{n-m}, \ldots, \mathrm{u}_n$, and we can write

$$\mathrm{e}_m^b(n) \in \mathrm{span}\{\mathrm{u}(n-m), \ldots, \mathrm{u}(n)\}.$$

Moreover, because $\mathrm{e}_m^b(n)$ is the error associated with the MSE-optimal backward predictor, $\mathrm{e}_m^b(n) \perp \mathrm{span}\{\mathrm{u}(n-m+1), \ldots, \mathrm{u}(n)\}$. However, the latter subspace is the one where $\mathrm{e}_{m-k}^b(n),\ k = 1, 2, \ldots, m$, lie. Hence, for $m = 1, 2, \ldots, l-1$, we can write,

$$\boxed{\mathrm{e}_m^b(n) \perp \mathrm{e}_k^b(n),\ k < m: \quad \text{Orthogonality of the Backward Errors.}}$$

Moreover, it is obvious that

$$\mathrm{span}\{\mathrm{e}_0^b(n), \mathrm{e}_1^b(n), \ldots, \mathrm{e}_{l-1}^b(n)\} = \mathrm{span}\{\mathrm{u}_n, \mathrm{u}_{n-1}, \ldots, \mathrm{u}_{n-l+1}\}.$$

Hence, the normalized vectors

$$\boxed{\tilde{\mathrm{e}}_m^b(n) := \frac{\mathrm{e}_m^b(n)}{||\mathrm{e}_m^b(n)||}, \quad m = 0, 1, \ldots, l-1: \quad \text{Orthonormal Basis,}}$$

form an *orthonormal* basis in $\mathrm{span}\{\mathrm{u}_n, \mathrm{u}_{n-1}, \ldots, \mathrm{u}_{n-l+1}\}$, see Figure 4.17. As a matter of fact, the pair in (4.77), (4.78) comprise a *Gram-Schmidt* orthogonalizer [47].

Let us now express $\hat{\mathrm{d}}_n$, or the projection of d_n in $\mathrm{span}\{\mathrm{u}_n, ..., \mathrm{u}_{n-l+1}\}$, in terms of the new set of orthogonal vectors,

$$\hat{\mathrm{d}}_n = \sum_{m=0}^{l-1} h_m \mathrm{e}_m^b(n), \tag{4.79}$$

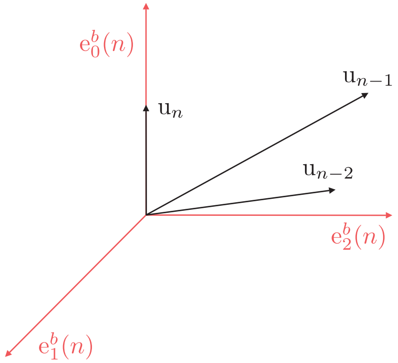


FIGURE 4.17

The optimal backward errors form an orthogonal basis in the respective input random signal space.

where the coefficients h_m are given by

$$h_m = \langle \hat{\mathrm{d}}_n, \frac{\mathrm{e}_m^b(n)}{||\mathrm{e}_m^b(n)||^2} \rangle = \frac{\mathbb{E}[\hat{\mathrm{d}}_n \mathrm{e}_m^{b*}(n)]}{||\mathrm{e}_m^b(n)||^2} = \frac{\mathbb{E}[(\mathrm{d}_n - \mathrm{e}_n)\, \mathrm{e}_m^{b*}(n)]}{||\mathrm{e}_m^b(n)||^2}$$

$$= \frac{\mathbb{E}[\mathrm{d}_n \mathrm{e}_m^{b*}(n)]}{||\mathrm{e}_m^b(n)||^2}, \tag{4.80}$$

where the orthogonality of the error, e_n, with the subspace spanned by the backward errors has been taken into account. From (4.67) and (4.80), and taking into account the respective definitions of the involved quantities, we readily obtain that

$$h_m = k_m^{w*}.$$

That is, the coefficients k_m^w, $m = 0, 1, \ldots, l-1$, in Levinson's algorithm are the parameters in the expansion of $\hat{\mathrm{d}}_n$ in terms of the orthogonal basis. Combining (4.77), (4.78) and (4.79) the *lattice-ladder* scheme of Figure 4.18 results, whose output is the MSE approximation $\hat{\mathrm{d}}_n$ of d_n.

Remarks 4.4.

- The lattice-ladder scheme is a highly efficient, *modular* structure. It comprises a sequence of successive similar stages. To increase the order of the filter, it suffices to add an extra stage, which is a highly desirable property in VLSI implementations. Moreover, lattice-ladder schemes enjoy a higher robustness, compared to Levinson's algorithm, with respect to numerical inaccuracies.
- *Cholesky Factorization*. The orthogonality property of the optimal MSE backward errors leads to another interpretation of the involved parameters. From the definition in (4.75), we get

$$\mathbf{e}_l^b(n) := \begin{bmatrix} \mathrm{e}_0^b(n) \\ \mathrm{e}_1^b(n) \\ \vdots \\ \mathrm{e}_{l-1}^b(n) \end{bmatrix} = U^{\mathrm{H}} \begin{bmatrix} \mathrm{u}_n \\ \mathrm{u}_{n-1} \\ \vdots \\ \mathrm{u}_{n-l+1} \end{bmatrix} = U^{\mathrm{H}} \mathbf{u}_l(n), \tag{4.81}$$

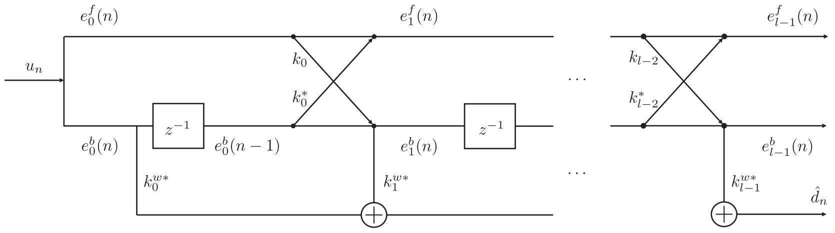


FIGURE 4.18

The lattice-ladder structure. In contrast to the transversal implementation in terms of $\boldsymbol{w}_m$, the parameterization is now in terms of k_m, k_m^w, $m = 0, 1, \ldots, l-1$, $k_0 = \frac{r^*(1)}{r(0)}$, and $k_0^w = \frac{p_0^*}{r(0)}$. Note the resulting highly modular structure.

where

$$U^{\mathrm{H}} := \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ -a_1(0) & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & 0 \\ -a_l(l-1) & -a_l(l-2) & \cdots & \cdots & 1 \end{bmatrix}$$

and

$$\boldsymbol{a}_m := [a_m(0), a_m(1), \ldots, a_m(m-1)]^{\mathrm{T}}, m = 1, 2, \ldots, l.$$

Due to the orthogonality of the involved backward errors,

$$\mathbb{E}[\mathbf{e}_l^b(n)\mathbf{e}_l^{bH}(n)] = U^{\mathrm{H}} \Sigma_l U = D$$

where

$$D := \operatorname{diag}\left\{\alpha_0^b, \alpha_1^b, \ldots, \alpha_{l-1}^b\right\},$$

or

$$\Sigma_l^{-1} = UD^{-1}U^{\mathrm{H}} = (UD^{-1/2})(UD^{-1/2})^{\mathrm{H}}.$$

That is, the prediction error powers and the weights of the optimal forward predictor provide the *Cholesky factorization* of the inverse covariance matrix.

- *The Schur Algorithm.* In a parallel processing environment, the inner products involved in Levinson's algorithm pose a bottleneck in the flow of the algorithm. Note that the updates for $\boldsymbol{w}_m$ and $\boldsymbol{a}_m$ can be performed fully in parallel. Schur's algorithm [45] is an alternative scheme that overcomes the bottleneck, and in a multiprocessor environment the complexity can go down to $O(l)$. The parameters involved in Schur's algorithm perform a Cholesky factorization of Σ_l (e.g., [21, 22]).
- Note that all these algorithmic schemes for the efficient solution of the normal equations owe their existence to the rich structure that the (autocorrelation) covariance matrix as well as the cross-correlation vector acquire when the involved jointly distributed random entities are random processes; their *time sequential nature* imposes such a structure. The derivation of the Levinson and lattice-ladder schemes reveal the flavor of the type of techniques that can be (and have extensively been) used to derive computational schemes for the online/adaptive versions and the related least-squares error loss function, to be discussed in Chapter 6. There, the algorithms may be computationally more involved, but the essence behind them is the same as for those used in the current section.

4.9 MEAN-SQUARE ERROR ESTIMATION OF LINEAR MODELS

We now turn our attention to the case where the underlying model that relates the input-output variables is a linear one. Not to be confused with what was treated in the previous sections, it must be stressed that, so far, we have been concerned with the linear estimator task. At no point in this stage of our discussion has the generation model of the data been brought in (with the exception in the comment of Remarks 4.2). We just adopted a linear estimator and obtained the MSE solution for it. The focus was on the solution and its properties. The emphasis here is on cases where the input-output variables are related via a linear data generation model.

