Advanced Digital Signal Processing and Noise Reduction, Second Edition.
Saeed V. Vaseghi

ISBNs: 0-471-62692-9 (Hardback): 0-470-84162-1 (Electronic)

15

CHANNEL EQUALIZATION AND BLIND DECONVOLUTION

Blind deconvolution is the process of unravelling two unknown signals that have been convolved. An important application of blind deconvolution is in blind equalization for restoration of a signal distorted in transmission through a communication channel. Blind equalization has a wide range of applications, for example in digital telecommunications for removal of intersymbol interference, in speech recognition for removal of the effects of microphones and channels, in deblurring of distorted images, in dereverberation of acoustic recordings, in seismic data analysis, etc.

In practice, blind equalization is only feasible if some useful statistics of the channel input, and perhaps also of the channel itself, are available. The success of a blind equalization method depends on how much is known about the statistics of the channel input, and how useful this knowledge is in the channel identification and equalization process. This chapter begins with an introduction to the basic ideas of deconvolution and channel equalization. We study blind equalization based on the channel input power spectrum, equalization through separation of the input signal and channel response models, Bayesian equalization, nonlinear adaptive equalization for digital communication channels, and equalization of maximum-phase channels using higher-order statistics.

15.1 Introduction

In this chapter we consider the recovery of a signal distorted, in transmission through a channel, by a convolutional process and observed in additive noise. The process of recovery of a signal convolved with the impulse response of a communication channel, or a recording medium, is known as deconvolution or equalization. Figure 15.1 illustrates a typical model for a distorted and noisy signal, followed by an equalizer. Let $x(m)$, $n(m)$ and $y(m)$ denote the channel input, the channel noise and the observed channel output respectively. The channel input/output relation can be expressed as

$$y(m)=h[x(m)]+n(m) \tag{15.1}$$

where the function $h[\cdot]$ is the channel distortion. In general, the channel response may be time-varying and non-linear. In this chapter, it is assumed that the effects of a channel can be modelled using a stationary, or a slowly time-varying, linear transversal filter. For a linear transversal filter model of the channel, Equation (15.1) becomes

$$y(m)=\sum_{k=0}^{P-1} h_k(m)x(m-k)+n(m) \tag{15.2}$$

where $h_k(m)$ are the coefficients of a P^{th} order linear FIR filter model of the channel. For a time-invariant channel model, $h_k(m)=h_k$.

In the frequency domain, Equation (15.2) becomes

$$Y(f)=X(f)H(f)+N(f) \tag{15.3}$$

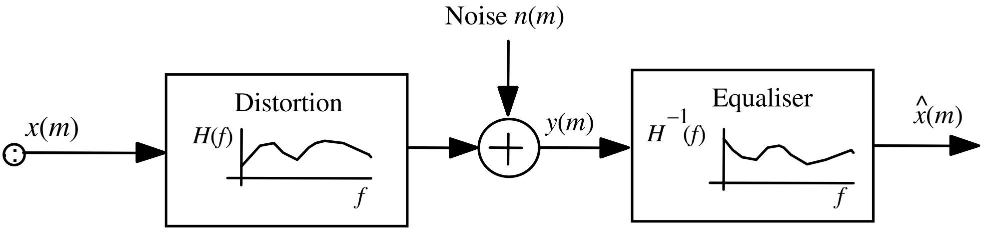

Figure 15.1 Illustration of a channel distortion model followed by an equalizer.

where $Y(f)$, $X(f)$, $H(f)$ and $N(f)$ are the frequency spectra of the channel output, the channel input, the channel response and the additive noise respectively. Ignoring the noise term and taking the logarithm of Equation (15.3) yields

$$\ln|Y(f)| = \ln|X(f)| + \ln|H(f)| \tag{15.4}$$

From Equation (15.4), in the log-frequency domain the effect of channel distortion is the addition of a "tilt" term $\ln|H(f)|$ to the signal spectrum.

15.1.1 The Ideal Inverse Channel Filter

The ideal inverse-channel filter, or the ideal equalizer, recovers the original input from the channel output signal. In the frequency domain, the ideal inverse channel filter can be expressed as

$$H(f)H^{\text{inv}}(f) = 1 \tag{15.5}$$

In Equation (15.5) $H^{\text{inv}}(f)$ is used to denote the inverse channel filter. For the ideal equalizer we have $H^{\text{inv}}(f) = H^{-1}(f)$, or, expressed in the log-frequency domain $\ln H^{\text{inv}}(f) = -\ln H(f)$. The general form of Equation (15.5) is given by the z-transform relation

$$H(z)H^{\text{inv}}(z) = z^{-N} \tag{15.6}$$

for some value of the delay N that makes the channel inversion process causal. Taking the inverse Fourier transform of Equation (15.5), we have the following convolutional relation between the impulse responses of the channel $\{h_k\}$ and the ideal inverse channel response $\{h_k^{\text{inv}}\}$:

$$\sum_k h_k^{\text{inv}} h_{i-k} = \delta(i) \tag{15.7}$$

where $\delta(i)$ is the Kronecker delta function. Assuming the channel output is noise-free and the channel is invertible, the ideal inverse channel filter can be used to reproduce the channel input signal with zero error, as follows.

The inverse filter output $\hat{x}(m)$, with the distorted signal $y(m)$ as the input, is given as

$$\begin{aligned}\hat{x}(m) &= \sum_k h_k^{\text{inv}}\, y(m-k) \\ &= \sum_k h_k^{\text{inv}} \sum_j h_j\, x(m-k-j) \\ &= \sum_i x(m-i) \sum_k h_k^{inv}\, h_{i-k}\end{aligned} \tag{15.8}$$

The last line of Equation (15.8) is derived by a change of variables $i=k+j$ in the second line and rearrangement of the terms. For the ideal inverse channel filter, substitution of Equation (15.7) in Equation (15.8) yields

$$\hat{x}(m) = \sum_i \delta(i)\, x(m-i) = x(m) \tag{15.9}$$

which is the desired result. In practice, it is not advisable to implement $H^{\text{inv}}(f)$ simply as $H^{-1}(f)$ because, in general, a channel response may be non-invertible. Even for invertible channels, a straightforward implementation of the inverse channel filter $H^{-1}(f)$ can cause problems. For example, at frequencies where $H(f)$ is small, its inverse $H^{-1}(f)$ is large, and this can lead to noise amplification if the signal-to-noise ratio is low.

15.1.2 Equalization Error, Convolutional Noise

The equalization error signal, also called the convolutional noise, is defined as the difference between the channel equalizer output and the desired signal:

$$\begin{aligned}v(m) &= x(m) - \hat{x}(m) \\ &= x(m) - \sum_{k=0}^{P-1} \hat{h}_k^{\text{inv}}\, y(m-k)\end{aligned} \tag{15.10}$$

where $\hat{h}_k^{\text{inv}}$ is an estimate of the inverse channel filter. Assuming that there is an ideal equalizer h_k^{inv} that can recover the channel input signal $x(m)$ from the channel output $y(m)$, we have

$$x(m) = \sum_{k=0}^{P-1} h_k^{\text{inv}}\, y(m-k) \tag{15.11}$$

Substitution of Equation (15.11) in Equation (15.10) yields

$$\begin{aligned} v(m) &= \sum_{k=0}^{P-1} h_k^{\text{inv}}\, y(m-k) - \sum_{k=0}^{P-1} \hat{h}_k^{\text{inv}}\, y(m-k) \\ &= \sum_{k=0}^{P-1} \tilde{h}_k^{\text{inv}}\, y(m-k) \end{aligned} \tag{15.12}$$

where $\tilde{h}_k^{\text{inv}} = h_k^{\text{inv}} - \hat{h}_k^{\text{inv}}$. The equalization error signal $v(m)$ may be viewed as the output of an error filter $\tilde{h}_k^{\text{inv}}$ in response to the input $y(m-k)$, hence the name "convolutional noise" for $v(m)$. When the equalization process is proceeding well, such that $\hat{x}(m)$ is a good estimate of the channel input $x(m)$, then the convolutional noise is relatively small and decorrelated and can be modelled as a zero mean Gaussian random process.

15.1.3 Blind Equalization

The equalization problem is relatively simple when the channel response is known and invertible, and when the channel output is not noisy. However, in most practical cases, the channel response is unknown, time-varying, non-linear, and may also be non-invertible. Furthermore, the channel output is often observed in additive noise.

Digital communication systems provide equalizer-training periods, during which a *training* pseudo-noise (PN) sequence, also available at the receiver, is transmitted. A synchronised version of the PN sequence is generated at the receiver, where the channel input and output signals are used for the identification of the channel equalizer as illustrated in Figure 15.2(a). The obvious drawback of using training periods for channel equalization is that power, time and bandwidth are consumed for the equalization process.

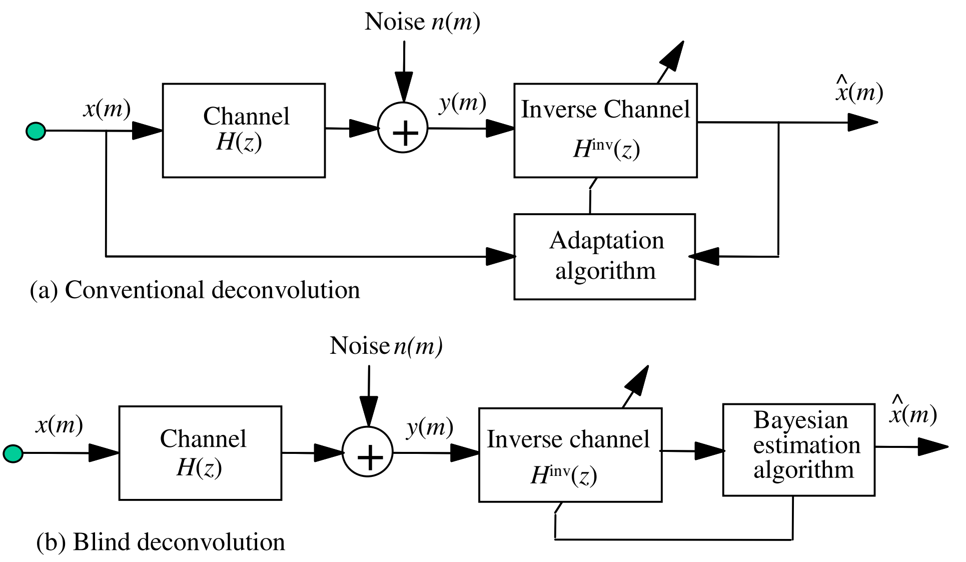

Figure 15.2 A comparative illustration of (a) a conventional equalizer with access to channel input and output, and (b) a blind equalizer.

It is preferable to have a "blind" equalization scheme that can operate without access to the channel input, as illustrated in Figure 15.2(b). Furthermore, in some applications, such as the restoration of acoustic recordings, or blurred images, all that is available is the distorted signal and the only restoration method applicable is blind equalization.

Blind equalization is feasible only if some statistical knowledge of the channel input, and perhaps that of the channel, is available. Blind equalization involves two stages of channel identification, and deconvolution of the input signal and the channel response, as follows:

(a) *Channel identification.* The general form of a channel estimator can be expressed as

$$\hat{\boldsymbol{h}} = \psi(\boldsymbol{y}, \mathcal{M}_x, \mathcal{M}_h) \tag{15.13}$$

where ψ is the channel estimator, the vector $\hat{\boldsymbol{h}}$ is an estimate of the channel response, $\boldsymbol{y}$ is the channel output, and $\mathcal{M}_{\boldsymbol{x}}$ and $\mathcal{M}_{\boldsymbol{h}}$ are statistical models of the channel input and the channel response respectively.

Channel identification methods rely on utilisation of a knowledge of the following characteristics of the input signal and the channel:

(i)The distribution of the channel input signal: for example, in decision-directed channel equalization, described in Section 15.5, the knowledge that the input is a binary signal is used in a binary decision device to estimate the channel input and to "direct" the equalizer adaptation process.

(ii) the relative durations of the channel input and the channel impulse response: the duration of a channel impulse response is usually orders of magnitude smaller than that of the channel input. This observation is used in Section 15.3.1 to estimate a stationary channel from the long-time averages of the channel output.

(iii) The stationary, or time-varying characteristics of the input signal process and the channel: in Section 15.3.1, a method is described for the recovery of a non-stationary signal convolved with the impulse response of a stationary channel.

(b) *Channel equalization.* Assuming that the channel is invertible, the channel input signal $x(m)$ can be recovered using an inverse channel filter as

$$\hat{x}(m)=\sum_{k=0}^{P-1}\hat{h}_k^{\text{inv}}\,y(m-k) \tag{15.14}$$

In the frequency domain, Equation (15.14) becomes

$$\hat{X}(f)=\hat{H}^{\text{inv}}(f)Y(f) \tag{15.15}$$

In practice, perfect recovery of the channel input may not be possible, either because the channel is non-invertible or because the output is observed in noise. A channel is non-invertible if:

(i) The channel transfer function is maximum-phase: the transfer function of a maximum-phase channel has zeros outside the unit circle, and hence the inverse channel has unstable poles. Maximum-phase channels are considered in the following section.