Let us assume that we are given two jointly distributed random vectors, $\mathbf{y}$ and $\boldsymbol{\theta}$, which are related according to the following linear model,

$$\mathbf{y} = X\boldsymbol{\theta} + \boldsymbol{\eta}, \tag{4.82}$$

where $\boldsymbol{\eta}$ denotes the set of the involved noise variables. Note that such a model covers the case of our familiar regression task, where the unknown parameters $\boldsymbol{\theta}$ are considered random, which is in line with the Bayesian philosophy, as discussed in Chapter 3. Once more, we assume zero-mean vectors; otherwise, the respective mean values are subtracted. The dimensions of $\mathbf{y}$ ($\boldsymbol{\eta}$) and $\boldsymbol{\theta}$ may not necessarily be the same; to be in line with the notation used in Chapter 3, let $\mathbf{y}, \boldsymbol{\eta} \in \mathbb{R}^N$ and $\boldsymbol{\theta} \in \mathbb{R}^l$. Hence, X is a $N \times l$ matrix. Note that the matrix X is considered to be deterministic and not a random one.

Assume the covariance matrices of our zero-mean variables,

$$\Sigma_\theta = \mathbb{E}[\boldsymbol{\theta}\boldsymbol{\theta}^{\mathrm{T}}], \quad \Sigma_\eta = \mathbb{E}[\boldsymbol{\eta}\boldsymbol{\eta}^{\mathrm{T}}],$$

are known. The goal is to compute a matrix, H, of dimension $l \times N$, so that the linear estimator

$$\hat{\boldsymbol{\theta}} = H\mathbf{y} \tag{4.83}$$

minimizes the mean-square error cost

$$J(H) := \mathbb{E}\left[(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})^{\mathrm{T}}(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})\right] = \sum_{i=1}^{l} \mathbb{E}\left[|\theta_i - \hat{\theta}_i|^2\right]. \tag{4.84}$$

Note that this is a *multichannel* estimation task and it is equivalent with solving l optimization tasks, one for each component, θ_i, of $\boldsymbol{\theta}$. Defining the error vector as

$$\boldsymbol{\varepsilon} := \boldsymbol{\theta} - \hat{\boldsymbol{\theta}},$$

the cost function is equal to the trace of the corresponding *error covariance matrix*, so that

$$J(H) := \operatorname{trace}\left\{\mathbb{E}[\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}^{\mathrm{T}}]\right\}.$$

Focusing on the i-th component in (4.83), we write

$$\hat{\theta}_i = \boldsymbol{h}_i^{\mathrm{T}}\mathbf{y}, \quad i = 1, 2, \ldots, l, \tag{4.85}$$

where $\boldsymbol{h}_i^{\mathrm{T}}$ is the i-th row of H and its optimal estimate is given by

$$\boldsymbol{h}_{*,i} := \arg\min_{\boldsymbol{h}_i} \mathbb{E}\left[|\theta_i - \hat{\theta}_i|^2\right] = \mathbb{E}\left[|\theta_i - \boldsymbol{h}_i^{\mathrm{T}}\mathbf{y}|^2\right]. \tag{4.86}$$

Minimizing (4.86) is exactly the same task as that of the linear estimation considered in previous section (with $\mathbf{y}$ in place of $\mathbf{x}$ and θ_i in place of y), hence,

$$\Sigma_y \boldsymbol{h}_{*,i} = \boldsymbol{p}_i, \quad i = 1, 2, \ldots, l,$$

where

$$\Sigma_y = \mathbb{E}[\mathbf{y}\mathbf{y}^{\mathrm{T}}] \quad \text{and} \quad \boldsymbol{p}_i = \mathbb{E}[\mathbf{y}\theta_i], \quad i = 1, 2, .., l,$$

or

$$\boldsymbol{h}_{*,i}^{\mathrm{T}} = \boldsymbol{p}_i^{\mathrm{T}} \Sigma_y^{-1}, \quad i = 1, 2, \ldots, l,$$

and finally,

$$H_* = \Sigma_{y\theta} \Sigma_y^{-1}, \quad \hat{\boldsymbol{\theta}} = \Sigma_{y\theta} \Sigma_y^{-1} \mathbf{y}, \tag{4.87}$$

where

$$\Sigma_{y\theta} := \begin{bmatrix} \boldsymbol{p}_1^{\mathrm{T}} \\ \boldsymbol{p}_2^{\mathrm{T}} \\ \vdots \\ \boldsymbol{p}_l^{\mathrm{T}} \end{bmatrix} = \mathbb{E}[\boldsymbol{\theta}\mathbf{y}^{\mathrm{T}}] \tag{4.88}$$

is an $l \times N$ cross-correlation matrix. All that is now required is to compute Σ_y and $\Sigma_{y\theta}$. To this end,

$$\begin{aligned} \Sigma_y = \mathbb{E}\left[\mathbf{y}\mathbf{y}^{\mathrm{T}}\right] &= \mathbb{E}\left[(X\boldsymbol{\theta} + \boldsymbol{\eta})\left(\boldsymbol{\theta}^{\mathrm{T}} X^{\mathrm{T}} + \boldsymbol{\eta}^{\mathrm{T}}\right)\right] \\ &= X \Sigma_\theta X^{\mathrm{T}} + \Sigma_\eta, \end{aligned} \tag{4.89}$$

where the independence of the zero mean vectors $\boldsymbol{\theta}$ and $\boldsymbol{\eta}$ has been used. Similarly,

$$\Sigma_{y\theta} = \mathbb{E}\left[\boldsymbol{\theta}\mathbf{y}^{\mathrm{T}}\right] = \mathbb{E}\left[\boldsymbol{\theta}\left(\boldsymbol{\theta}^{\mathrm{T}} X^{\mathrm{T}} + \boldsymbol{\eta}^{\mathrm{T}}\right)\right] = \Sigma_\theta X^{\mathrm{T}}, \tag{4.90}$$

and combining (4.87), (4.89), and (4.90), we obtain

$$\hat{\boldsymbol{\theta}} = \Sigma_\theta X^{\mathrm{T}} \left(\Sigma_\eta + X \Sigma_\theta X^{\mathrm{T}}\right)^{-1} \mathbf{y}. \tag{4.91}$$

Employing from Appendix A.1 the matrix identity

$$\left(A^{-1} + B^{\mathrm{T}} C^{-1} B\right)^{-1} B^{\mathrm{T}} C^{-1} = A B^{\mathrm{T}} \left(B A B^{\mathrm{T}} + C\right)^{-1}$$

in (4.91) we obtain

$$\boxed{\hat{\boldsymbol{\theta}} = (\Sigma_\theta^{-1} + X^{\mathrm{T}} \Sigma_\eta^{-1} X)^{-1} X^{\mathrm{T}} \Sigma_\eta^{-1} \mathbf{y}: \quad \text{MSE Linear Estimator.}} \tag{4.92}$$

In case of complex-valued variables, the only difference is that transposition is replaced by Hermitian transposition.

Remarks 4.5.

- Recall from Chapter 3 that the optimal MSE estimator of $\boldsymbol{\theta}$ given the values of $\mathbf{y}$ is provided by

$$\mathbb{E}[\boldsymbol{\theta}|\mathbf{y}].$$

However, as it was shown in Problem 3.14, if $\boldsymbol{\theta}$ and $\mathbf{y}$ are jointly Gaussian vectors, then the optimal estimator is linear (affine for nonzero mean variables) and it coincides with the MSE linear estimator of (4.92).

- If we allow nonzero mean values, then instead of (4.83) the affine model should be adopted,

$$\hat{\boldsymbol{\theta}} = H\mathbf{y} + \boldsymbol{\mu}.$$

Then

$$\mathbb{E}[\hat{\boldsymbol{\theta}}] = H\,\mathbb{E}\left[\mathbf{y}\right] + \boldsymbol{\mu} \Rightarrow \boldsymbol{\mu} = \mathbb{E}[\hat{\boldsymbol{\theta}}] - H\,\mathbb{E}[\mathbf{y}].$$

Hence,

$$\hat{\theta} = \mathbb{E}[\hat{\theta}] + H\left(\mathbf{y} - \mathbb{E}[\mathbf{y}]\right),$$

and finally,

$$\hat{\theta} - \mathbb{E}[\hat{\theta}] = H\left(\mathbf{y} - \mathbb{E}[\mathbf{y}]\right),$$

which justifies our approach to subtract the means and work with zero-mean value variables. For nonzero mean values, the analogue of (4.92) is

$$\hat{\theta} = \mathbb{E}[\hat{\theta}] + \left(\Sigma_\theta^{-1} + X^{\mathrm{T}}\Sigma_\eta^{-1}X\right)^{-1} X^{\mathrm{T}}\Sigma_\eta^{-1}\left(\mathbf{y} - \mathbb{E}[\mathbf{y}]\right). \tag{4.93}$$

Note that for zero-mean noise $\boldsymbol{\eta}$, $\mathbb{E}[\mathbf{y}] = X\,\mathbb{E}[\boldsymbol{\theta}]$.

- Compare (4.93) with (3.71) for the Bayesian inference approach. They are identical, provided that the covariance matrix of the prior (Gaussian) pdf is equal to Σ_θ and $\boldsymbol{\theta}_0 = \mathbb{E}[\hat{\theta}]$ for a zero-mean noise variable.