(ii) The channel transfer function maps many inputs to the same output: in these situations, a stable closed-form equation for the inverse channel does not exist, and instead an iterative deconvolution method is used. Figure 15.3 illustrates the frequency response of a channel that has one invertible and two non-invertible regions. In the non-invertible regions, the signal frequencies are heavily attenuated and lost to channel noise. In the invertible region, the signal is distorted but recoverable. This example illustrates that the inverse filter must be implemented with care in order to avoid undesirable results such as noise amplification at frequencies with low SNR.

15.1.4 Minimum- and Maximum-Phase Channels

For stability, all the poles of the transfer function of a channel must lie inside the unit circle. If all the zeros of the transfer function are also inside the unit circle then the channel is said to be a minimum-phase channel. If some of the zeros are outside the unit circle then the channel is said to be a maximum-phase channel. The inverse of a minimum-phase channel has all its poles inside the unit circle, and is therefore stable. The inverse of a maximum-phase channel has some of its poles outside the unit circle; therefore it has an exponentially growing impulse response and is unstable. However, a stable approximation of the inverse of a maximum-phase

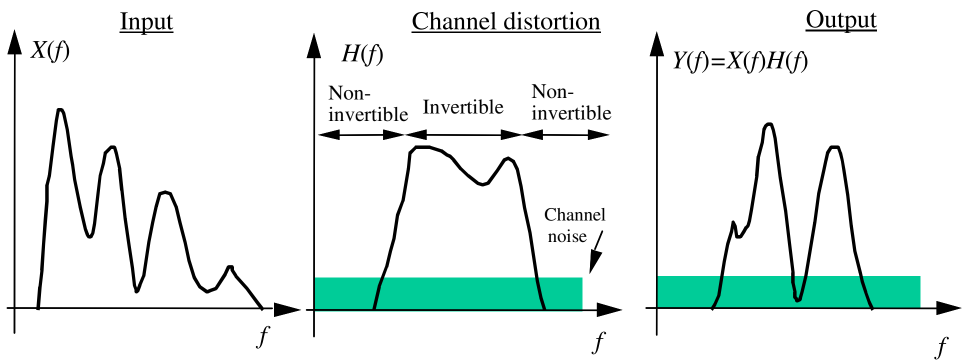

Figure 15.3 Illustration of the invertible and noninvertible regions of a channel.

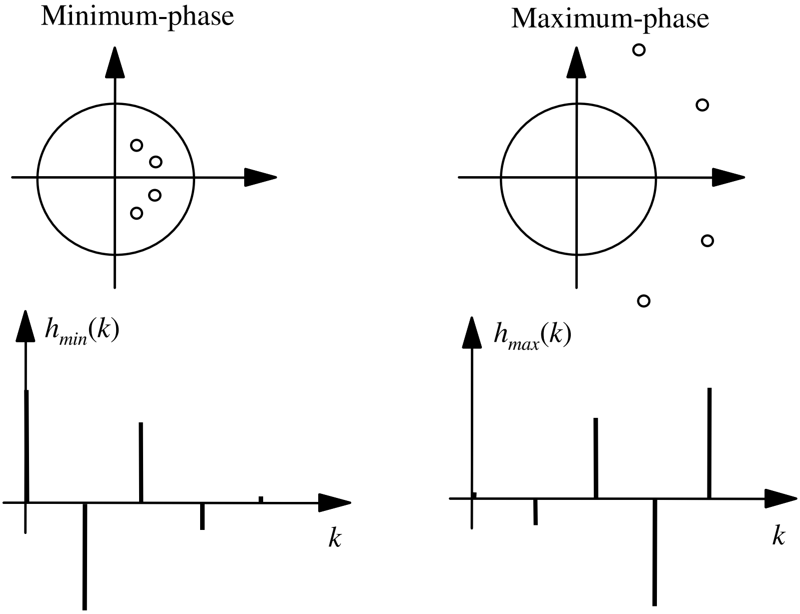

Figure 15.4 Illustration of the zero diagram and impulse response of fourth order maximum-phase and minimum-phase FIR filters.

channel may be obtained by truncating the impulse response of the inverse filter. Figure 15.3 illustrates examples of maximum-phase and minimum-phase fourth-order FIR filters.

When both the channel input and output signals are available, in the correct synchrony, it is possible to estimate the channel magnitude and phase response using the conventional least square error criterion. In blind deconvolution, there is no access to the exact instantaneous value or the timing of the channel input signal. The only information available is the channel output and some statistics of the channel input. The second order statistics of a signal (i.e. the correlation or the power spectrum) do not include the phase information; hence it is not possible to estimate the channel phase from the second-order statistics. Furthermore, the channel phase cannot be recovered if the input signal is Gaussian, because a Gaussian process of known mean is entirely specified by the autocovariance matrix, and autocovariance matrices do not include any phase information. For estimation of the phase of a channel, we can either use a non-linear estimate of the desired signal to direct the adaptation of a channel equalizer as in Section 15.5, or we can use the higher-order statistics as in Section 15.6.

15.1.5 Wiener Equalizer

In this section, we consider the least squared error Wiener equalization. Note that, in its conventional form, Wiener equalization is not a form of blind equalization, because the implementation of a Wiener equalizer requires the cross-correlation of the channel input and output signals, which are not available in a blind equalization application. The Wiener filter estimate of the channel input signal is given by

$$\hat{x}(m) = \sum_{k=0}^{P-1} \hat{h}_k^{\text{inv}} \, y(m-k) \tag{15.16}$$

where $\hat{h}_k^{\text{inv}}$ is an FIR Wiener filter estimate of the inverse channel impulse response. The equalization error signal $v(m)$ is defined as

$$v(m) = x(m) - \sum_{k=0}^{P-1} \hat{h}_k^{\text{inv}} \, y(m-k) \tag{15.17}$$

The Wiener equalizer with input $y(m)$ and desired output $x(m)$ is obtained from Equation (6.10) in Chapter 6 as

$$\hat{\boldsymbol{h}}^{\text{inv}} = \boldsymbol{R}_{yy}^{-1} \boldsymbol{r}_{xy} \tag{15.18}$$

where $\boldsymbol{R}_{yy}$ is the $P \times P$ autocorrelation matrix of the channel output, and $\boldsymbol{r}_{xy}$ is the P-dimensional cross-correlation vector of the channel input and output signals. A more expressive form of Equation (15.18) can be obtained by writing the noisy channel output signal in vector equation form as

$$\boldsymbol{y} = \boldsymbol{H}\boldsymbol{x} + \boldsymbol{n} \tag{4.19}$$

where $\boldsymbol{y}$ is an N-sample channel output vector, $\boldsymbol{x}$ is an $N+P$-sample channel input vector including the P initial samples, $\boldsymbol{H}$ is an $N\times(N+P)$ channel distortion matrix whose elements are composed of the coefficients of the channel filter, and $\boldsymbol{n}$ is a noise vector. The autocorrelation matrix of the channel output can be obtained from Equation (15.19) as

$$\boldsymbol{R}_{yy} = \mathcal{E}[\boldsymbol{y}\boldsymbol{y}^{\text{T}}] = \boldsymbol{H}\boldsymbol{R}_{xx}\boldsymbol{H}^{\text{T}} + \boldsymbol{R}_{nn} \tag{15.20}$$

where $\mathcal{E}[\cdot]$ is the expectation operator. The cross-correlation vector $\boldsymbol{r}_{xy}$ of the channel input and output signals becomes

$$\boldsymbol{r}_{xy} = \mathcal{E}[x\boldsymbol{y}] = \boldsymbol{H}\boldsymbol{r}_{xx} \tag{15.21}$$

Substitution of Equation (15.20) and (15.21) in (15.18) yields the Wiener equalizer as

$$\hat{\boldsymbol{h}}^{\text{inv}} = \left(\boldsymbol{H}\boldsymbol{R}_{\boldsymbol{xx}}\boldsymbol{H}^{\text{T}} + \boldsymbol{R}_{\boldsymbol{nn}}\right)^{-1}\boldsymbol{H}\boldsymbol{r}_{\boldsymbol{xx}} \tag{15.22}$$

The derivation of the Wiener equalizer in the frequency domain is as follows. The Fourier transform of the equalizer output is given by

$$\hat{X}(f) = \hat{H}^{\text{inv}}(f)Y(f) \tag{15.23}$$

where *Y(f)*, the channel output and $\hat{H}^{\text{inv}}(f)$ is the frequency response of the Wiener equalizer. The error signal *V(f)* is defined as

$$\begin{aligned} V(f) &= X(f) - \hat{X}(f) \\ &= X(f) - \hat{H}^{\text{inv}}(f)Y(f) \end{aligned} \tag{15.24}$$

As in Section 6.5 minimisation of the expectation of the squared magnitude of *V(f)* results in the frequency Wiener equalizer given by

$$\begin{aligned} \hat{H}^{\text{inv}}(f) &= \frac{P_{XY}(f)}{P_{YY}(f)} \\ &= \frac{P_{XX}(f)H^{*}(f)}{P_{XX}(f)\left|H(f)\right|^{2} + P_{NN}(f)} \end{aligned} \tag{15.25}$$

where $P_{XX}(f)$ is the channel input power spectrum, $P_{NN}(f)$ is the noise power spectrum, $P_{XY}(f)$ is the cross-power spectrum of the channel input and output signals, and *H(f)* is the frequency response of the channel. Note that in the absence of noise, $P_{NN}(f)=0$ and the Wiener inverse filter becomes $H^{\text{inv}}(f) = H^{-1}(f)$.

15.2 Blind Equalization Using Channel Input Power Spectrum

One of the early papers on blind deconvolution was by Stockham et al. (1975) on dereverberation of old acoustic recordings. Acoustic recorders, as illustrated in Figure 15.5, had a bandwidth of about 200 Hz to 4 kHz. However, the limited bandwidth, or even the additive noise or scratch noise pulses, are not considered as the major causes of distortions of acoustic recordings. The main distortion on acoustic recordings is due to reverberations of the recording horn instrument. An acoustic recording can be modelled as the convolution of the input audio signal $x(m)$ and the impulse response of a linear filter model of the recording instrument $\{h_k\}$, as in Equation (15.2), reproduced here for convenience

$$y(m) = \sum_{k=0}^{P-1} h_k x(m-k) + n(m) \tag{15.26}$$

or in the frequency domain as

$$Y(f) = X(f)H(f) + N(f) \tag{15.27}$$

where $H(f)$ is the frequency response of a linear time-invariant model of the acoustic recording instrument, and $N(f)$ is an additive noise. Multiplying

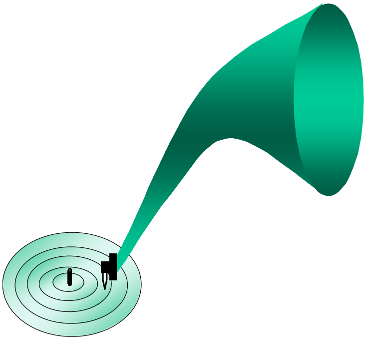

Figure 15.5 Illustration of the early acoustic recording process on a wax disc. Acoustic recordings were made by focusing the sound energy, through a horn via a sound box, diaphragm and stylus mechanism, onto a wax disc. The sound was distorted by reverberations of the horn.

both sides of Equation (15.27) with their complex conjugates, and taking the expectation, we obtain

$$\mathcal{E}[Y(f)Y^*(f)] = \mathcal{E}\left[\big(X(f)H(f)+N(f)\big)\big(X(f)H(f)+N(f)\big)^*\right] \quad (15.28)$$

Assuming the signal $X(f)$ and the noise $N(f)$ are uncorrelated Equation (15.28) becomes

$$P_{YY}(f) = P_{XX}(f)\left|H(f)\right|^2 + P_{NN}(f) \quad (15.29)$$

where $P_{YY}(f)$, $P_{XX}(f)$ and $P_{NN}(f)$ are the power spectra of the distorted signal, the original signal and the noise respectively. From Equation (15.29) an estimate of the spectrum of the channel response can be obtained as

$$\left|H(f)\right|^2 = \frac{P_{YY}(f) - P_{NN}(f)}{P_{XX}(f)} \quad (15.30)$$

In practice, Equation (15.30) is implemented using time-averaged estimates of the of the power spectra.

15.2.1 Homomorphic Equalization

In homomorphic equalization, the convolutional distortion is transformed, first into a multiplicative distortion through a Fourier transform of the distorted signal, and then into an additive distortion by taking the logarithm of the spectrum of the distorted signal. A further inverse Fourier transform operation converts the log-frequency variables into cepstral variables as illustrated in Figure 15.6. Through homomorphic transformation convolution becomes addition, and equalization becomes subtraction.