4.9.1 THE GAUSS-MARKOV THEOREM

We now turn our attention to the case where $\boldsymbol{\theta}$ in the regression model is considered to be an (unknown) constant, instead of a random vector. Thus, the linear model is now written as

$$\mathbf{y} = X\boldsymbol{\theta} + \boldsymbol{\eta}, \tag{4.94}$$

and the randomness of $\mathbf{y}$ is solely due to $\boldsymbol{\eta}$, which is assumed to be zero-mean with covariance matrix Σ_η. The goal is to design an *unbiased linear* estimator of $\boldsymbol{\theta}$, that minimizes the mean-square error,

$$\hat{\theta} = H\mathbf{y}, \tag{4.95}$$

and select H such as

$$\begin{aligned} &\text{minimize} \quad \text{trace}\left\{\mathbb{E}\left[(\boldsymbol{\theta} - \hat{\theta})(\boldsymbol{\theta} - \hat{\theta})^T\right]\right\} \\ &\text{subject to} \quad \mathbb{E}[\hat{\theta}] = \boldsymbol{\theta}. \end{aligned} \tag{4.96}$$

From (4.94) and (4.95), we get that

$$\mathbb{E}[\hat{\theta}] = H\,\mathbb{E}[\mathbf{y}] = H\,\mathbb{E}\left[(X\boldsymbol{\theta} + \boldsymbol{\eta})\right] = HX\boldsymbol{\theta},$$

which implies that the unbiased constraint is equivalent to

$$HX = I. \tag{4.97}$$

Employing (4.95), the error vector becomes

$$\boldsymbol{\epsilon} = \boldsymbol{\theta} - \hat{\theta} = \boldsymbol{\theta} - H\mathbf{y} = \boldsymbol{\theta} - H(X\boldsymbol{\theta} + \boldsymbol{\eta}) = H\boldsymbol{\eta}. \tag{4.98}$$

Hence, the constrained minimization in (4.96) can now be written as

$$\begin{aligned} &H_* = \arg\min_H \text{trace}\{H\Sigma_\eta H^T\}, \\ &\text{s.t.} \quad HX = I. \end{aligned} \tag{4.99}$$

Solving (4.99) results in (Problem 4.18)

$$H_* = (X^T \Sigma_\eta^{-1} X)^{-1} X^T \Sigma_\eta^{-1}, \tag{4.100}$$

and the associated minimum mean-square error is

$$J(H_*) := \text{MSE}(H_*) = \text{trace}\left\{(X^T \Sigma_\eta^{-1} X)^{-1}\right\}. \tag{4.101}$$

The reader can verify that

$$J(H) \geq J(H_*),$$

for any other linear unbiased estimator (Problem 4.19).

The previous result is known as the *Gauss-Markov* theorem. The optimal MSE linear unbiased estimator is given by

$$\boxed{\hat{\theta} = (X^T \Sigma_\eta^{-1} X)^{-1} X^T \Sigma_\eta^{-1} \mathbf{y} : \quad \text{BLUE}} \tag{4.102}$$

and it is also known as the *best linear unbiased estimator (*BLUE*)*, or the *minimum variance unbiased linear estimator*. For complex-valued variables, the transposition is simply replaced by the Hermitian one.

Remarks 4.6.

- For the BLUE to exist, $X^T \Sigma_\eta^{-1} X$ must be invertible. This is guaranteed if Σ_η is positive definite and the $N \times l$ matrix X, $N \geq l$, is full rank (Problem 4.20).
- Observe that the BLUE coincides with the maximum likelihood estimator (Chapter 3), if $\boldsymbol{\eta}$ follows a multivariate Gaussian distribution; recall that under this assumption, the Cramér-Rao bound is achieved. If this is not the case, there may be another unbiased estimator (nonlinear), which results in lower MSE. Recall also from Chapter 3, that there may be a biased estimator that results in lower MSE; see [13, 38] and the references therein for a related discussion.

Example 4.3 (Channel Identification)**.** The task is illustrated in Figure 4.11. Assume that we have access to a set of input-output observations, u_n and $d_n, n = 0, 1, 2, \ldots, N-1$. Moreover, we are given that the impulse response of the system comprises l taps and it is zero-mean and its covariance matrix is Σ_w. Also, the second-order statistics of the zero-mean noise are also known and we are given its covariance matrix, Σ_η. Then, assuming that the plant starts from zero initial conditions, we can adopt the following model relating the involved random variables (in line with the model in (4.82)),

$$\mathbf{d} := \begin{bmatrix} \mathrm{d}_0 \\ \mathrm{d}_1 \\ \vdots \\ \mathrm{d}_{l-1} \\ \vdots \\ \mathrm{d}_{N-1} \end{bmatrix} = U \begin{bmatrix} \mathrm{w}_0 \\ \mathrm{w}_1 \\ \vdots \\ \mathrm{w}_{l-1} \end{bmatrix} + \begin{bmatrix} \eta_0 \\ \eta_1 \\ \vdots \\ \eta_{l-1} \\ \vdots \\ \eta_{N-1} \end{bmatrix}, \tag{4.103}$$

where

$$U := \begin{bmatrix} u_0 & 0 & 0 & \cdots & 0 \\ u_1 & u_0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ u_{l-1} & u_{l-2} & \cdots & \cdots & u_0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ u_{N-1} & \cdots & \cdots & \cdots & u_{N-l} \end{bmatrix}.$$

Note that U is treated deterministically. Then, recalling (4.92) and plugging in the set of obtained measurements, the following estimate results:

$$\hat{\boldsymbol{w}} = (\Sigma_w^{-1} + U^{\mathrm{T}} \Sigma_\eta^{-1} U) U^{\mathrm{T}} \Sigma_\eta^{-1} \boldsymbol{d}. \tag{4.104}$$

4.9.2 CONSTRAINED LINEAR ESTIMATION: THE BEAMFORMING CASE

We have already dealt with a constrained linear estimation task in Section 4.9.1 in our effort to obtain an unbiased estimator of a fixed-value parameter vector. In the current section, we will see that the procedure developed there is readily applicable for cases where the unknown parameter vector is required to respect certain linear constraints.

We will demonstrate such a constrained task in the context of *beamforming*. Figure 4.19 illustrates the basic block diagram of the beamforming task. A beamformer comprises a set of antenna elements. We consider the case where the antenna elements are uniformly spaced along a straight line. The goal is to linearly combine the signals received by the individual antenna elements, so as to

- turn the main beam of the array to a specific direction in space, and
- optimally reduce the noise.

The first goal imposes a constraint to the designer, which will guarantee that the gain of the array is high for the specific desired direction; for the second goal, we will adopt MSE arguments.

In a more formal way, assume that the transmitter is far enough away, so as to guarantee that the wavefronts that the array "sees" are planar. Let $\mathrm{s}(t)$ be the information random process transmitted at a carrier frequency, ω_c; hence, the modulated signal is

$$\mathrm{r}(t) = \mathrm{s}(t) e^{j\omega_c t}.$$

If Δx is the distance between successive elements of the array, then a wavefront that arrives at time t_0 at the first element will reach the i-th element delayed by

$$\Delta_{t_i} = t_i - t_0 = i \frac{\Delta x \cos\phi}{c}, \quad i = 0, 1, \ldots, l-1,$$

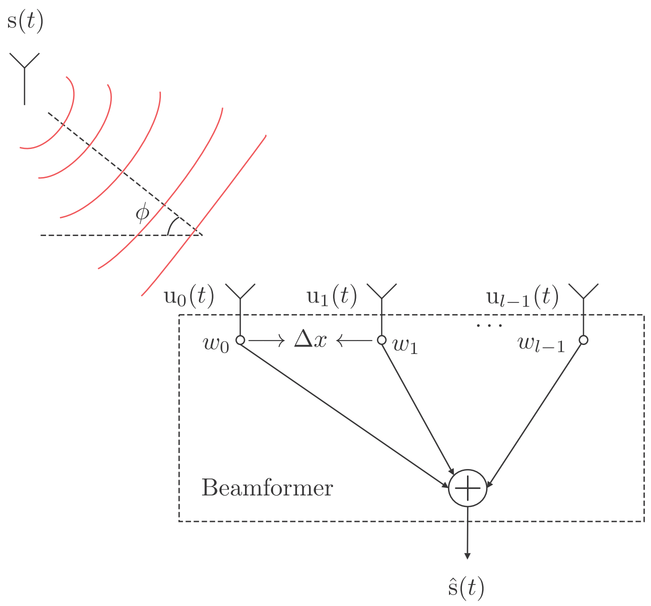


FIGURE 4.19

The task of the beamformer is to obtain estimates of the weights $w_0, \ldots, w_{l-1}$, so as to minimize the effect of noise and at the same time to impose a constraint that, in the absence of noise, would leave signals impinging the array from the desired angle, ϕ, unaffected.