Ignoring the additive noise term and transforming both sides of Equation (15.27) into log-spectral variables yields

$$\ln Y(f) = \ln X(f) + \ln H(f) \quad (15.31)$$

Note that in the log-frequency domain, the effect of channel distortion is the addition of a tilt to the spectrum of the channel input. Taking the expectation of Equation (15.31) yields

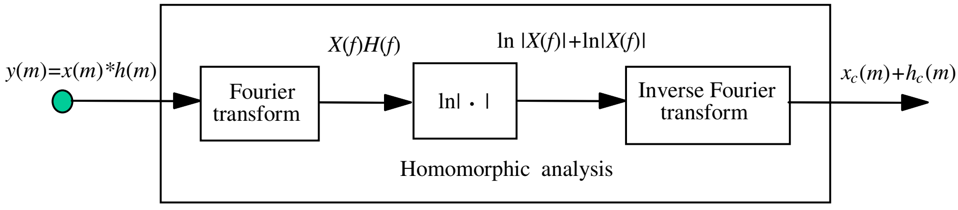

Figure 15.6 Illustration of homomorphic analysis in deconvolution.

$$\mathcal{E}[\ln Y(f)] = \mathcal{E}[\ln X(f)] + \ln H(f) \tag{15.32}$$

In Equation (15.32), it is assumed that the channel is time-invariant; hence $\mathcal{E}[\ln H(f)] = \ln H(f)$. Using the relation $\ln z = \ln|z| + j\angle z$, the term $\mathcal{E}[\ln X(f)]$ can be expressed as

$$\mathcal{E}[\ln X(f)] = \mathcal{E}[\ln|X(f)|] + j\mathcal{E}[\angle X(f)] \tag{15.33}$$

The first term on the right-hand side of Equation (15.33), $\mathcal{E}[\ln|X(f)|]$, is non-zero, and represents the frequency distribution of the signal power in decibels, whereas the second term $\mathcal{E}[\angle X(f)]$ is the expectation of the phase, and can be assumed to be zero. From Equation (15.32), the log-frequency spectrum of the channel can be estimated as

$$\ln H(f) = \mathcal{E}[\ln Y(f)] - \mathcal{E}[\ln X(f)] \tag{15.34}$$

In practice, when only a single record of a signal is available, the signal is divided into a number of segments, and the average signal spectrum is obtained over time across the segments. Assuming that the length of each segment is long compared with the duration of the channel impulse response, we can write an approximate convolutional relation for the i^{th} signal segment as

$$y_i(m) \approx x_i(m) * h_i(m) \tag{15.35}$$

The segments are windowed, using a Hamming or a Hanning window, to reduce the spectral leakage due to end effects at the edges of the segment. Taking the complex logarithm of the Fourier transform of Equation (15.35) yields

$$\ln Y_i(f) = \ln X_i(f) + \ln H_i(f) \tag{15.36}$$

Taking the time averages over N segments of the distorted signal record yields

$$\frac{1}{N}\sum_{i=0}^{N-1}\ln Y_i(f)=\frac{1}{N}\sum_{i=0}^{N-1}\ln X_i(f)+\frac{1}{N}\sum_{i=0}^{N-1}\ln H_i(f) \tag{15.37}$$

Estimation of the channel response from Equation (15.37) requires the average log spectrum of the undistorted signal $X(f)$. In Stockham's method for restoration of acoustic records, the expectation of the signal spectrum is obtained from a modern recording of the same musical material as that of the acoustic recording. From Equation (15.37), the estimate of the logarithm of the channel is given by

$$\ln \hat{H}(f)=\frac{1}{N}\sum_{i=0}^{N-1}\ln Y_i(f)-\frac{1}{N}\sum_{i=0}^{N-1}\ln X_i^{\mathcal{M}}(f) \tag{15.38}$$

where $X^{\mathcal{M}}(f)$ is the spectrum of a modern recording. The equalizer can then be defined as

$$\ln H^{\text{inv}}(f)=\begin{cases}-\ln \hat{H}(f), & 200\text{ Hz}\le f\le 4000\text{ Hz}\\ -40\text{ dB}, & \text{otherwise}\end{cases} \tag{15.39}$$

In Equation (15.39), the inverse acoustic channel is implemented in the range between 200 and 4000 Hz, where the channel is assumed to be invertible. Outside this range, the signal is dominated by noise, and the inverse filter is designed to attenuate the noisy signal.

15.2.2 Homomorphic Equalization Using a Bank of High-Pass Filters

In the log-frequency domain, channel distortion may be eliminated using a bank of high-pass filters. Consider a time sequence of log-spectra of the output of a channel described as

$$\ln Y_t(f)=\ln X_t(f)+\ln H_t(f) \tag{15.40}$$

where $Y_t(f)$ and $X_t(f)$ are the channel input and output derived from a Fourier transform of the t^{th} signal segment. From Equation (15.40), the effect of a time-invariant channel is to add a constant term ln $H(f)$ to each frequency component of the channel input $X_t(f)$, and the overall result is a time-invariant tilt of the log-frequency spectrum of the original signal. This observation suggests the use of a bank of narrowband high-pass notch filters for the removal of the additive distortion term ln$H(f)$. A simple first-order recursive digital filter with its notch at zero frequency is given by

$$\ln \hat{X}_t(f) = \alpha \ln \hat{X}_{t-1}(f) + \ln Y_t(f) - \ln Y_{t-1}(f) \tag{15.41}$$

where the parameter α controls the bandwidth of the notch at zero frequency. Note that the filter bank also removes any dc component of the signal ln $X(f)$; for some applications, such as speech recognition, this is acceptable.

15.3 Equalization Based on Linear Prediction Models

Linear prediction models, described in Chapter 8, are routinely used in applications such as seismic signal analysis and speech processing, for the modelling and identification of a minimum-phase channel. Linear prediction theory is based on two basic assumptions: that the channel is minimum-phase and that the channel input is a random signal. Standard linear prediction analysis can be viewed as a blind deconvolution method, because both the channel response and the channel input are unknown, and the only information is the channel output and the assumption that the channel input is random and hence has a flat power spectrum. In this section, we consider blind deconvolution using linear predictive models for the channel and its input. The channel input signal is modelled as

$$X(z) = E(z)A(z) \tag{15.42}$$

where $X(z)$ is the z-transform of the channel input signal, $A(z)$ is the z-transfer function of a linear predictive model of the channel input and $E(z)$ is the z-transform of a random excitation signal. Similarly, the channel output can be modelled by a linear predictive model $H(z)$ with input $X(z)$ and output $Y(z)$ as

$$Y(z) = X(z)H(z) \tag{15.43}$$

Figure 15.7 illustrates a cascade linear prediction model for a channel input process $X(z)$ and a channel response $H(z)$. The channel output can be expressed as

$$\begin{aligned} Y(z) &= E(z)A(z)H(z) \\ &= E(z)D(z) \end{aligned} \tag{15.44}$$

where

$$D(z) = A(z)H(z) \tag{15.45}$$

The z-transfer function of the linear prediction models of the channel input signal and the channel can be expanded as

$$A(z) = \frac{G_1}{1-\sum_{k=1}^{P} a_k z^{-k}} = \frac{G_1}{\prod_{k=1}^{P}(1-\alpha_k z^{-1})} \tag{15.46}$$

$$H(z) = \frac{G_2}{1-\sum_{k=1}^{Q} b_k z^{-k}} = \frac{G_2}{\prod_{k=1}^{Q}(1-\beta_k z^{-1})} \tag{15.47}$$

where $\{a_k, \alpha_k\}$ and $\{b_k, \beta_k\}$ are the coefficients and the poles of the linear prediction models for the channel input signal and the channel respectively. Substitution of Equations (15.46) and (15.47) in Equation (15.45) yields the combined input-channel model as

$$D(z) = \frac{G}{1-\sum_{k=1}^{P+Q} \mathrm{d}_k z^{-k}} = \frac{G}{\prod_{k=1}^{P+Q}(1-\gamma_k z^{-1})} \tag{15.48}$$

The total number of poles of the combined model for the input signal and the channel is the sum of the poles of the input signal model and the channel model.

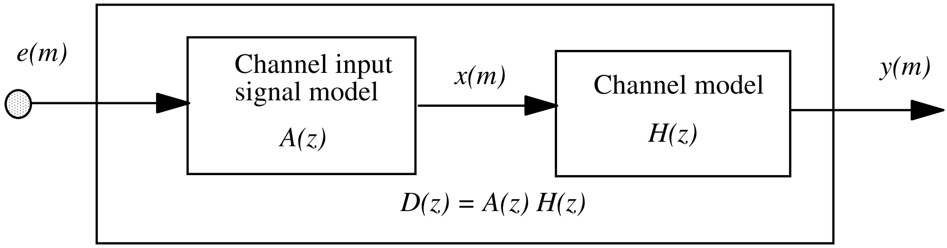

Figure 15.7 A distorted signal modelled as cascade of a signal model and a channel model.

15.3.1 Blind Equalization Through Model Factorisation

A model-based approach to blind equalization is to factorise the channel output model $D(z)=A(z)H(z)$ into a channel input signal model $A(z)$ and a channel model $H(z)$. If the channel input model $A(z)$ and the channel model $H(z)$ are non-factorable then the only factors of $D(z)$ are $A(z)$ and $H(z)$. However, z-transfer functions are factorable into the roots, the so-called poles and zeros, of the models. One approach to model-based deconvolution is to factorize the model for the convolved signal into its poles and zeros, and classify the poles and zeros as either belonging to the signal or belonging to the channel.

Spencer and Rayner (1990) developed a method for blind deconvolution through factorization of linear prediction models, based on the assumption that the channel is stationary with time-invariant poles whereas the input signal is non-stationary with time-varying poles. As an application, they considered the restoration of old acoustic recordings where a time-varying audio signal is distorted by the time-invariant frequency response of the recording equipment. For a simple example, consider the case when the signal and the channel are each modelled by a second-order linear predictive model. Let the time-varying second-order linear predictive model for the channel input signal $x(m)$ be

$$x(m)=a_1(m)x(m-1)+a_2(m)x(m-2)+G_1(m)e(m) \tag{15.49}$$

where $a_1(m)$ and $a_2(m)$ are the time-varying coefficients of the linear predictor model, $G_1(m)$ is the input gain factor and $e(m)$ is a zero-mean, unit variance, random signal. Now let $\alpha_1(m)$ and $\alpha_2(m)$ denote the time-varying poles of the predictor model of Equation (15.49); these poles are the roots of the polynomial

$$1-a_1(m)z^{-1}-a_2(m)z^{-2}=\left[1-z^{-1}\alpha_1(m)\right]\left[1-z^{-1}\alpha_2(m)\right]=0 \qquad (15.50)$$

Similarly, assume that the channel can be modelled by a second-order stationary linear predictive model as

$$y(m)=h_1 y(m-1)+h_2 y(m-2)+G_2 x(m) \qquad (15.51)$$

where h_1 and h_2 are the time-invariant predictor coefficients and G_2 is the channel gain. Let β_1 and β_2 denote the poles of the channel model; these are the roots of the polynomial

$$1-h_1 z^{-1}-h_2 z^{-2}=(1-z^{-1}\beta_1)(1-z^{-1}\beta_2)=0 \qquad (15.52)$$

The combined cascade of the two second-order models of Equations (15.49) and (15.51) can be written as a fourth-order linear predictive model with input $e(m)$ and output $y(m)$:

$$y(m)=d_1(m)y(m-1)+d_2(m)y(m-2)+d_3(m)y(m-3)+d_4(m)y(m-4)+Ge(m) \qquad (15.53)$$

where the combined gain $G=G_1G_2$. The poles of the fourth order predictor model of Equation (15.53) are the roots of the following polynomial:

$$\begin{aligned}1-d_1(m)z^{-1}-d_2(m)z^{-2}-d_3(m)z^{-3}-d_4(m)z^{-4} &= \\ =\left[1-z^{-1}\alpha_1(m)\right]\left[1-z^{-1}\alpha_2(m)\right]\left[1-z^{-1}\beta_1\right]\left[1-z^{-1}\beta_2\right] &= 0\end{aligned} \qquad (15.54)$$

In Equation (15.54) the poles of the fourth order predictor are $\alpha_1(m)$, $\alpha_2(m)$, β_1 and β_2. The above argument on factorisation of the poles of time-varying and stationary models can be generalised to a signal model of order P and a channel model of order Q.

In Spencer and Rayner, the separation of the stationary poles of the channel from the time-varying poles of the channel input is achieved through a clustering process. The signal record is divided into N segments and each segment is modelled by an all-pole model of order $P+Q$ where P and Q are the assumed model orders for the channel input and the channel respectively. In all, there are $N(P+Q)$ values which are clustered to form $P+Q$ clusters. Even if both the signal and the channel were stationary, the poles extracted from different segments would have variations due to the random character of the signals from which the poles are extracted. Assuming that the variances of the estimates of the stationary poles are small compared with the variations of the time-varying poles, it is expected that, for each stationary pole of the channel, the N values extracted from N segments will form an N-point cluster of a relatively small variance. These clusters can be identified and the centre of each cluster taken as a pole of the channel model This method assumes that the poles of the time-varying signal are well separated in space from the poles of the time-invariant signal.

15.4 Bayesian Blind Deconvolution and Equalization

The Bayesian inference method, described in Chapter 4, provides a general framework for inclusion of statistical models of the channel input and the channel response. In this section we consider the Bayesian equalization method, and study the case where the channel input is modelled by a set of hidden Markov models. The Bayesian risk for a channel estimate $\hat{\boldsymbol{h}}$ is defined as

$$
\begin{aligned}
\mathcal{R}(\hat{\boldsymbol{h}} \mid \boldsymbol{y}) &= \int_{\boldsymbol{H}} \int_{\boldsymbol{X}} C(\hat{\boldsymbol{h}}, \boldsymbol{h}) f_{\boldsymbol{X},\boldsymbol{H}|\boldsymbol{Y}}(\boldsymbol{x}, \boldsymbol{h} | \boldsymbol{y}) \, d\boldsymbol{x} \, d\boldsymbol{h} \\
&= \frac{1}{f_{\boldsymbol{Y}}(\boldsymbol{y})} \int_{\boldsymbol{H}} C(\hat{\boldsymbol{h}}, \boldsymbol{h}) f_{\boldsymbol{Y}|\boldsymbol{H}}(\boldsymbol{y} | \boldsymbol{h}) f_{\boldsymbol{H}}(\boldsymbol{h}) \, d\boldsymbol{h}
\end{aligned} \tag{15.55}
$$

where $C(\hat{\boldsymbol{h}}, \boldsymbol{h})$ is the cost of estimating the channel $\boldsymbol{h}$ as $\hat{\boldsymbol{h}}$, $f_{\boldsymbol{X},\boldsymbol{H}|\boldsymbol{Y}}(\boldsymbol{x}, \boldsymbol{h} | \boldsymbol{y})$ is the joint posterior density of the channel $\boldsymbol{h}$ and the channel input $\boldsymbol{x}$, $f_{\boldsymbol{Y}|\boldsymbol{H}}(\boldsymbol{y} | \boldsymbol{h})$ is the observation likelihood, and $f_{\boldsymbol{H}}(\boldsymbol{h})$ is the prior pdf of the channel. The Bayesian estimate is obtained by minimisation of the risk function $\mathcal{R}(\hat{\boldsymbol{h}} \mid \boldsymbol{y})$. There are a variety of Bayesian-type solutions depending on the choice of the cost function and the prior knowledge, as described in Chapter 4.

In this section, it is assumed that the convolutional channel distortion is transformed into an additive distortion through transformation of the channel output into log-spectral or cepstral variables. Ignoring the channel noise, the relation between the cepstra of the channel input and output signals is given by

$$\boldsymbol{y}(m)=\boldsymbol{x}(m)+\boldsymbol{h} \tag{15.56}$$

where the cepstral vectors $\boldsymbol{x}(m)$, $\boldsymbol{y}(m)$ and $\boldsymbol{h}$ are the channel input, the channel output and the channel respectively.

15.4.1 Conditional Mean Channel Estimation

A commonly used cost function in the Bayesian risk of Equation (15.55) is the mean square error $C(\boldsymbol{h}-\hat{\boldsymbol{h}})=|\boldsymbol{h}-\hat{\boldsymbol{h}}|^2$, which results in the conditional mean (CM) estimate defined as

$$\hat{\boldsymbol{h}}^{CM}=\int_{\boldsymbol{H}} \boldsymbol{h} f_{\boldsymbol{H}|\boldsymbol{Y}}(\boldsymbol{h}\,|\,\boldsymbol{y})\,d\boldsymbol{h} \tag{15.57}$$

The posterior density of the channel input signal may be conditioned on an estimate of the channel vector $\hat{\boldsymbol{h}}$ and expressed as $f_{\boldsymbol{X}|\boldsymbol{Y},\boldsymbol{H}}(\boldsymbol{x}\,|\,\boldsymbol{y},\hat{\boldsymbol{h}})$. The conditional mean of the channel input signal given the channel output $\boldsymbol{y}$ and an estimate of the channel $\hat{\boldsymbol{h}}$ is

$$\begin{aligned}\hat{\boldsymbol{x}}^{CM}&=\mathcal{E}[\boldsymbol{x}\,|\,\boldsymbol{y},\hat{\boldsymbol{h}}]\\&=\int_{\boldsymbol{X}} \boldsymbol{x} f_{\boldsymbol{X}|\boldsymbol{Y},\boldsymbol{H}}(\boldsymbol{x}\,|\,\boldsymbol{y},\hat{\boldsymbol{h}})\,d\boldsymbol{x}\end{aligned} \tag{15.58}$$

Equations (15.57) and (15.58) suggest a two-stage iterative method for channel estimation and the recovery of the channel input signal.

15.4.2 Maximum-Likelihood Channel Estimation

The ML channel estimate is equivalent to the case when the Bayes cost function and the channel prior are uniform. Assuming that the channel input signal has a Gaussian distribution with mean vector $\boldsymbol{\mu_x}$ and covariance matrix $\boldsymbol{\Sigma}_{xx}$, the likelihood of a sequence of N P-dimensional channel output vectors $\{\boldsymbol{y}(m)\}$ given a channel input vector $\boldsymbol{h}$ is

$$
\begin{aligned}
f_{Y|H}(\boldsymbol{y}(0),\dots,\boldsymbol{y}(N-1)|\boldsymbol{h}) &= \prod_{m=0}^{N-1} f_X(\boldsymbol{y}(m)-\boldsymbol{h}) \\
&= \prod_{m=0}^{N-1} \frac{1}{(2\pi)^{P/2}|\boldsymbol{\Sigma}_{xx}|^{1/2}} \exp\left\{[\boldsymbol{y}(m)-\boldsymbol{h}-\boldsymbol{\mu}_x]^{\mathrm{T}} \boldsymbol{\Sigma}_{xx}^{-1} [\boldsymbol{y}(m)-\boldsymbol{h}-\boldsymbol{\mu}_x]\right\}
\end{aligned}
\tag{15.59}
$$

To obtain the ML estimate of the channel $\boldsymbol{h}$, the derivative of the log likelihood function $\ln f_Y(\boldsymbol{y}|\boldsymbol{h})$ with respect to $\boldsymbol{h}$ is set to zero to yield

$$
\hat{\boldsymbol{h}}^{ML} = \frac{1}{N} \sum_{m=0}^{N-1} (\boldsymbol{y}(m) - \boldsymbol{\mu}_x)
\tag{15.60}
$$

15.4.3 Maximum A Posteriori Channel Estimation

The MAP estimate, like the ML estimate, is equivalent to a Bayesian estimator with a uniform cost function. However, the MAP estimate includes the prior pdf of the channel. The prior pdf can be used to confine the channel estimate within a desired subspace of the parameter space. Assuming that the channel input vectors are statistically independent, the posterior pdf of the channel given the observation sequence $\boldsymbol{Y}=\{\boldsymbol{y}(0), ..., \boldsymbol{y}(N-1)\}$ is

$$
\begin{aligned}
f_{H|Y}(\boldsymbol{h}|\boldsymbol{y}(0),\dots,\boldsymbol{y}(N-1)) &= \prod_{m=0}^{N-1} \frac{1}{f_Y(\boldsymbol{y}(m))} f_{Y|H}(\boldsymbol{y}(m)|\boldsymbol{h}) f_H(\boldsymbol{h}) \\
&= \prod_{m=0}^{N-1} \frac{1}{f_Y(\boldsymbol{y}(m))} f_X(\boldsymbol{y}(m)-\boldsymbol{h}) f_H(\boldsymbol{h})
\end{aligned}
\tag{15.61}
$$

Assuming that the channel input $\boldsymbol{x}(m)$ is Gaussian, $f_X(\boldsymbol{x}(m))=\mathcal{N}(\boldsymbol{x}, \boldsymbol{\mu}_x, \boldsymbol{\Sigma}_{xx})$, with mean vector $\boldsymbol{\mu}_x$ and covariance matrix $\boldsymbol{\Sigma}_{xx}$, and that the channel $\boldsymbol{h}$ is also Gaussian, $f_H(\boldsymbol{h})=\mathcal{N}(\boldsymbol{h}, \boldsymbol{\mu}_h, \boldsymbol{\Sigma}_{hh})$, with mean vector $\boldsymbol{\mu}_h$ and covariance matrix $\boldsymbol{\Sigma}_{hh}$, the logarithm of the posterior pdf is

$$\ln f_{\boldsymbol{H}|\boldsymbol{Y}}\left(\boldsymbol{h}|\boldsymbol{y}(0),\ldots,\boldsymbol{y}(N-1)\right) = -\sum_{m=0}^{N-1}\ln f\left(\boldsymbol{y}(m)\right) - NP\ln(2\pi) - \frac{1}{2}\ln\left(|\boldsymbol{\Sigma}_{\boldsymbol{xx}}||\boldsymbol{\Sigma}_{\boldsymbol{hh}}|\right)$$
$$-\sum_{m=0}^{N-1}\frac{1}{2}\left\{[\boldsymbol{y}(m)-\boldsymbol{h}-\boldsymbol{\mu}_{\boldsymbol{x}}]^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{xx}}^{-1}[\boldsymbol{y}(m)-\boldsymbol{h}-\boldsymbol{\mu}_{\boldsymbol{x}}] + (\boldsymbol{h}-\boldsymbol{\mu}_{\boldsymbol{h}})^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{hh}}^{-1}(\boldsymbol{h}-\boldsymbol{\mu}_{\boldsymbol{h}})\right\} \tag{15.62}$$

The MAP channel estimate, obtained by setting the derivative of the log posterior function $\ln f_{\boldsymbol{H}|\boldsymbol{Y}}(\boldsymbol{h}|\boldsymbol{y})$ to zero, is

$$\hat{\boldsymbol{h}}^{MAP} = (\boldsymbol{\Sigma}_{\boldsymbol{xx}} + \boldsymbol{\Sigma}_{\boldsymbol{hh}})^{-1}\boldsymbol{\Sigma}_{\boldsymbol{hh}}(\bar{\boldsymbol{y}} - \boldsymbol{\mu}_{\boldsymbol{x}}) + (\boldsymbol{\Sigma}_{\boldsymbol{xx}} + \boldsymbol{\Sigma}_{\boldsymbol{hh}})^{-1}\boldsymbol{\Sigma}_{\boldsymbol{xx}}\boldsymbol{\mu}_{\boldsymbol{h}} \tag{15.63}$$

where

$$\bar{\boldsymbol{y}} = \frac{1}{N}\sum_{m=0}^{N-1}\boldsymbol{y}(m) \tag{15.64}$$

is the time-averaged estimate of the mean of observation vector. Note that for a Gaussian process the MAP and conditional mean estimates are identical.

15.4.4 Channel Equalization Based on Hidden Markov Models

This section considers blind deconvolution in applications where the statistics of the channel input are modelled by a set of hidden Markov models. An application of this method, illustrated in Figure 15.8, is in recognition of speech distorted by a communication channel or a microphone. A hidden Markov model (HMM) is a finite-state Bayesian model, with a Markovian state prior and a Gaussian observation likelihood (see chapter 5). An N-state HMM can be used to model a non-stationary process, such as speech, as a chain of N stationary states connected by a set of Markovian state transitions. The likelihood of an HMM $\mathcal{M}_i$ and a sequence of N P-dimensional channel input vectors $\boldsymbol{X}$=[$\boldsymbol{x}$(0), ..., $\boldsymbol{x}$(N–1)] can be expressed in terms of the state transition and the observation pdfs of $\mathcal{M}_i$ as

$$f_{\boldsymbol{X}|\mathcal{M}}(\boldsymbol{X} \mid \mathcal{M}_i) = \sum_{\mathbf{s}} f_{\boldsymbol{X}|\mathcal{M},\boldsymbol{S}}(\boldsymbol{X} \mid \mathcal{M}_i, \mathbf{s}) P_{\boldsymbol{S}|\mathcal{M}}(\boldsymbol{s} \mid \mathcal{M}_i) \tag{15.65}$$

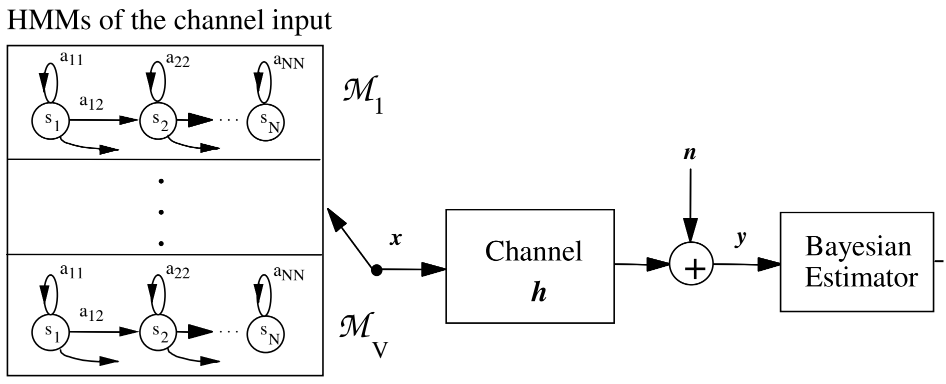

Figure 15.8 Illustration of a channel with the input modelled by a set of HMMs.

where $f_{X|\mathcal{M},S}(X \mid \mathcal{M}_i, \mathbf{s})$ is the likelihood that the sequence $\boldsymbol{X}=[\boldsymbol{x}(0), ..., \boldsymbol{x}(N-1)]$ was generated by the state sequence $\boldsymbol{s}=[s(0), ..., s(N-1)]$ of the model $\mathcal{M}_i$, and $P_{s|\mathcal{M}}(\boldsymbol{s} \mid \mathcal{M}_i)$ is the Markovian prior pmf of the state sequence $\boldsymbol{s}$. The Markovian prior entails that the probability of a transition to the state i at time m depends only on the state at time m–1 and is independent of the previous states. The transition probability of a Markov process is defined as

$$a_{ij} = P\big(s(m) = j \mid s(m-1) = i\big) \tag{15.66}$$

where a_{ij} is the probability of making a transition from state i to state j. The HMM state observation probability is often modelled by a multivariate Gaussian pdf as

$$f_{X|\mathcal{M},S}(\boldsymbol{x} \mid \mathcal{M}_i, s) = \frac{1}{(2\pi)^{P/2} \,|\boldsymbol{\Sigma}_{\boldsymbol{xx},s}|^{1/2}} \exp\left\{-\frac{1}{2}[\boldsymbol{x}-\boldsymbol{\mu}_{x,s}]^{\mathrm{T}} \boldsymbol{\Sigma}_{\boldsymbol{xx},s}^{-1} [\boldsymbol{x}-\boldsymbol{\mu}_{x,s}]\right\} \tag{15.67}$$

where $\boldsymbol{\mu}_{\boldsymbol{x},s}$ and $\boldsymbol{\Sigma}_{\boldsymbol{xx},s}$ are the mean vector and the covariance matrix of the Gaussian observation pdf of the HMM state s of the model $\mathcal{M}_i$.