where c is the speed of propagation, ϕ is the angle formed by the array and the direction propagation of the wavefronts, and l the number of array elements. We know from our basic electromagnetic courses that

$$c = \frac{\omega_c \lambda}{2\pi},$$

where λ is the respective wavelength. Taking a snapshot at time t, the signal received from direction ϕ at the i-th element will be

$$\begin{aligned} \mathrm{r}_i(t) &= \mathrm{s}(t - \Delta t_i)e^{j\omega_c(t - i\frac{2\pi\Delta x\cos\phi}{\omega_c\lambda})} \\ &\simeq \mathrm{s}(t)e^{j\omega_c t}e^{-2\pi j\frac{i\Delta x\cos\phi}{\lambda}}, \quad i = 0, 1, \ldots, l-1, \end{aligned}$$

where we have assumed a relatively low time signal variation. After converting the received signals in the baseband (multiplying by $e^{-j\omega_c t}$), the vector of the received signals (one per array element), at time t, can be written in the following linear regression-type formulation,

$$\mathbf{u}(t) := \begin{bmatrix} \mathrm{u}_0(t) \\ \mathrm{u}_1(t) \\ \vdots \\ \mathrm{u}_{l-1}(t) \end{bmatrix} = \boldsymbol{x}\mathrm{s}(t) + \boldsymbol{\eta}(t), \tag{4.105}$$

where

$$\boldsymbol{x} := \begin{bmatrix} 1 \\ e^{-2\pi j \frac{\Delta x \cos\phi}{\lambda}} \\ \vdots \\ e^{-2\pi j \frac{(l-1)\Delta x \cos\phi}{\lambda}} \end{bmatrix},$$

and the vector $\boldsymbol{\eta}(t)$ contains the additive noise plus any other interference due to signals coming from directions other than ϕ, so that

$$\boldsymbol{\eta}(t) = [\eta_0(t), \ldots, \eta_{l-1}(t)]^{\mathrm{T}},$$

and it is assumed to be of zero mean; $\boldsymbol{x}$ is also known as the *steering vector*. The output of the beamformer, acting on the input vector signal, will be

$$\hat{\mathrm{s}}(t) = \boldsymbol{w}^{\mathrm{H}}\mathbf{u}(t),$$

where the Hermitian transposition has to be used, because now the involved signals are complex-valued.

We will first impose the constraint. Ideally, in the absence of noise, one would like to recover signals impinging on the array from the desired direction, ϕ, exactly. Thus, $\boldsymbol{w}$ should satisfy the following constraint

$$\boldsymbol{w}^{\mathrm{H}}\boldsymbol{x} = 1, \tag{4.106}$$

which guarantees that $\hat{\mathrm{s}}(t) = \mathrm{s}(t)$ in the absence of noise. Note that (4.106) is an instance of (4.97) if we consider $\boldsymbol{w}^{\mathrm{H}}$ and $\boldsymbol{x}$ in place of H and X, respectively. To account for the noise, we require the MSE

$$\mathbb{E}\left[|\mathrm{s}(t) - \hat{\mathrm{s}}(t)|^2\right] = \mathbb{E}\left[|\mathrm{s}(t) - \boldsymbol{w}^{\mathrm{H}}\mathbf{u}(t)|^2\right],$$

to be minimized. However,

$$\mathrm{s}(t) - \boldsymbol{w}^{\mathrm{H}}\mathbf{u}(t) = \mathrm{s}(t) - \boldsymbol{w}^{\mathrm{H}}\left(\boldsymbol{x}\mathrm{s}(t) + \boldsymbol{\eta}(t)\right) = -\boldsymbol{w}^{\mathrm{H}}\boldsymbol{\eta}(t).$$

Hence, the optimal $\boldsymbol{w}_*$ results from the following constrained task

$$\boldsymbol{w}_* := \arg\min_{w}(\boldsymbol{w}^H \Sigma_\eta \boldsymbol{w})$$

$$\text{s.t.} \quad \boldsymbol{w}^H\boldsymbol{x} = 1, \tag{4.107}$$

which is an instance of (4.99) and the solution is given by (4.100); adapting it to the current notation and to its complex-valued formulation, we get

$$\boxed{\boldsymbol{w}_*^{\mathrm{H}} = \frac{\boldsymbol{x}^{\mathrm{H}}\Sigma_\eta^{-1}}{\boldsymbol{x}^{\mathrm{H}}\Sigma_\eta^{-1}\boldsymbol{x}},} \tag{4.108}$$

and

$$\hat{\mathrm{s}}(t) = \boldsymbol{w}_*^{\mathrm{H}}\mathbf{u}(t) = \frac{\boldsymbol{x}^{\mathrm{H}}\Sigma_\eta^{-1}\mathbf{u}(t)}{\boldsymbol{x}^{\mathrm{H}}\Sigma_\eta^{-1}\boldsymbol{x}}. \tag{4.109}$$

The minimum MSE is equal to

$$\text{MSE}(\boldsymbol{w}_*) = \frac{1}{\boldsymbol{x}^H \Sigma_\eta^{-1} \boldsymbol{x}}. \tag{4.110}$$

An alternative formulation for the cost function in order to estimate the weights of the beamformer, which is often met in practice, builds upon the goal to minimize the output power, subject to the same constraint as before,

$$\begin{aligned} \boldsymbol{w}_* &:= \arg\min_{\boldsymbol{w}} \mathbb{E}\left[|\boldsymbol{w}^H \mathbf{u}(n)|^2\right], \\ \text{s.t.} \quad & \boldsymbol{w}^H \boldsymbol{x} = 1, \end{aligned}$$

or equivalently

$$\begin{aligned} \boldsymbol{w}_* &:= \arg\min_{\boldsymbol{w}} \boldsymbol{w}^H \Sigma_u \boldsymbol{w}, \\ \text{s.t.} \quad & \boldsymbol{w}^H \boldsymbol{x} = 1. \end{aligned} \tag{4.111}$$

This time, the beamformer is pushed to reduce its output signal, which, due to the presence of the constraint, is equivalent with optimally minimizing the contributions originating from the noise as well as from all other interference sources impinging on the array from different, to ϕ, directions. The resulting solution of (4.111) is obviously the same as (4.109) and (4.110) if one replaces Σ_η with Σ_u.

This type of linearly constraint task is known as *linearly constrained minimum variance* (LMV) or *Capon* beamforming or *minimum variance distortionless response (MVDR) beamforming*. For a concise introduction to beamforming, see, e.g., [48].

Widely linear versions for the beamforming task have also been proposed, e.g., [10, 32] (Problem 4.21).

Figure 4.20 shows the resulting beam-pattern as a function of the angle ϕ. The desired angle for designing the optimal set of weights in (4.108) is $\phi = \pi$. The number of antenna elements is $l = 10$, the spacing has been chosen as $\frac{\Delta x}{\lambda} = 0.5$, and the noise covariance matrix as $0.1I$. The beam-pattern amplitude is in dBs, meaning the vertical axis shows $20\log_{10}(|\boldsymbol{w}_*^{\mathrm{H}}\boldsymbol{x}(\phi)|)$. Thus, any signal arriving from directions ϕ, not close to $\phi = \pi$, will be absorbed. The main beam can become sharper if more elements are used.

4.10 TIME-VARYING STATISTICS: KALMAN FILTERING

So far, our discussion about the linear estimation task has been limited to stationary environments, where the statistical properties of the involved random variables are assumed to be invariant with time. However, very often in practice this is not the case and the statistical properties may be different at different time instants. As a matter of fact, a large effort in the subsequent chapters will be devoted to studying the estimation task under time-varying environments.

Rudolf Kalman is the third scientist, after Wiener and Kolmogorov, whose significant contributions laid the foundations of estimation theory. Kalman is Hungarian-born and emigrated to the United States. He is the father of what is today known as system theory based on the state-space formulation, as opposed to the more limited input-output description of systems.

In 1960, in two seminal papers, Kalman proposed the celebrated Kalman filter, which exploits the state-space formulation in order to accommodate in an elegant way time-varying dynamics [18, 19].

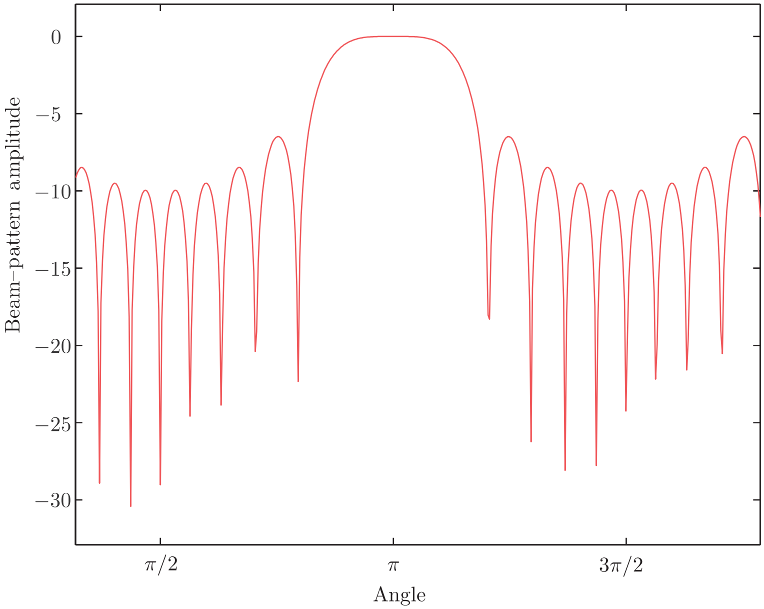


FIGURE 4.20

The amplitude beam-pattern, in dBs, as a function of angle, ϕ with respect to the planar array.