The HMM-based channel equalization problem can be stated as follows: Given a sequence of N P-dimensional channel output vectors $\boldsymbol{Y}=[\boldsymbol{y}(0), ..., \boldsymbol{y}(N-1)]$, and the prior knowledge that the channel input sequence is drawn from a set of V HMMs $\mathcal{M}=\{\mathcal{M}_i \; i=1, ..., V\}$, estimate the channel response and the channel input.

The joint posterior pdf of an input word $\mathcal{M}_i$ and the channel vector $\boldsymbol{h}$ can be expressed as

$$f_{\mathcal{M},\boldsymbol{H}|\boldsymbol{Y}}(\mathcal{M}_i, \boldsymbol{h} \mid \boldsymbol{Y}) = P_{\mathcal{M}|\boldsymbol{H},\boldsymbol{Y}}(\mathcal{M}_i | \boldsymbol{h}, \boldsymbol{Y}) f_{\boldsymbol{H}|\boldsymbol{Y}}(\boldsymbol{h} \mid \boldsymbol{Y}) \tag{15.68}$$

Simultaneous joint estimation of the channel vector $\boldsymbol{h}$ and classification of the unknown input word $\mathcal{M}_i$ is a non-trivial exercise. The problem is usually approached iteratively by making an estimate of the channel response, and then using this estimate to obtain the channel input as follows. From Bayes' rule, the posterior pdf of the channel $\boldsymbol{h}$ conditioned on the assumption that the input model is $\mathcal{M}_i$ and given the observation sequence $\boldsymbol{Y}$ can be expressed as

$$f_{\boldsymbol{H}|\mathcal{M},\boldsymbol{Y}}(\boldsymbol{h} | \mathcal{M}_i, \boldsymbol{Y}) = \frac{1}{f_{\boldsymbol{Y}|\mathcal{M}}(\boldsymbol{Y} \mid \mathcal{M}_i)} f_{\boldsymbol{Y}|\mathcal{M},\boldsymbol{H}}(\boldsymbol{Y} | \mathcal{M}_i, \boldsymbol{h}) f_{\boldsymbol{H}|\mathcal{M}}(\boldsymbol{h} \mid \mathcal{M}_i) \tag{15.69}$$

The likelihood of the observation sequence, given the channel and the input word model, can be expressed as

$$f_{\boldsymbol{Y}|\mathcal{M},\boldsymbol{H}}(\boldsymbol{Y} \mid \mathcal{M}_i, \boldsymbol{h}) = f_{\boldsymbol{X}|\mathcal{M}}(\boldsymbol{Y} - \boldsymbol{h} \mid \mathcal{M}_i) \tag{15.70}$$

where it is assumed that the channel output is transformed into cepstral variables so that the channel distortion is additive. For a given input model $\mathcal{M}_i$, and state sequence $\boldsymbol{s}=[s(0), s(1), ..., s(N-1)]$, the pdf of a sequence of N independent observation vectors $\boldsymbol{Y}=[\boldsymbol{y}(0), \boldsymbol{y}(1), ..., \boldsymbol{y}(N-1)]$ is

$$\begin{aligned} f_{\boldsymbol{Y}|\boldsymbol{H},\mathrm{S},\mathcal{M}}(\boldsymbol{Y} \mid \boldsymbol{h}, \mathrm{s}, \mathcal{M}_i) &= \prod_{m=0}^{N-1} f_{\boldsymbol{X}|S,\mathrm{M}}(\boldsymbol{y}(m) - \boldsymbol{h} \mid s(m), \mathcal{M}_i) \\ &= \prod_{m=0}^{N-1} \frac{1}{(2\pi)^{P/2} \left| \boldsymbol{\Sigma}_{\boldsymbol{xx},s(m)} \right|^{1/2}} \exp\left\{ -\frac{1}{2} \left[\boldsymbol{y}(m) - \boldsymbol{h} - \boldsymbol{\mu}_{\boldsymbol{x},s(m)} \right]^{\mathrm{T}} \boldsymbol{\Sigma}_{\boldsymbol{xx},s(m)}^{-1} \left[\boldsymbol{y}(m) - \boldsymbol{h} - \boldsymbol{\mu}_{\boldsymbol{x},s(m)} \right] \right\} \end{aligned} \tag{15.71}$$

Taking the derivative of the log-likelihood of Equation (15.71) with respect to the channel vector $\boldsymbol{h}$ yields a maximum likelihood channel estimate as

$$\hat{\boldsymbol{h}}^{ML}(\boldsymbol{Y},\mathbf{s}) = \sum_{m=0}^{N-1} \left(\sum_{k=0}^{N-1} \boldsymbol{\Sigma}_{xx,s(k)}^{-1}\right)^{-1} \boldsymbol{\Sigma}_{xx,s(m)}^{-1} \left(\boldsymbol{y}(m) - \boldsymbol{\mu}_{x,s(m)}\right) \quad (15.72)$$

Note that when all the state observation covariance matrices are identical the channel estimate becomes

$$\hat{\boldsymbol{h}}^{ML}(\boldsymbol{Y},\mathbf{s}) = \frac{1}{N}\sum_{m=0}^{N-1}\left(\boldsymbol{y}(m) - \boldsymbol{\mu}_{\boldsymbol{x},s(m)}\right) \quad (15.73)$$

The ML estimate of Equation (15.73) is based on the ML state sequence $\boldsymbol{s}$ of $\mathcal{M}_i$. In the following section we consider the conditional mean estimate over all state sequences of a model.

15.4.5 MAP Channel Estimate Based on HMMs

The conditional pdf of a channel $\boldsymbol{h}$ averaged over all HMMs can be expressed as

$$f_{\boldsymbol{H}|\boldsymbol{Y}}(\boldsymbol{h}\,|\,\boldsymbol{Y}) = \sum_{i=1}^{V}\sum_{\boldsymbol{s}} f_{\boldsymbol{H}|\boldsymbol{Y},\boldsymbol{S},\mathcal{M}}(\boldsymbol{h}|\boldsymbol{Y},\boldsymbol{s},\mathcal{M}_i)P_{\boldsymbol{S}|\mathcal{M}}(\boldsymbol{s}|\mathcal{M}_i)P_{\mathcal{M}}(\mathcal{M}_i) \quad (15.74)$$

where $P_{\mathcal{M}}(\mathcal{M}_i)$ is the prior pmf of the input words. Given a sequence of N P-dimensional observation vectors $\boldsymbol{Y}=[\boldsymbol{y}(0), ..., \boldsymbol{y}(N–1)]$, the posterior pdf of the channel $\boldsymbol{h}$ along a state sequence $\boldsymbol{s}$ of an HMM $\mathcal{M}_i$ is defined as

$$f_{\boldsymbol{Y}|\boldsymbol{H},\mathrm{S},\mathrm{M}}(\boldsymbol{h}|\boldsymbol{Y},\mathbf{s},\mathcal{M}_i) = \frac{1}{f_{\boldsymbol{Y}}(\boldsymbol{Y})} f_{\boldsymbol{Y}|\boldsymbol{H},\mathrm{S},\mathcal{M}}(\boldsymbol{Y}|\boldsymbol{h},\mathbf{s},\mathcal{M}_i) f_{\boldsymbol{H}}(\boldsymbol{h})$$

$$= \frac{1}{f_{\boldsymbol{Y}}(\boldsymbol{Y})}\prod_{m=0}^{N-1}\frac{1}{(2\pi)^P\left|\boldsymbol{\Sigma}_{\boldsymbol{xx},s(m)}\right|^{1/2}\left|\boldsymbol{\Sigma}_{\boldsymbol{hh}}\right|^{1/2}}\exp\left\{-\frac{1}{2}\left[\boldsymbol{y}(m)-\boldsymbol{h}-\boldsymbol{\mu}_{\boldsymbol{x},s(m)}\right]^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{xx},s(m)}^{-1}\left[\boldsymbol{y}(m)-\boldsymbol{h}-\boldsymbol{\mu}_{\boldsymbol{x},s(m)}\right]\right\}$$

$$\times \exp\left[-\frac{1}{2}(\boldsymbol{h}-\boldsymbol{\mu}_{\boldsymbol{h}})^{\mathrm{T}}\boldsymbol{\Sigma}_{\boldsymbol{hh}}^{-1}(\boldsymbol{h}-\boldsymbol{\mu}_{\boldsymbol{h}})\right] \quad (15.75)$$

where it is assumed that each state of the HMM has a Gaussian distribution with mean vector $\boldsymbol{\mu}_{\boldsymbol{x},s(m)}$ and covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{xx},s(m)}$, and that the channel $\boldsymbol{h}$ is also Gaussian-distributed, with mean vector $\boldsymbol{\mu}_{\boldsymbol{h}}$ and covariance matrix

Σ_{hh}. The MAP estimate along state s, on the left-hand side of Equation (15.75), can be obtained as

$$\hat{h}^{MAP}(Y,\mathbf{s},\mathcal{M}_i)=\sum_{m=0}^{N-1}\left[\sum_{k=0}^{N-1}\left(\Sigma_{xx,s(k)}^{-1}+\Sigma_{hh}^{-1}\right)\right]^{-1}\Sigma_{xx,s(m)}^{-1}\left[y(m)-\mu_{x,s(m)}\right]$$
$$+\left[\sum_{k=0}^{N-1}\left(\Sigma_{xx,s(k)}^{-1}+\Sigma_{hh}^{-1}\right)\right]^{-1}\Sigma_{hh}^{-1}\mu_h \tag{15.76}$$

The MAP estimate of the channel over all state sequences of all HMMs can be obtained as

$$\hat{h}(Y)=\sum_{i=1}^{V}\sum_{\mathbf{S}}\hat{h}^{MAP}(Y,\mathbf{s},\mathcal{M}_i)P_{\mathbf{S}|\mathcal{M}}(\mathbf{s}\,|\,\mathcal{M}_i)P_{\mathcal{M}}(\mathcal{M}_i) \tag{15.77}$$

15.4.6 Implementations of HMM-Based Deconvolution

In this section, we consider three implementation methods for HMM-based channel equalization.

Method I: Use of the Statistical Averages Taken Over All HMMs

A simple approach to blind equalization, similar to that proposed by Stockham, is to use as the channel input statistics the average of the mean vectors and the covariance matrices, taken over all the states of all the HMMs as

$$\mu_x=\frac{1}{VN_s}\sum_{i=1}^{V}\sum_{j=1}^{N_s}\mu_{\mathcal{M}_i,j},\quad \Sigma_{xx}=\frac{1}{VN_s}\sum_{i=1}^{V}\sum_{j=1}^{N_s}\Sigma_{\mathcal{M}_i,j} \tag{15.78}$$

where $\mu_{\mathcal{M}_i,j}$ and $\Sigma_{\mathcal{M}_i,j}$ are the mean and the covariance of the j^{th} state of the i^{th} HMM, V and N_s denote the number of models and number of states per model respectively. The maximum likelihood estimate of the channel, $\hat{h}^{ML}$, is defined as

$$\hat{\boldsymbol{h}}^{ML} = (\bar{\boldsymbol{y}} - \boldsymbol{\mu}_{\mathbf{x}}) \tag{15.79}$$

where $\bar{\boldsymbol{y}}$ is the time-averaged channel output. The estimate of the channel input is

$$\hat{\boldsymbol{x}}(m) = \boldsymbol{y}(m) - \hat{\boldsymbol{h}}^{ML} \tag{15.80}$$

Using the averages over all states and models, the MAP channel estimate becomes

$$\hat{\boldsymbol{h}}^{MAP}(\boldsymbol{Y}) = \sum_{m=0}^{N-1} (\boldsymbol{\Sigma}_{\boldsymbol{xx}} + \boldsymbol{\Sigma}_{\boldsymbol{hh}})^{-1} \boldsymbol{\Sigma}_{\boldsymbol{hh}} \; (\boldsymbol{y}(m) - \boldsymbol{\mu}_{\boldsymbol{x}}) + (\boldsymbol{\Sigma}_{\boldsymbol{xx}} + \boldsymbol{\Sigma}_{\boldsymbol{hh}})^{-1} \boldsymbol{\Sigma}_{\boldsymbol{xx}} \, \boldsymbol{\mu}_{\boldsymbol{h}} \tag{15.81}$$

Method II: Hypothesised-Input HMM Equalization

In this method, for each candidate HMM in the input vocabulary, a channel estimate is obtained and then used to equalise the channel output, prior to the computation of a likelihood score for the HMM. Thus a channel estimate $\hat{\boldsymbol{h}}_w$ is based on the hypothesis that the input word is w. It is expected that a better channel estimate is obtained from the correctly hypothesised HMM, and a poorer estimate from an incorrectly hypothesised HMM. The hypothesised-input HMM algorithm is as follows (Figure 15.9):

For $i = 1$ to number of words V {

step 1 Using each HMM, $\mathcal{M}_i$, make an estimate of the channel, $\hat{\boldsymbol{h}}_i$,

step 2 Using the channel estimate, $\hat{\boldsymbol{h}}_i$, estimate the channel input $\hat{\boldsymbol{x}}(m) = \boldsymbol{y}(m) - \hat{\boldsymbol{h}}_i$

step 3 Compute a probability score for model $\mathcal{M}_i$, given the estimate $[\hat{\mathbf{x}}(m)]$. }

Select the channel estimate associated with the most probable word.