We will derive the basic recursions of the Kalman filter in the general context of two jointly distributed random vectors $\mathbf{y}$, $\mathbf{x}$. The task is to estimate the values of $\mathbf{x}$ given observations on $\mathbf{y}$. Let $\mathbf{y}$ and $\mathbf{x}$ be linearly related via the following set of recursions

$$\mathbf{x}_n = F_n \mathbf{x}_{n-1} + \boldsymbol{\eta}_n, \quad n \geq 0, \quad \text{State Equation,} \tag{4.112}$$

$$\mathbf{y}_n = H_n \mathbf{x}_n + \mathbf{v}_n, \quad n \geq 0, \quad \text{Output Equation,} \tag{4.113}$$

where $\boldsymbol{\eta}_n, \mathbf{x}_n \in \mathbb{R}^l$, $\mathbf{v}_n, \mathbf{y}_n \in \mathbb{R}^k$. The vector $\mathbf{x}_n$ is known as the *state* of the system at time n and $\mathbf{y}_n$ is the output, which is the vector that can be observed (measured); $\boldsymbol{\eta}_n$ and $\mathbf{v}_n$ are the noise vectors, known as *process* noise and *measurement* noise, respectively. Matrices F_n and H_n are of appropriate dimensions and they are assumed to be known. Observe that the so-called *state equation* provides the information related to the time-varying dynamics of the corresponding system. It turns out that a large number of real-world tasks can be brought into the form of (4.112) and (4.113). The model is known as the *state-space* model for $\mathbf{y}_n$. In order to derive the time-varying estimator, $\hat{\mathbf{x}}_n$, given the measured values of $\mathbf{y}_n$, the following assumptions will be adopted:

- $\mathbb{E}[\boldsymbol{\eta}_n \boldsymbol{\eta}_n^T] = Q_n$, $\mathbb{E}[\boldsymbol{\eta}_n \boldsymbol{\eta}_m^T] = O$, $n \neq m$,
- $\mathbb{E}[\mathbf{v}_n \mathbf{v}_n^T] = R_n$, $\mathbb{E}[\mathbf{v}_n \mathbf{v}_m^T] = O$, $n \neq m$,
- $\mathbb{E}[\boldsymbol{\eta}_n \mathbf{v}_m^T] = O$, $\forall n, m$,
- $\mathbb{E}[\boldsymbol{\eta}_n] = \mathbb{E}[\mathbf{v}_n] = \mathbf{0}$, $\forall n$,

where O denotes a matrix with zero elements. That is, $\boldsymbol{\eta}_n, \mathbf{v}_n$ are uncorrelated; moreover, noise vectors at different time instants are also considered uncorrelated. Versions where some of these conditions are relaxed are also available. The respective covariance matrices, Q_n, R_n, are assumed to be known.

The development of the time-varying estimation task evolves around two types of estimators for the state variables:

- The first one is denoted as

  $$\hat{\mathbf{x}}_{n|n-1},$$

  and it is based on all information that has been received up to and including time instant $n-1$; in other words, the obtained observations of $\mathbf{y}_0, \mathbf{y}_1, \ldots, \mathbf{y}_{n-1}$. This is known as the a priori or *prior* estimator.
- The second estimator at time n is known as the *posterior* one, it is denoted as

  $$\hat{\mathbf{x}}_{n|n},$$

  and it is computed by updating $\hat{\mathbf{x}}_{n|n-1}$ after $\mathbf{y}_n$ has been observed.

For the development of the algorithm, assume that at time $n-1$ all required information is available; that is, the value of the posterior estimator as well the respective error covariance matrix

$$\hat{\mathbf{x}}_{n-1|n-1}, \quad P_{n-1|n-1} := \mathbb{E}[\mathbf{e}_{n-1|n-1}\mathbf{e}^{\mathrm{T}}_{n-1|n-1}],$$

where

$$\mathbf{e}_{n-1|n-1} := \mathbf{x}_{n-1} - \hat{\mathbf{x}}_{n-1|n-1}.$$

Step 1: Using $\hat{\mathbf{x}}_{n-1|n-1}$, predict $\hat{\mathbf{x}}_{n|n-1}$ using the state equation; that is,

$$\hat{\mathbf{x}}_{n|n-1} = F_n \hat{\mathbf{x}}_{n-1|n-1}. \tag{4.114}$$

In other words, ignore the contribution from the noise. This is natural, because prediction cannot involve the unobserved variables.

Step 2: Obtain the respective error covariance matrix,

$$P_{n|n-1} = \mathbb{E}[(\mathbf{x}_n - \hat{\mathbf{x}}_{n|n-1})(\mathbf{x}_n - \hat{\mathbf{x}}_{n|n-1})^{\mathrm{T}}]. \tag{4.115}$$

However,

$$\begin{aligned} \mathbf{e}_{n|n-1} := \mathbf{x}_n - \hat{\mathbf{x}}_{n|n-1} &= F_n\mathbf{x}_{n-1} + \boldsymbol{\eta}_n - F_n\hat{\mathbf{x}}_{n-1|n-1} \\ &= F_n\mathbf{e}_{n-1|n-1} + \boldsymbol{\eta}_n. \end{aligned} \tag{4.116}$$

Combining (4.115) and (4.116), it is straightforward to see that

$$P_{n|n-1} = F_n P_{n-1|n-1} F_n^{\mathrm{T}} + Q_n. \tag{4.117}$$

Step 3: Update $\hat{\mathbf{x}}_{n|n-1}$. To this end, adopt the following recursion

$$\hat{\mathbf{x}}_{n|n} = \hat{\mathbf{x}}_{n|n-1} + K_n\mathbf{e}_n, \tag{4.118}$$

where

$$\mathbf{e}_n := \mathbf{y}_n - H_n\hat{\mathbf{x}}_{n|n-1}. \tag{4.119}$$

This time update recursion, once the observations for $\mathbf{y}_n$ have been received, has a form that we will meet over and over again in this book. The "new" (posterior) estimate is equal to the "old" (prior) one, that is based on *the past history, plus a correction term*; the latter is proportional to the error $\mathbf{e}_n$ in predicting the newly arrived observations vector and its prediction based on the "old" estimate. Matrix K_n, known as the *Kalman gain*, controls the amount of correction and its value is computed so as to minimize the mean-square error; in other words,

$$J(K_n) := \mathbb{E}[\mathbf{e}_{n|n}^T \mathbf{e}_{n|n}] = \text{trace}\{P_{n|n}\}, \tag{4.120}$$

where

$$P_{n|n} = \mathbb{E}[\mathbf{e}_{n|n} \mathbf{e}_{n|n}^T], \tag{4.121}$$

and

$$\mathbf{e}_{n|n} := \mathbf{x}_n - \hat{\mathbf{x}}_{n|n}.$$

It can be shown that, the optimal Kalman gain is equal to (Problem 4.22)

$$K_n = P_{n|n-1} H_n^T S_n^{-1}, \tag{4.122}$$

where

$$S_n = R_n + H_n P_{n|n-1} H_n^T. \tag{4.123}$$

Step 4: The final recursion that is now needed in order to complete the scheme is that for the update of $P_{n|n}$. Combining the definitions in (4.119) and (4.121) with (4.118), the following results (Problem 4.23),

$$P_{n|n} = P_{n|n-1} - K_n H_n P_{n|n-1}. \tag{4.124}$$

The algorithm has now been derived. All that is now needed is to select the initial conditions, which are chosen such as

$$\hat{\boldsymbol{x}}_{1|0} = \mathbb{E}[\mathbf{x}_1] \tag{4.125}$$

$$P_{1|0} = \mathbb{E}\left[(\mathbf{x}_1 - \hat{\boldsymbol{x}}_{1|0})(\mathbf{x}_1 - \hat{\boldsymbol{x}}_{1|0})^T\right] = \Pi_0, \tag{4.126}$$

for some initial guess Π_0. The Kalman algorithm is summarized in Algorithm 4.2.

Algorithm 4.2 (Kalman filtering).

```
• Input: $F_n, H_n, Q_n, R_n, \boldsymbol{y}_n, n = 1, 2, \ldots$
• Initialization:
    • $\hat{\boldsymbol{x}}_{1|0} = \mathbb{E}[\mathbf{x}_1]$
    • $P_{1|0} = \Pi_0$
• For $n = 1, 2, \ldots,$ Do
    • $S_n = R_n + H_n P_{n|n-1} H_n^T$
    • $K_n = P_{n|n-1} H_n^T S_n^{-1}$
    • $\hat{\boldsymbol{x}}_{n|n} = \hat{\boldsymbol{x}}_{n|n-1} + K_n(\boldsymbol{y}_n - H_n \hat{\boldsymbol{x}}_{n|n-1})$
    • $P_{n|n} = P_{n|n-1} - K_n H_n P_{n|n-1}$
    • $\hat{\boldsymbol{x}}_{n+1|n} = F_{n+1} \hat{\boldsymbol{x}}_{n|n}$
    • $P_{n+1|n} = F_{n+1} P_{n|n} F_{n+1}^T + Q_{n+1}$
• End For
```

For complex-valued variables, transposition is replaced by the Hermitian operation.

Remarks 4.7.

- Besides the previously derived basic scheme, there are a number of variants. Although, in theory, they are all equivalent, their practical implementation may lead to different performance. Observe that $P_{n|n}$ is computed as the difference of two positive definite matrices; this may lead to obtain a $P_{n|n}$ that is not positve definite, due to numerical errors. This can cause the algorithm to diverge. A popular alternative is the so-called *information filtering* scheme, which propagates the inverse state-error covariance matrices, $P_{n|n}^{-1}$, $P_{n|n-1}^{-1}$ [20]. In contrast, the scheme in Algorithm 4.2 is known as the *covariance* Kalman algorithm (Problem 4.24).

  To cope with the numerical stability issues, a family of algorithms propagates the factors of $P_{n|n}$ (or $P_{n|n}^{-1}$), resulting from the respective Cholesky factorization [5, 40].
- There are different approaches to arrive at the Kalman filtering recursions. An alternative derivation is based on the orthogonality principle applied to the so-called *innovations process* associated with the observation sequence, so that

  $$\boldsymbol{\epsilon}(n) = \mathbf{y}_n - \hat{\mathbf{y}}_{n|1:n-1},$$

  where $\hat{\mathbf{y}}_{n|1:n-1}$ is the prediction based on the past observations history [17]. In Chapter 17, we are going to rederive the Kalman recursions looking at it as a Bayesian network.
- Kalman filtering is a generalization of the optimal mean-square linear filtering. It can be shown that when the involved processes are stationary, Kalman filter converges in its steady-state to our familiar normal equations [31].
- *Extended Kalman filters*. In (4.112) and (4.113), both the state as well as the output equations have a linear dependence on the state vector $\mathbf{x}_n$. Kalman filtering, in a more general formulation, can be cast as

  $$\mathbf{x}_n = \boldsymbol{f}_n(\mathbf{x}_{n-1}) + \boldsymbol{\eta}_n,$$

  $$\mathbf{y}_n = \boldsymbol{h}_n(\mathbf{x}_n) + \mathbf{v}_n,$$

  where $\boldsymbol{f}_n$ and $\boldsymbol{h}_n$ are nonlinear vector functions. In the extended Kalman filtering (EKF), the idea is to *linearize* the functions $\boldsymbol{h}_n(\cdot)$ and $\boldsymbol{f}_n(\cdot)$, at each time instant, via their Taylor series expansions and keep the linear term only, so that

  $$\boldsymbol{F}_n = \frac{\partial \boldsymbol{f}_n(\boldsymbol{x}_n)}{\partial \boldsymbol{x}_n}\bigg|_{\boldsymbol{x}_n=\hat{\boldsymbol{x}}_{n-1|n-1}},$$

  $$\boldsymbol{H}_n = \frac{\partial \boldsymbol{h}_n(\boldsymbol{x}_n)}{\partial \boldsymbol{x}_n}\bigg|_{\boldsymbol{x}_n=\hat{\boldsymbol{x}}_{n|n-1}},$$

  and then proceed by using the updates derived for the linear case.

  By its very definition, the EKF is suboptimal and often in practice one may face divergence of the algorithm; in general, it must be stated that its practical implementation needs to be done with care. Having said that, it must be pointed out that it is heavily used in a number of practical systems.

  Unscented Kalman Filters is an alternative way to cope with the nonlinearity, and the main idea springs from probabilistic arguments. A set of points are deterministically selected from a Gaussian approximation, of $p(\boldsymbol{x}_n|\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n)$; these points are propagated through the nonlinearities and estimates of the mean values and covariances are obtained [15]. Particle filtering, to be discussed in Chapter 17, is another powerful and popular approach to deal with nonlinear state-space models via probabilistic arguments.

More recently, extensions of Kalman filtering in reproducing kernel Hilbert spaces offers an alternative approach to deal with nonlinearities [52].

A number of Kalman filtering versions for distributed learning (Chapter 5) have appeared in, e.g., [9, 23, 33, 43]. In the latter of the references, subspace learning methods are utilized in the prediction stage associated with the state variables.

- The literature on Kalman filtering is huge, especially when applications are concerned. The interested reader may consult more specialized texts, for example, [4, 8, 17] and the references therein.

Example 4.4 (Autoregressive Process Estimation)**.** Let us consider an AR process (Chapter 2) of order l, represented as

$$\mathrm{x}_n = -\sum_{i=1}^{l} a_i \mathrm{x}_{n-i} + \eta_n, \tag{4.127}$$

where η_n is a white noise sequence of variance σ_η^2. Our task is to obtain an estimate $\hat{x}_n$ of x_n, having observed a noisy version of it, y_n. The corresponding random variables are related as

$$\mathrm{y}_n = \mathrm{x}_n + \mathrm{v}_n. \tag{4.128}$$

To this end, the Kalman filtering formulation will be used. Note that the MSE linear estimation, presented in Section 4.9, cannot be used here. As we have already discussed in Chapter 2, an AR process is asymptotically stationary; for finite time samples, the initial conditions at time $n = 0$ are "remembered" by the process and the respective (second) order statistics are time dependent, hence it is a nonstationary process. However, Kalman filtering is specially suited for such cases.

Let us rewrite (4.127) and (4.128) as

$$\begin{bmatrix} \mathrm{x}_n \\ \mathrm{x}_{n-1} \\ \mathrm{x}_{n-2} \\ \vdots \\ \mathrm{x}_{n-l+1} \end{bmatrix} = \begin{bmatrix} -a_1 & -a_2 & \cdots & -a_{l-1} & -a_l \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathrm{x}_{n-1} \\ \mathrm{x}_{n-2} \\ \mathrm{x}_{n-3} \\ \vdots \\ \mathrm{x}_{n-l} \end{bmatrix} + \begin{bmatrix} \eta_n \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

$$\mathrm{y}_n = \begin{bmatrix} 1 & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} \mathrm{x}_n \\ \vdots \\ \mathrm{x}_{n-l+1} \end{bmatrix} + \mathrm{v}_n$$

or

$$\mathbf{x}_n = F\mathbf{x}_{n-1} + \boldsymbol{\eta}, \tag{4.129}$$

$$\mathrm{y}_n = H\mathbf{x}_n + \mathrm{v}_n, \tag{4.130}$$

where the definitions of $F_n \equiv F$ and $H_n \equiv H$ are obvious and

$$Q_n = \begin{bmatrix} \sigma_n^2 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \end{bmatrix}, \quad R_n = \sigma_\mathrm{v}^2 \quad \text{(scalar)}.$$

Figure 4.21a shows the values of a specific realization y_n, and Figure 4.21b the corresponding realization of the AR(2) (red) together with the predicted Kalman filter sequence $\hat{x}_n$. Observe that the match is

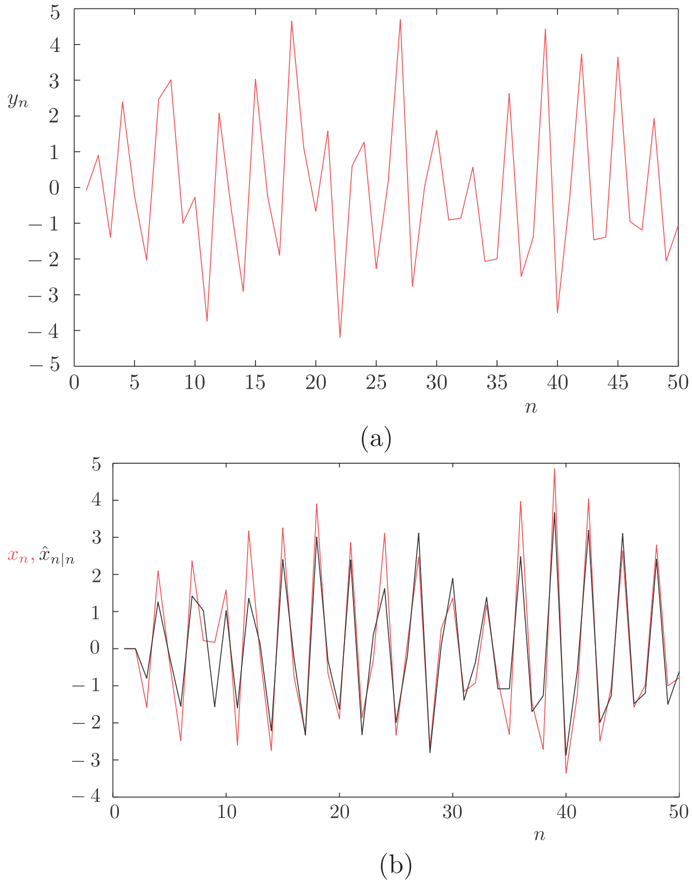


FIGURE 4.21

(a) A realization of the observation sequence, y_n, which is used by the Kalman filter to obtain the predictions of the state variable. (b) The AR process (state variable) in red together with the predicted by the Kalman filter sequence (gray), for Example 4.4. The Kalman filter has removed the effect of the noise v_n.

very good. For the generation of the AR process we used $l = 2, \alpha_1 = 0.95, \alpha_2 = 0.9, \sigma_\eta^2 = 0.5$. For the Kalman filter output noise, $\sigma_v^2 = 1$.

PROBLEMS

4.1 Show that the set of equations

$$\boldsymbol{\Sigma\theta} = \boldsymbol{p}$$

has a unique solution if $\Sigma > 0$ and infinite many if Σ is singular.