Figure 15.10 shows the ML channel estimates of two channels using unweighted average and hypothesised-input methods.

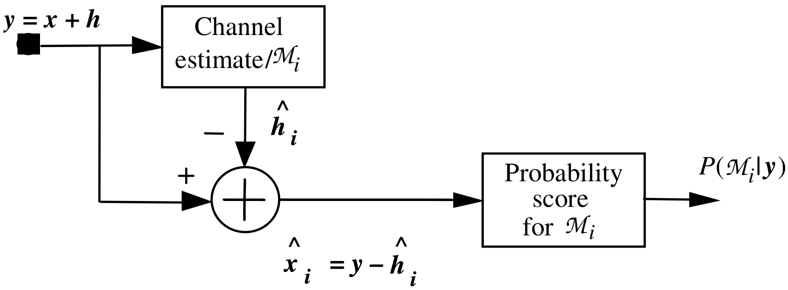

Figure 15.9 Hypothesised channel estimation procedure.

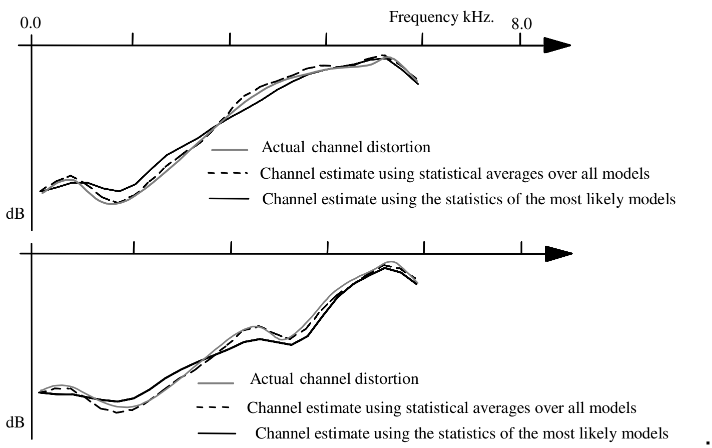

Figure 15.10 Illustration of actual and estimated channel response for two channels.

Method III: Decision-Directed Equalization

Blind adaptive equalizers are often composed of two distinct sections: an adaptive linear equalizer followed by a non-linear estimator to improve the equalizer output. The output of the non-linear estimator is the final estimate of the channel input, and is used as the desired signal *to direct* the equalizer adaptation. The use of the output of the non-linear estimator as the desired signal assumes that the linear equalization filter removes a large part of the channel distortion, thereby enabling the non-linear estimator to produce an accurate estimate of the channel input. A method of ensuring that the equalizer locks into, and cancels a large part of the channel distortion is to use a startup, equalizer training period during which a known signal is transmitted.

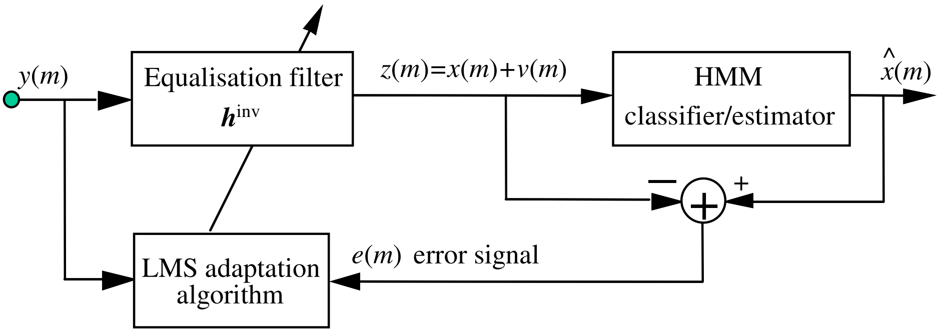

Figure 15.11 A decision-directed equalizer.

Figure 15.11 illustrates a blind equalizer incorporating an adaptive linear filter followed by a hidden Markov model classifier/estimator. The HMM classifies the output of the filter as one of a number of likely signals and provides an enhanced output, which is also used for adaptation of the linear filter. The output of the equalizer $z(m)$ is expressed as the sum of the input to the channel $x(m)$ and a so-called convolutional noise term $v(m)$ as

$$z(m) = x(m) + v(m) \tag{15.82}$$

The HMM may incorporate state-based Wiener filters for suppression of the convolutional noise $v(m)$ as described in Section 5.5. Assuming that the LMS adaptation method is employed, the adaptation of the equalizer coefficient vector is governed by the following recursive equation:

$$\hat{h}^{\text{inv}}(m) = \hat{h}^{\text{inv}}(m-1) + \mu e(m) y(m) \tag{15.83}$$

where $\hat{h}^{\text{inv}}(m)$ is an estimate of the optimal inverse channel filter, μ is an adaptation step size and the error signal $e(m)$ is defined as

$$e(m) = \hat{x}^{HMM}(m) - z(m) \tag{15.84}$$

where $\hat{x}^{HMM}(m)$ is the output of the HMM-based estimator and is used as the correct estimate of the desired signal to direct the adaptation process.

15.5 Blind Equalization for Digital Communication Channels

High speed transmission of digital data over analog channels, such as telephone lines or a radio channels, requires adaptive equalization to reduce decoding errors caused by channel distortions. In telephone lines, the channel distortions are due to the non-ideal magnitude response and the nonlinear phase response of the lines. In radio channel environments, the distortions are due to non-ideal channel response as well as the effects of multipath propagation of the radio waves via a multitude of different routes with different attenuations and delays. In general, the main types of distortions suffered by transmitted symbols are amplitude distortion, time dispersion and fading. Of these, time dispersion is perhaps the most important, and has received a great deal of attention. Time dispersion has the effect of smearing and elongating the duration of each symbol. In high speed communication systems, where the data symbols closely follow each other, time dispersion results in an overlap of successive symbols, an effect known as intersymbol interference (ISI), illustrated in Figure 15.12.

In a digital communication system, the transmitter modem takes N bits of binary data at a time, and encodes them into one of 2^N analog symbols for transmission, at the signalling rate, over an analog channel. At the receiver the analog signal is sampled and decoded into the required digital format. Most digital modems are based on multilevel phase-shift keying, or combined amplitude and phase shift keying schemes. In this section we consider multi-level pulse amplitude modulation (M-ary PAM) as a convenient scheme for the study of adaptive channel equalization.

Assume that at the transmitter modem, the k^{th} set of N binary digits is mapped into a pulse of duration T_s seconds and an amplitude $a(k)$. Thus the modulator output signal, which is the input to the communication channel, is given as

$$x(t)=\sum_{k} a(k)r(t-kT_s) \tag{15.85}$$

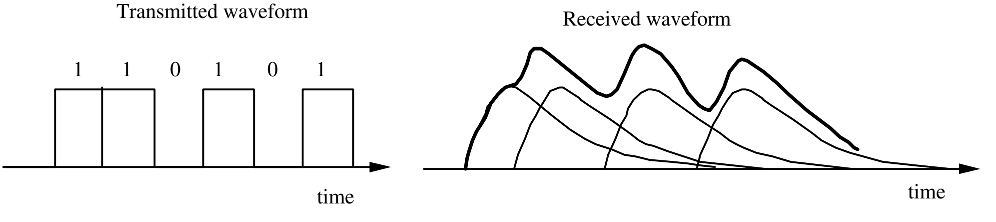

Figure 15.12 Illustration of intersymbol interference in a binary pulse amplitude modulation system.

where $r(t)$ is a pulse of duration T_s and with an amplitude $a(k)$ that can assume one of $M=2^N$ distinct levels. Assuming that the channel is linear, the channel output can be modelled as the convolution of the input signal and channel response:

$$y(t)=\int_{-\infty}^{\infty} h(\tau)x(t-\tau)d\tau \tag{15.86}$$

where $h(t)$ is the channel impulse response. The sampled version of the channel output is given by the following discrete-time equation:

$$y(m)=\sum_k h_k\, x(m-k) \tag{15.87}$$

To remove the channel distortion, the sampled channel output $y(m)$ is passed to an equalizer with an impulse response $\hat{h}_k^{\text{inv}}$. The equalizer output $z(m)$ is given as

$$\begin{aligned} z(m) &= \sum_k \hat{h}_k^{\text{inv}}\, y(m-k) \\ &= \sum_j x(m-j)\sum_k \hat{h}_k^{inv}\, h_{j-k} \end{aligned} \tag{15.88}$$

where Equation (15.87) is used to obtain the second line of Equation (15.88). The ideal equalizer output is $z(m)=x(m–D)=a(m–D)$ for some delay D that depends on the channel response and the length of the equalizer. From Equation (15.88), the channel distortion would be cancelled if

$$h_m^c = h_m * \hat{h}_m^{\text{inv}} = \delta(m-D) \tag{15.89}$$

where h_m^c is the combined impulse response of the cascade of the channel and the equalizer. A particular form of channel equalizer, for the elimination of ISI, is the Nyquist *zero-forcing* filter, where the impulse response of the combined channel and equalizer is defined as

$$h^c(kT_s + D) = \begin{cases} 1, & k = 0 \\ 0, & k \neq 0 \end{cases} \tag{15.90}$$

Note that in Equation (15.90), at the sampling instants the channel distortion is cancelled, and hence there is no ISI at the sampling instants. A function that satisfies Equation (15.90) is the sinc function $h^c(t)=\sin(\pi f_s t)/\pi f_s t$, where $f_s=1/T_s$. Zero-forcing methods are sensitive to deviations of $h^c(t)$ from the requirement of Equation (15.90), and also to jitters in the synchronisation and the sampling process.

15.5.1 LMS Blind Equalization

In this section, we consider the more general form of the LMS-based adaptive equalizer followed by a nonlinear estimator. In a conventional sample-adaptive filter, the filter coefficients are adjusted to minimise the mean squared distance between the filter output and the desired signal. In blind equalization, the desired signal (which is the channel input) is not available. The use of an adaptive filter for blind equalization, requires an internally generated desired signal as illustrated in Figure 15.13. Digital blind equalizers are composed of two distinct sections: an adaptive equalizer that removes a large part of the channel distortion, followed by a non-linear estimator for an improved estimate of the channel input. The output of the non-linear estimator is the final estimate of the channel input, and is used as the desired signal *to direct* the equalizer adaptation. A method of ensuring that the equalizer removes a large part of the channel distortion is to use a start-up, equalizer training, period during which a known signal is transmitted.

Assuming that the LMS adaptation method is employed, the adaptation of the equalizer coefficient vector is governed by the following recursive equation:

$$\hat{\boldsymbol{h}}^{\text{inv}}(m)=\hat{\boldsymbol{h}}^{\text{inv}}(m-1)+\mu e(m)\boldsymbol{y}(m) \tag{15.91}$$

where $\hat{\boldsymbol{h}}^{\text{inv}}(m)$ is an estimate of the optimal inverse channel filter $\boldsymbol{h}^{\text{inv}}$, the scalar μ is the adaptation step size, and the error signal $e(m)$ is defined as

$$\begin{aligned} e(m) &= \psi(z(m)) - z(m) \\ &= \hat{x}(m) - z(m) \end{aligned} \tag{15.92}$$

where $\hat{x}(m) = \psi(z(m))$ is a non-linear estimate of the channel input. For example, in a binary communication system with an input alphabet $\{\pm a\}$ we

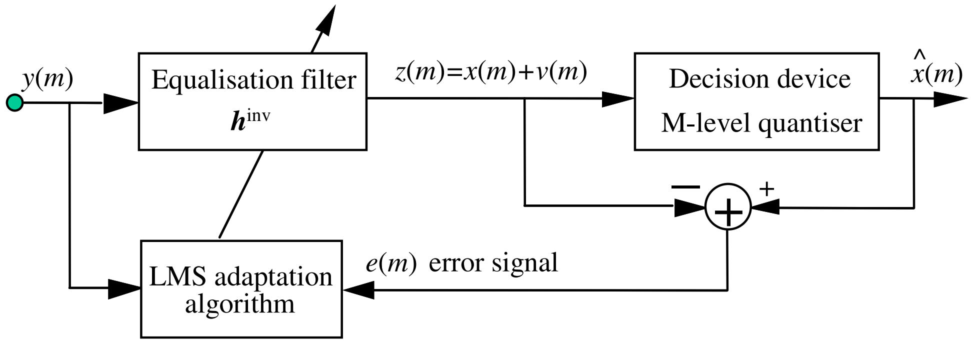

Figure 15.13 Configuration of an adaptive channel equalizer with an estimate of the channel input used as an "internally" generated desired signal

can use a signum non-linearity such that $\hat{x}(m) = a.sgn(z(m))$ where the function $sgn(\cdot)$ gives the sign of the argument. In the following, we use a Bayesian framework to formulate the nonlinear estimator $\psi()$.