4.2 Show that the set of equations

$$\Sigma\boldsymbol{\theta} = \boldsymbol{p}$$

always has a solution.

4.3 Show that the shape of the isovalue contours of the mean-square error ($J(\boldsymbol{\theta})$) surface

$$J(\boldsymbol{\theta}) = J(\boldsymbol{\theta}_*) + (\boldsymbol{\theta} - \boldsymbol{\theta}_*)^{\mathrm{T}}\Sigma(\boldsymbol{\theta} - \boldsymbol{\theta}_*)$$

are ellipses whose axes depend on the eigenstructure of Σ.
Hint. Assume that Σ has discrete eigenvalues.

4.4 Prove that if the true relation between the input x and the true output y is linear, meaning

$$\mathrm{y} = \boldsymbol{\theta}_o^{\mathrm{T}}\mathbf{x} + \mathrm{v}_n, \quad \boldsymbol{\theta}_o \in \mathbb{R}^l,$$

where v is independent of x, then, the optimal MSE estimate $\boldsymbol{\theta}_*$ satisfies

$$\boldsymbol{\theta}_* = \boldsymbol{\theta}_o.$$

4.5 Show that if

$$\mathrm{y} = \boldsymbol{\theta}_o^{\mathrm{T}}\mathbf{x} + \mathrm{v}, \quad \boldsymbol{\theta}_o \in \mathbb{R}^k,$$

where v is independent of x, then the optimal MSE $\boldsymbol{\theta}_* \in \mathbb{R}^l$, $l < k$ is equal to the top l components of $\boldsymbol{\theta}_o$, if the components of $\mathbf{x}$ are uncorrelated.

4.6 Derive the normal equations by minimizing the cost in (4.15).
Hint. Express the cost in terms of the real part $\boldsymbol{\theta}_r$ and its imaginary part $\boldsymbol{\theta}_i$ of $\boldsymbol{\theta}$ and optimize with respect to $\boldsymbol{\theta}_r, \boldsymbol{\theta}_i$.

4.7 Consider the multichannel filtering task

$$\hat{\mathbf{y}} = \begin{bmatrix} \hat{\mathrm{y}}_r \\ \hat{\mathrm{y}}_i \end{bmatrix} = \Theta \begin{bmatrix} \mathbf{x}_r \\ \mathbf{x}_i \end{bmatrix}.$$

Estimate Θ so that to minimize the error norm:

$$\mathbb{E}[||\mathbf{y} - \hat{\mathbf{y}}||^2].$$

4.8 Show that (4.34) is the same as (4.25).

4.9 Show that the MSE achieved by a linear complex-valued estimator is always larger than that obtained by a widely linear one. Equality is achieved only under the circularity conditions.

4.10 Show that under the second-order circularity assumption, the conditions in (4.39) hold true.

4.11 Show that if

$$f : \mathbb{C} \longrightarrow \mathbb{R},$$

then the Cauchy-Riemann conditions are violated.

4.12 Derive the optimality condition in (4.45).

4.13 Show Eqs. (4.50) and (4.51).

4.14 Derive the normal equations for Example (4.2).

4.15 The input to the channel is a white noise sequence s_n of variance σ_s^2. The output of the channel is the AR process

$$\mathrm{y}_n = a_1\mathrm{y}_{n-1} + \mathrm{s}_n. \tag{4.131}$$

The channel also adds white noise η_n of variance σ_η^2. Design an optimal equalizer of order two, which at its output recovers an approximation of s_{n-L}. Sometimes, this equalization task is also known as whitening, because in this case the action of the equalizer is to "whiten" the AR process.

4.16 Show that the forward and backward MSE optimal predictors are conjugate reverse of each other.

4.17 Show that the MSE prediction errors ($\alpha_m^f = \alpha_m^b$) are updated according to the recursion

$$\alpha_m^b = \alpha_{m-1}^b (1 - |\kappa_{m-1}|^2).$$

4.18 Derive the BLUE for the Gauss-Markov theorem.

4.19 Show that the mean-square error (which in this case coincides with the variance of the estimator) of any linear unbiased estimator is higher than that associated with the BLUE.

4.20 Show that if Σ_η is positive definite, then $X^{\mathrm{T}} \Sigma_\eta^{-1} X$ is also positive definite if X is full rank.

4.21 Derive a MSE optimal linearly constrained widely linear beamformer.

4.22 Prove that the Kalman gain that minimizes the error covariance matrix

$$P_{n|n} = \mathbb{E}[(\mathbf{x}_n - \hat{\mathbf{x}}_{n|n})(\mathbf{x}_n - \hat{\mathbf{x}}_{n|n})^{\mathrm{T}}]$$

is given by

$$K_n = P_{n|n-1} H_n^{\mathrm{H}} (R_n + H_n P_{n|n-1} H_n^{\mathrm{T}})^{-1}.$$

Hint. Use the following formulas

$$\frac{\partial \operatorname{trace}\{AB\}}{\partial A} = B^{\mathrm{T}} \ (AB \text{ a square matrix})$$

$$\frac{\partial \operatorname{trace}\{ACA^{\mathrm{T}}\}}{\partial A} = 2AC, \ (C = C^{\mathrm{T}}).$$

4.23 Show that in Kalman filtering, the prior and posterior error covariance matrices are related as

$$P_{n|n} = P_{n|n-1} - K_n H_n P_{n|n-1}.$$

4.24 Derive the Kalman algorithm in terms of the inverse state-error covariance matrices, $P_{n|n}^{-1}$. In statistics, the inverse error covariance matrix is related to Fisher's information matrix: hence, the name of the scheme.

MATLAB Exercises

4.25 Consider the image deblurring task described in Section 4.6.

- Download the "boat" image from Waterloo's Image repository.[8] Alternatively, you may use any grayscale image of your choice. You can load the image into MATLAB's memory using the "imread" function (also, you may want to apply the function "im2double" to get an array consisting of doubles).
- Create a blurring point spread function (PSF) using MATLAB's command "fspecial." For example, you can write

[8] http://links.uwaterloo.ca/.

```
PSF = fspecial('motion',20,45);
```

The blurring effect is produced using the "imfilter" function

```
J = imfilter(I,PSF,'conv', 'circ');
```

where I is the original image.

- Add some white gaussian noise to the image using MATLAB's function "imnoise," as follows:

```
J = imnoise(J, 'gaussian', noise_mean, noise_var);
```

Use a small value of noise variance, such as 10^{-6}.

- To perform the deblurring, you need to employ the "deconvwnr" function. For example, if J is the array that contains the blurred image (with the noise) and PSF is the point spread function that produced the blurring, then the command

```
K = deconvwnr(J, PSF, C);
```

returns the deblurred image K, provided that the choice of C is reasonable. As a first attempt, select $C = 10^{-4}$. Use various values for C of your choice. Comment on the results.

4.26 Consider the noise cancelation task described in Example 4.1. Write the necessary code to solve the problem using MATLAB according to the following steps:

(a) Create 5000 data samples of the signal $s_n = \cos(\omega_0 n)$, for $\omega_0 = 2 \times 10^{-3}\pi$.

(b) Create 5000 data samples of the AR process $v_1(n) = a_1 \cdot v_1(n-1) + \eta_n$ (initializing at zero), where η_n represents zero mean Gaussian noise with variance $\sigma_\eta^2 = 0.0025$ and $a_1 = 0.8$.

(c) Add the two sequences (i.e., $d_n = s_n + v_1(n)$) and plot the result. This represents the contaminated signal.

(d) Create 5000 data samples of the AR process $v_2(n) = a_2 v_2(n-1) + \eta_n$ (initializing at zero), where η_n represents the same noise sequence and $a_2 = 0.75$.

(e) Solve for the optimum (in the MSE sense) $\boldsymbol{w} = [w_0, w_1]^T$. Create the sequence of the restored signal $\hat{s}_n = d_n - w_0 v_2(n) - w_1 v_2(n-1)$ and plot the result.

(f) Repeat steps 26b-26e using $a_2 = 0.9, 0.8, 0.7, 0.6, 0.5, 0.3$. Comment on the results.

(g) Repeat steps 26b-26e using $\sigma_\eta^2 = 0.01, 0.05, 0.1, 0.2, 0.5$, for $a_2 = 0.9, 0.8, 0.7, 0.6, 0.5, 0.3$. Comment on the results.

4.27 Consider the channel equalization task described in Example 4.2. Write the necessary code to solve the problem using MATLAB according to the following steps:

(a) Create a signal s_n consisting of 50 equiprobable ± 1 samples. Plot the result using MATLAB's function "stem."

(b) Create the sequence $u_n = 0.5 s_n + s_{n-1} + \eta_n$, where η_n denotes zero mean Gaussian noise with $\sigma_\eta^2 = 0.01$. Plot the result with "stem."

(c) Find the optimal $\boldsymbol{w}_* = [w_0, w_1, w_2]^T$, solving the normal equations.

(d) Construct the sequence of the reconstructed signal $\hat{s}_n = \text{sgn}(w_0 u_n + w_1 u_{n-1} + w_2 u_{n-2})$. Plot the result with "stem" using red color for the correctly reconstructed values (i.e., those that satisfy $s_n = \hat{s}_n$) and black color for errors.

(e) Repeat steps 27b-27d using different noise levels for σ_η^2. Comment on the results.

4.28 Consider the autoregressive process estimation task described in Example 4.4. Write the necessary code to solve the problem using MATLAB according to the following steps:

(a) Create 500 samples of the AR sequence $x_n = -a_1 x_{n-1} - a_2 x_{n-2} + \eta_n$ (initializing at zeros), where $a_1 = 0.2$, $a_2 = 0.1$, and η_n denotes zero mean Gaussian noise with $\sigma_n^2 = 0.5$.

(b) Create the sequence $y_n = x_n + v_n$, where v_n denotes zero mean Gaussian noise with $\sigma_v^2 = 1$.

(c) Implement the Kalman filtering algorithm as described in Algorithm 4.2, using y_n as inputs and the matrices F, H, Q, R as described in Example 4.4. To initialize the algorithm, you can use $\hat{\boldsymbol{x}}_{1|0} = [0, 0]^T$ and $P_{1|0} = 0.1 \cdot I_2$. Plot the predicted values $\hat{x}_n$ versus the original sequence x_n. Play with the values of the different parameters and comment on the obtained results.

REFERENCES

[1] T. Adali, V.D. Calhoun, Complex ICA of brain imaging data, IEEE Signal Process. Mag. 24(5) (2007) 136-139.

[2] T. Adali, H. Li, Complex-valued adaptive signal processing, in: T. Adali, S. Haykin (Eds.), Adaptive Signal Processing: Next Generation Solutions, John Wiley, 2010.

[3] T. Adali, P. Schreier, Optimization and estimation of complex-valued signals: theory and applications in filtering and blind source separation, IEEE Signal Process. Mag. 31(5) (2014) 112-128.

[4] B.D.O. Anderson, J.B. Moore, Optimal Filtering, Prentice Hall, Englewood Cliffs, NJ, 1979.

[5] G.J. Bierman, Factorization Methods for Discrete Sequential Estimation, Academic Press, New York, 1977.

[6] P. Bouboulis, S. Theodoridis, Extension of Wirtinger's calculus to Reproducing kernel Hilbert spaces and the complex kernel LMS, IEEE Trans. Signal Process. 53(3) (2011) 964-978.

[7] D.H. Brandwood, A complex gradient operator and its application in adaptive array theory, IEEE Proc. 130(1) (1983) 11-16.

[8] R.G. Brown, P.V.C. Hwang, Introduction to Random Signals and Applied Kalman Filtering, second ed., John Wiley Sons, Inc., 1992.

[9] F.S. Cattivelli, A.H. Sayed, Diffusion strategies for distributed Kalman filtering and smoothing, IEEE Trans. Automat. Control 55(9) (2010) 2069-2084.

[10] P. Chevalier, J.P. Delmas, A. Oukaci, Optimal widely linear MVDR beamforming for noncircular signals, in: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, 2009, pp. 3573-3576.

[11] P. Delsarte, Y. Genin, The split Levinson algorithm, IEEE Trans. Signal Process. 34 (1986) 470-478.

[12] J. Dourbin, The fiting of time series models, Rev. Int. Stat. Inst. 28 (1960) 233-244.

[13] Y.C. Eldar. Minimax, MSE estimation of deterministic parameters with noise covariance uncertainties, IEEE Trans. Signal Process. 54 (2006) 138-145.

[14] R.C. Gonzalez, R.E. Woods, Digital Image Processing, Addison-Wesley, 1993.

[15] S. Julier, A skewed approach to filtering, Proc. SPIE 3373 (1998) 271-282.

[16] T. Kailath, An innovations approach to least-squares estimation: Part 1. Linear filtering in additive white noise, IEEE Trans. Automat. Control AC-13 (1968) 646-655.

[17] T. Kailath, A.H. Sayed, B. Hassibi, Linear Estimation, Prentice Hall, Englewood Cliffs, 2000.

[18] R.E. Kalman, A new approach to linear filtering and prediction problems, Trans. ASME J. Basic Eng. 82 (1960) 34-45.

[19] R.E. Kalman, R.S. Bucy, New results in linear filtering and prediction theory, Trans. ASME J. Basic Eng. 83 (1961) 95-107.

[20] P.G. Kaminski, A.E. Bryson, S.F. Schmidt, Discrete square root filtering: A survey, IEEE Transactions on Automatic Control 16 (1971) 727-735.

[21] N. Kalouptsidis, S. Theodoridis, Parallel implementation of efficient LS algorithms for filtering and prediction, IEEE Trans. Acoust. Speech Signal Process. 35 (1987) 1565-1569.

[22] N. Kalouptsidis, S. Theodoridis (Eds.), Adaptive System Identification and Signal Processing Algorithms, Prentice Hall, 1993.

[23] U.A. Khan, J. Moura, Distributing the Kalman filter for large-scale systems, IEEE Trans. Signal Process. 56(10) (2008) 4919-4935.

[24] A.N. Kolmogorov, Stationary sequences in Hilbert spaces, Bull. Math. Univ. Moscow 2 (1941) (in Russian).

[25] K. Kreutz-Delgado, The complex Gradient Operator and the ℂℝ-Calculus, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.86.6515&rep=rep1&type=pdf, 2006.

[26] N. Levinson, The Wiener error criterion in filter design and prediction, J. Math. Phys. 25 (1947) 261-278.

[27] H. Li, T. Adali, Optimization in the complex domain for nonlinear adaptive filtering, in: Proceedings, 33rd Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, 2006, pp. 263-267.

[28] X.-L. Li, T. Adali, Complex-valued linear and widely linear filtering using MSE and Gaussian entropy, IEEE Trans. Signal Process. 60 (2012) 5672-5684.

[29] D.J.C. MacKay, Information Theory, Inference, and Learning Algorithms, Cambridge University Press, 2003.

[30] D. Mandic, V.S.L. Guh, Complex Valued Nonlinear Adaptive Filters, John Wiley, 2009.

[31] J.M. Mendel, Lessons in Digital Estimation Theory, Prentice Hall, Englewood Cliffs, NJ, 1995.

[32] T. McWhorter, P. Schreier, Widely linear beamforming, in: Proceedings 37th Asilomar Conference on Signals, Systems, Computers, Pacific Grove, CA, 1993, pp. 759.

[33] P.V. Overschee, B.D. Moor, Subspace Identification for Linear Systems: Theory, Implementation, Applications, Kluwer Academic Publishers, 1996.

[34] M. Petrou, C. Petrou, Image Processing: The fundamentals, second ed., John Wiley, 2010.

[35] B. Picinbono, On circularity, IEEE Trans. Signal Process. 42(12) (1994) 3473-3482.

[36] B. Picinbono, P. Chevalier, Widely linear estimation with complex data, IEEE Trans. Signal Process. 43(8) (1995) 2030-2033.

[37] B. Picinbono, Random Signals and Systems, Prentice Hall, 1993.

[38] T. Piotrowski, I. Yamada, MV-PURE estimator: minimum-variance pseudo-unbiased reduced-rank estimator for linearly constrained ill-conditioned inverse problems, IEEE Trans. Signal Process. 56 (2008) 3408-3423.

[39] I.R. Porteous, Clifford Algebras and Classical Groups, Cambridge University Press, 1995.

[40] J.E. Potter, New statistical formulas, Space Guidance Analysis Memo, No 40, Instrumentation Laboratory, MIT, 1963.

[41] J. Proakis, Digital Communications, second ed., McGraw Hill, 1989.

[42] J.G. Proakis, D.G. Manolakis, Digital Signal Processing: Principles, Algorithms and Applications, second ed., MacMillan, 1992.

[43] O.-S. Reza, Distributed Kalman filtering for sensor networks, in: Proceedings IEEE Conference on Decision and Control, 2007, pp. 5492-5498.

[44] A.H. Sayed, Fundamentals of Adaptive Filtering, John Wiley, 2003.

[45] J. Schur, Über Potenzreihen, die im Innern des Einheitskreises beschränkt sind, J. Reine Angew. Math. 147 (1917) 205-232.

[46] K. Slavakis, P. Bouboulis, S. Theodoridis, Adaptive learning in complex reproducing kernel Hilbert spaces employing Wirtinger's Subgradients, IEEE Trans. Neural Networks Learn. Syst. 23(3) (2012) 425-438.

[47] G. Strang, Linear Algebra and its Applications, fourth ed., Hartcourt Brace Jovanovich, 2005.

[48] M. Viberg, Introduction to Array Processing, in: R. Chellappa, S. Theodoridis (Eds.), Academic Library in Signal Processing, vol. 3, Academic Press, 2014, pp. 463-499.

[49] N. Wiener, E. Hopf, Über eine klasse singulärer integralgleichungen, S.B. Preuss. Akad. Wiss, 1931, pp. 696-706.

[50] N. Wiener, Extrapolation, Interpolation and Smoothing of Stationary Time Series, MIT Press, Cambridge, MA, 1949.

[51] W. Wirtinger, Zur formalen theorie der funktionen von mehr komplexen veränderlichen, Math. Ann. 97 (1927) 357-375.

[52] P. Zhu, B. Chen, J.C. Principe, Learning nonlinear generative models of time series with a Kalman filter in RKHS, IEEE Trans. Signal Process. 62(1) (2014) 141-155.