Assuming that the channel input is an uncorrelated process and the equalizer removes a large part of the channel distortion, the equalizer output can be expressed as the sum of the desired signal (the channel input) plus an uncorrelated additive noise term:

$$z(m) = x(m) + v(m) \tag{15.93}$$

where $v(m)$ is the so-called convolutional noise defined as

$$\begin{aligned} v(m) &= x(m) - \sum_k \hat{h}_k^{\text{inv}} y(m-k) \\ &= \sum_k (h_k^{\text{inv}} - \hat{h}_k^{\text{inv}}) y(m-k) \end{aligned} \tag{15.94}$$

In the following, we assume that the non-linear estimates of the channel input are correct, and hence the error signals $e(m)$ and $v(m)$ are identical. Owing to the averaging effect of the channel and the equalizer, each sample of convolutional noise is affected by many samples of the input process. From the central limit theorem, the convolutional noise $e(m)$ can be modelled by a zero-mean Gaussian process as

$$f_E(e(m)) = \frac{1}{\sqrt{2\pi}\,\sigma_e} \exp\left(-\frac{e^2(m)}{2\sigma_e^2} \right) \tag{15.95}$$

where σ_e^2, the noise variance, can be estimated using the recursive time-update equation

$$\sigma_e^2(m) = \rho\sigma_e^2(m-1) + (1-\rho)e^2(m) \tag{15.96}$$

where $\rho < 1$ is the adaptation factor. The Bayesian estimate of the channel input given the equalizer output can be expressed in a general form as

$$\hat{x}(m) = \arg\min_{\hat{x}(m)} \int_X C\big(x(m), \hat{x}(m)\big) f_{X|Z}\big(x(m) \mid z(m)\big)\, dx(m) \tag{15.97}$$

where $C\big(x(m), \hat{x}(m)\big)$ is a cost function and $f_{X|Z}(x(m)|z(m))$ is the posterior pdf of the channel input signal. The choice of the cost function determines the type of the estimator as described in Chapter 4. Using a uniform cost function in Equation (15.97) yields the maximum a posteriori (MAP) estimate

$$\begin{aligned} \hat{x}^{MAP}(m) &= \arg\max_{x(m)} f_{X|Z}\big(x(m) \mid z(m)\big) \\ &= \arg\max_{x(m)} f_E\big(z(m) - x(m)\big) P_X\big(x(m)\big) \end{aligned} \tag{15.98}$$

Now, as an example consider an M-ary pulse amplitude modulation system, and let $\{a_i \quad i=1, ..., M\}$ denote the set of M pulse amplitudes with a probability mass function

$$P_X\big(x(m)\big) = \sum_{i=1}^{M} P_i \delta\big(x(m) - a_i\big) \tag{15.99}$$

The pdf of the equalizer output $z(m)$ can be expressed as the mixture pdf

$$f_Z\big(z(m)\big) = \sum_{i=1}^{M} P_i f_E\big(x(m) - a_i\big) \tag{15.100}$$

The posterior density of the channel input is

$$P_{X|Z}(x(m)=a_i|z(m))=\frac{1}{f_Z(z(m))}f_E(z(m)-a_i)P_X(x(m)=a_i) \qquad (15.101)$$

and the MAP estimate is obtained from

$$\hat{x}^{MAP}(m)=\arg\max_{a_i}(f_E(z(m)-a_i)P_X(x(m)=a_i)) \qquad (15.102)$$

Note that the classification of the continuous-valued equalizer output $z(m)$ into one of M discrete channel input symbols is basically a non-linear process. Substitution of the zero-mean Gaussian model for the convolutional noise $e(m)$ in Equation (102) yields

$$\hat{x}^{MAP}(m)=\arg\max_{a_i}\left[P_X(x(m)=a_i)\exp\left\{-\frac{[z(m)-a_i]^2}{2\sigma_e^2}\right\}\right] \qquad (15.103)$$

Note that when the symbols are equiprobable, the MAP estimate reduces to a simple threshold decision device. Figure 15.13 shows a channel equalizer followed by an M-level quantiser. In this system, the output of the equalizer filter is passed to an M-ary decision circuit. The decision device, which is essentially an M-level quantiser, classifies the channel output into one of M valid symbols. The output of the decision device is taken as an internally generated desired signal to direct the equalizer adaptation.

15.5.2 Equalization of a Binary Digital Channel

Consider a binary PAM communication system with an input symbol alphabet $\{a_0, a_1\}$ and symbol probabilities $P(a_0)=P_0$ and $P(a_1)=P_1=1-P_0$. The pmf of the amplitude of the channel input signal can be expressed as

$$P(x(m))=P_0\delta(x(m)-a_0)+P_1\delta(x(m)-a_1) \qquad (15.104)$$

Assume that at the output of the linear adaptive equalizer in Figure 15.13, the convolutional noise $v(m)$ is a zero-mean Gaussian process with variance

σ_v^2. Therefore the pdf of the equalizer output $z(m)=x(m)+v(m)$ is a mixture of two Gaussian pdfs and can be described as

$$f_Z(z(m)) = \frac{P_0}{\sqrt{2\pi}\,\sigma_v} \exp\left\{-\frac{[z(m)-a_0]^2}{2\sigma_v^2}\right\} + \frac{P_1}{\sqrt{2\pi}\,\sigma_v} \exp\left\{-\frac{[z(m)-a_1]^2}{2\sigma_v^2}\right\} \tag{15.105}$$

The MAP estimate of the channel input signal is

$$\hat{x}(m) = \begin{cases} a_0 & \text{if } \dfrac{P_0}{\sqrt{2\pi}\sigma_v} \exp\left\{-\dfrac{[z(m)-a_0]^2}{2\sigma_v^2}\right\} < \dfrac{P_1}{\sqrt{2\pi}\sigma_v} \exp\left\{-\dfrac{[z(m)-a_1]^2}{2\sigma_v^2}\right\} \\ a_1 & \text{otherwise} \end{cases} \tag{15.106}$$

For the case when the channel alphabet consists of $a_0=-a$, $a_1=a$ and $P_0=P_1$, the MAP estimator is identical to the signum function $sgn(x(m))$, and the error signal is given by

$$e(m) = z(m) - sgn(z(m))\,a \tag{15.107}$$

Figure 15.14 shows the error signal as a function of $z(m)$. An undesirable property of a hard non-linearity, such as the $sgn(\cdot)$ function, is that it produces a large error signal at those instances when $z(m)$ is around zero,

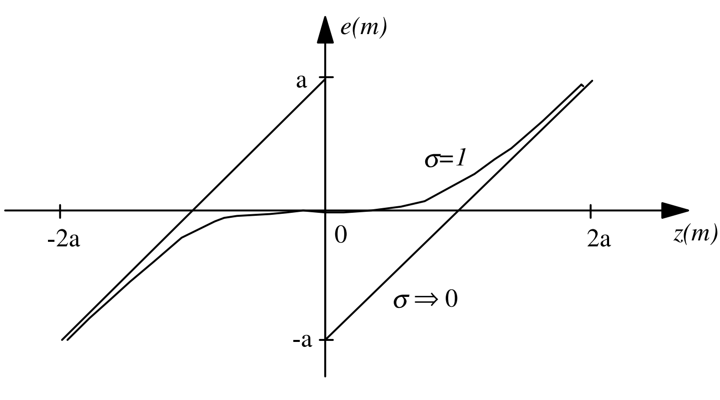

Figure 15.14 Comparison of the error functions produced by the hard non-linearity of a sign function Equation (15.107) and the soft non-linearity of Equation

and a decision based on the sign of $z(m)$ is most likely to be incorrect.

A large error signal based on an incorrect decision would have an unsettling effect on the convergence of the adaptive equalizer. It is desirable to have an error function that produces small error signals when $z(m)$ is around zero. Nowlan and Hinton proposed a soft non-linearity of the following form

$$e(m)=z(m)-\frac{e^{2az(m)/\sigma^2}-1}{e^{2az(m)/\sigma^2}+1}a \tag{15.108}$$

The error $e(m)$ is small when the magnitude of $z(m)$ is small and large when magnitude of $z(m)$ is large.

15.6 Equalization Based on Higher-Order Statistics

The second-order statistics of a random process, namely the autocorrelation or its Fourier transform the power spectrum, are central to the development the linear estimation theory, and form the basis of most statistical signal processing methods such as Wiener filters and linear predictive models. An attraction of the correlation function is that a Gaussian process, of a known mean vector, can be completely described in terms of the covariance matrix, and many random processes can be well characterised by Gaussian or mixture Gaussian models. A shortcoming of second-order statistics is that they do not include the phase characteristics of the process. Therefore, given the channel output, it is not possible to estimate the channel phase from the second-order statistics. Furthermore, as a Gaussian process of known mean depends entirely on the autocovariance function, it follows that blind deconvolution, based on a Gaussian model of the channel input, cannot estimate the channel phase.

Higher-order statistics, and the probability models based on them, can model both the magnitude and the phase characteristics of a random process. In this section, we consider blind deconvolution based on higher-order statistics and their Fourier transforms known as the higher-order spectra. The prime motivation in using the higher-order statistics is their ability to model the phase characteristics. Further motivations are the potential of the higher order statistics to model channel non-linearities, and to estimate a non-Gaussian signal in a high level of Gaussian noise.

15.6.1 Higher-Order Moments, Cumulants and Spectra

The k^{th} *order* moment of a random variable X is defined as

$$\begin{aligned} m_k &= \mathcal{E}[x^k] \\ &= (-j)^k \left. \frac{\partial^k \Phi_X(\omega)}{\partial \omega^k} \right|_{\omega=0} \end{aligned} \tag{15.109}$$

where $\Phi_X(\omega)$ is the *characteristic function* of the random variable X defined as

$$\Phi_X(\omega) = \mathcal{E}[\exp(j\omega x)] \tag{15.110}$$

From Equations (15.109) and (15.110), the first moment of X is $m_1 = \mathcal{E}[x]$, the second moment of X is $m_2 = \mathcal{E}[x^2]$, and so on. The joint k^{th} order moment $(k=k_1+k_2)$ of two random variables X_1 and X_2 is defined as

$$\mathcal{E}[x_1^{k_1} x_2^{k_2}] = (-j)^{k_1+k_2} \left. \frac{\partial^{k_1} \partial^{k_2} \Phi_{X_1 X_2}(\omega_1, \omega_2)}{\partial \omega_1^{k_1} \partial \omega_2^{k_2}} \right|_{\omega_1=\omega_2=0} \tag{15.111}$$

and in general the joint k^{th} order moment of N random variables is defined as

$$\begin{aligned} m_k &= \mathcal{E}[x_1^{k_1} x_2^{k_2} \ldots x_N{}^{k_N}] \\ &= (-\mathrm{j})^k \left. \frac{\partial^k \Phi(\omega_1, \omega_2, \ldots, \omega_N)}{\partial \omega_1^{k_1} \partial \omega_2^{k_2} \cdots \partial \omega_N^{k_N}} \right|_{\omega_1=\omega_2=\cdots=\omega_N=0} \end{aligned} \tag{15.112}$$

where $k=k_1+k_2+\ldots+k_N$ and the joint characteristic function is

$$\Phi(\omega_1, \omega_2, \ldots, \omega_N) = \mathcal{E}\big[\exp\big(j\omega_1 x_1 + \omega_2 x_2 + \cdots + \omega_N x_N\big)\big] \tag{15.113}$$

Now the higher-order moments can be applied for characterization of discrete-time random processes. The k^{th} order moment of a random process $x(m)$ is defined as

$$m_x(\tau_1, \tau_2, \ldots, \tau_{K-1}) = \mathcal{E}[x(m), x(m+\tau_1) x(m+\tau_2) \cdots x(m+\tau_{k-1})] \tag{15.114}$$

Note that the second-order moment $\mathcal{E}[x(m)x(m+\tau)]$ is the autocorrelation function.

Cumulants

Cumulants are similar to moments; the difference is that the moments of a random process are derived from the characteristic function $\Phi_X(\omega)$, whereas the cumulant generating function $C_X(\omega)$ is defined as the logarithm of the characteristic function as

$$C_X(\omega) = \ln \Phi_X(\omega) = \ln \mathcal{E}[\exp(j\omega x)] \tag{15.115}$$

Using a Taylor series expansion of the term $\mathcal{E}$ [exp($j\omega x$)] in Equation (15.115) the cumulant generating function can be expanded as

$$C_X(\omega) = \ln\left(1 + m_1(j\omega) + \frac{m_2}{2!}(j\omega)^2 + \frac{m_3}{3!}(j\omega)^3 + \cdots + \frac{m_n}{n!}(j\omega)^n + \cdots\right) \tag{15.116}$$

where $m_k = \mathcal{E}[x^k]$ is the k^{th} moment of the random variable x. The k^{th} order cumulant of a random variable is defined as

$$c_k = (-j)^k \left. \frac{\partial^k C_X(\omega)}{\partial \omega^k} \right|_{\omega=0} \tag{15.117}$$

From Equations (15.116) and (15.117), we have

$$c_1 = m_1 \tag{15.118}$$

$$c_2 = m_2 - m_1^2 \tag{15.119}$$

$$c_3 = m_3 - 3m_1 m_2 + 2m_1^2 \tag{15.120}$$

and so on. The general form of the k^{th} order ($k = k_1 + k_2 + \cdots + k_N$) joint cumulant generating function is

$$c_{k_1 \cdots k_N} = (-\mathrm{j})^{k_1+\cdots+k_N} \left. \frac{\partial^{k_1+\cdots+k_N} \ln\Phi_X(\omega_1,\cdots,\omega_N)}{\partial\omega_1^{k_1}\cdots\partial\omega_N^{k_N}} \right|_{\omega_1=\omega_2=\cdots=\omega_N=0} \tag{15.121}$$

The cumulants of a zero mean random process $x(m)$ are given as

$$c_x = \mathcal{E}[x(k)] = m_x = 0 \qquad \text{(mean)} \tag{15.122}$$

$$\begin{aligned} c_x(k) &= \mathcal{E}[x(m)x(m+k)] - \mathcal{E}[x(m)]^2 \\ &= m_x(k) - m_x^2 = m_x(k) \qquad \text{(covariance)} \end{aligned} \tag{15.123}$$

$$\begin{aligned} c_x(k_1,k_2) &= m_x(k_1,k_2) - m_x[m_x(k_1) + m_x(k_2) + m_x(k_2-k_1)] + 2(m_x)^3 \\ &= m_x(k_1,k_2) \qquad \text{(skewness)} \end{aligned} \tag{15.124}$$

$$\begin{aligned} c_x(k_1,k_2,k_3) = m_x(k_1,k_2,k_3) &- m_x(k_1)m_x(k_3-k_2) \\ &- m_x(k_2)m_x(k_3-k_1) - m_x(k_3)m_x(k_2-k_1) \end{aligned} \tag{15.125}$$

and so on. Note that $m_x(k_1, k_2, ..., k_N)=\mathcal{E}[x(m)x(m+k_1), x(m+k_2), ..., x(m+k_N)]$. The general formulation of the k^{th} order cumulant of a random process $x(m)$ (Rosenblatt) is defined as

$$c_x(k_1,k_2,\ldots,k_n) = m_x(k_1,k_2,\ldots,k_n) - m_x^G(k_1,k_2,\ldots,k_n) \qquad for \quad n = 3, 4, \ldots \tag{15.126}$$

where $m_x^G(k_1,k_2,\ldots,k_n)$ is the k^{th} order moment of a Gaussian process having the same mean and autocorrelation as the random process $x(m)$. From Equation (15.126), it follows that for a Gaussian process, the cumulants of order greater than 2 are identically zero.

Higher-Order Spectra

The k^{th} order spectrum of a signal $x(m)$ is defined as the $(k–1)$-dimensional Fourier transform of the k^{th} order cumulant sequence as

$$C_X(\omega_1,\dots,\omega_{k-1})=\frac{1}{(2\pi)^{k-1}}\sum_{\tau_1=-\infty}^{\infty}\cdots\sum_{\tau_{k-1}=-\infty}^{\infty}c_x(\tau_1,\dots,\tau_{k-1})e^{-j(\omega_1\tau_1+\cdots+\omega_{k-1}\tau_{k-1})} \tag{15.127}$$

For the case k=2, the second-order spectrum is the power spectrum given as

$$C_X(\omega)=\frac{1}{2\pi}\sum_{\tau=-\infty}^{\infty}c_x(\tau)e^{-j\omega\tau} \tag{15.128}$$

The *bi-spectrum* is defined as

$$C_X(\omega_1,\omega_2)=\frac{1}{(2\pi)^2}\sum_{\tau_1=-\infty}^{\infty}\sum_{\tau_2=-\infty}^{\infty}c_x(\tau_1,\tau_2)e^{-j(\omega_1\tau_1+\omega_2\tau_2)} \tag{15.129}$$

and the *tri-spectrum* is

$$C_X(\omega_1,\omega_2,\omega_3)=\frac{1}{(2\pi)^3}\sum_{\tau_1=-\infty}^{\infty}\sum_{\tau_2=-\infty}^{\infty}\sum_{\tau_3=-\infty}^{\infty}c_x(\tau_1,\tau_2,\tau_3)e^{-j(\omega_1\tau_1+\omega_2\tau_2+\omega_3\tau_3)} \tag{15.130}$$

Since the term $e^{j\omega t}$ is periodic with a period of 2π, it follows that higher order spectra are periodic in each ω_k with a period of 2π.

15.6.2 Higher-Order Spectra of Linear Time-Invariant Systems

Consider a linear time-invariant system with an impulse response sequence $\{h_k\}$, input signal $x(m)$ and output signal $y(m)$. The relation between the kth-order cumulant spectra of the input and output signals is given by

$$C_Y(\omega_1,\dots,\omega_{k-1})=H(\omega_1)\cdots H(\omega_{k-1})H^*(\omega_1+\cdots+\omega_{k-1})C_X(\omega_1,\dots,\omega_{k-1}) \tag{15.131}$$

where $H(\omega)$ is the frequency response of the linear system $\{h_k\}$. The magnitude of the kth-order spectrum of the output signal is given as

$$|C_Y(\omega_1,\ldots,\omega_{k-1})| = |H(\omega_1)|\cdots|H(\omega_{k-1})|\,|H(\omega_1+\cdots+\omega_{k-1})|\,|C_X(\omega_1,\ldots,\omega_{k-1})| \tag{15.132}$$

and the phase of the k^{th}-order spectrum is

$$\Phi_Y(\omega_1,\ldots,\omega_{k-1}) = \Phi_H(\omega_1)+\cdots+\Phi_H(\omega_{k-1}) - \Phi_H(\omega_1+\cdots+\omega_{k-1}) + \Phi_X(\omega_1,\ldots,\omega_{k-1}) \tag{15.133}$$

15.6.3 Blind Equalization Based on Higher-Order Cepstra

In this section, we consider blind equalization of a maximum-phase channel, based on higher order cepstra. Assume that the channel can be modelled by an all-zero filter, and that its z-transfer function $H(z)$ can be expressed as the product of a maximum-phase polynomial factor and a minimum-phase factor as

$$H(z) = G H_{\min}(z) H_{\max}(z^{-1}) z^{-D} \tag{15.134}$$

$$H_{\min}(z) = \prod_{i=1}^{P_1} (1-\alpha_i z^{-1}), \quad |\alpha_i|<1 \tag{15.135}$$

$$H_{\max}(z^{-1}) = \prod_{i=1}^{P_2} (1-\beta_i z), \quad |\beta_i|<1 \tag{15.136}$$

where G is a gain factor, $H_{\min}(z)$ is a minimum-phase polynomial with all its zeros inside the unit circle, $H_{\max}(z^{-1})$ is a maximum-phase polynomial with all its zeros outside the unit circle, and z^{-D} inserts D unit delays in order to make Equation (15.134) causal. The complex cepstrum of $H(z)$ is defined as

$$h_c(m) = Z^{-1}\left(\ln H(z)\right) \tag{15.137}$$

where Z^{-1} denotes the inverse z-transform. At $z=e^{j\omega}$, the z-transform is the discrete Fourier transform (DFT), and the cepstrum of a signal is obtained by taking the inverse DFT of the logarithm of the signal spectrum. In the following we consider cepstra based on the power spectrum and the higher-order spectra, and show that the higher-order cepstra have the ability to retain maximum-phase information. Assuming that the channel input $x(m)$ is a zero-mean uncorrelated process with variance σ_x^2, the power spectrum of the channel output can be expressed as

$$P_Y(\omega)=\frac{\sigma_x^2}{2\pi}H(\omega)H^*(\omega) \tag{15.138}$$

The cepstrum of the power spectrum of $y(m)$ is defined as

$$\begin{aligned} y_c(m) &= IDFT(\ln P_Y(\omega)) \\ &= IDFT\left(\ln\left(\sigma_x^2 G^2 / 2\pi\right) + \ln H_{\min}(\omega) + H_{\max}(-\omega) + \ln H^*_{\min}(\omega) + H^*_{\max}(-\omega)\right) \end{aligned} \tag{15.139}$$

where IDFT is the inverse discrete Fourier transform. Substituting Equations (15.135) and (15.36) in (15.139), the cepstrum can be expressed as

$$y_c(m)=\begin{cases} \ln\left(G^2\sigma_x^2/2\pi\right), & m=0 \\ -\left(A^{(m)}+B^{(m)}\right)/m, & m>0 \\ \left(A^{(-m)}+B^{(-m)}\right)/m, & m<0 \end{cases} \tag{15.140}$$

where $A^{(m)}$ and $B^{(m)}$ are defined as

$$A^{(m)}=\sum_{i=1}^{P_1}\alpha_i^m \tag{15.141}$$

$$B^{(m)}=\sum_{i=1}^{P_2}\beta_i^m \tag{15.142}$$

Note from Equation (15.140) that the along the index m, the maximum-phase information $B^{(m)}$ and the minimum-phase information $A^{(m)}$ overlap and cannot be separated.

Bi-Cepstrum

The bi-cepstrum of a signal is defined as the inverse Fourier transform of the logarithm of the bi-spectrum:

$$y_c(m_1,m_2)=IDFT_2[\log C_Y(\omega_1,\omega_2)] \tag{15.143}$$

where $IDFT_2[.]$ denotes the two-dimensional inverse discrete Fourier transform. The relationship between the bi-spectra of the input and output of a linear system is

$$C_Y(\omega_1,\omega_2)=H(\omega_1)H(\omega_2)H^*(\omega_1+\omega_2)C_X(\omega_1,\omega_2) \tag{15.144}$$

Assuming that the input *x(m)* of the linear time-invariant system $\{h_k\}$ is an uncorrelated non-Gaussian process, the bi-spectrum of the output can be written as

$$\begin{aligned}C_Y(\omega_1,\omega_2)=&\frac{\gamma_x^{(3)}G^3}{(2\pi)^2}H_{\min}(\omega_1)H_{\max}(-\omega_1)H_{\min}(\omega_2)H_{\max}(-\omega_2)\\&\times H^*_{\min}(\omega_1+\omega_2)H^*_{\max}(-\omega_1-\omega_2)\end{aligned} \tag{15.145}$$

where $\gamma_x^{(3)}/(2\pi)^2$ is the third-order cumulant of the uncorrelated random input process $x(m)$. Taking the logarithm of Equation (15.145) yields

$$\begin{aligned}\ln C_y(\omega_1,\omega_2)=&\ln|A|+\ln H_{\min}(\omega_1)+\ln H_{\max}(-\omega_1)+\ln H_{\min}(\omega_2)+\ln H_{\max}(-\omega_2)\\&\ln H^*_{\min}(\omega_1+\omega_2)+\ln H^*_{\max}(-\omega_1-\omega_2)\end{aligned} \tag{15.146}$$

where $A=\gamma_x^{(3)}G^3/(2\pi)^2$. The bi-cepstrum is obtained through the inverse Discrete Fourier transform of Equation (15.146) as

$$y_c(m_1,m_2)=\begin{cases}\ln|A|, & m_1=m_2=0\\ -A^{(m_1)}/m_1, & m_1>0, m_2=0\\ -A^{(m_2)}/m_2, & m_2>0, m_1=0\\ -B^{(-m_1)}/m_1, & m_1<0, m_2=0\\ B^{(-m_2)}/m_2, & m_2<0, m_1=0\\ -B^{(m_2)}/m_2, & m_1=m_2>0\\ A^{(-m_2)}/m_2, & m_1=m_2<0\\ 0, & \text{otherwise}\end{cases} \tag{15.147}$$

Note from Equation (15.147) that the maximum-phase information $B^{(m)}$ and the minimum-phase information $A^{(m)}$ are separated and appear in different regions of the bi-cepstrum indices m_1 and m_2.

The higher-order cepstral coefficients can be obtained either from the IDFT of higher-order spectra as in Equation (15.147) or using parametric methods as follows. In general, the cepstral and cumulant coefficients can be related by a convolutional equation. Pan and Nikias (1988) have shown that the recursive relation between the bi-cepstrum coefficients and the third-order cumulants of a random process is

$$y_c(m_1,m_2) * \left[-m_1 c_y(m_1,m_2)\right] = -m_1 c_y(m_1,m_2) \tag{15.148}$$

Substituting Equation (15.147) in Equation (15.148) yields

$$\sum_{i=1}^{\infty} A^{(i)}[c_x(m_1-i,m_2)-c_x(m_1+i,m_2+i)] + B^{(i)}[c_x(m_1-i,m_2-i)-c_x(m_1+i,m_2)] \\ = -m_1 c_x(m_1,m_2) \tag{15.149}$$

The truncation of the infinite summation in Equation (15.149) provides an approximate equation as

$$\sum_{i=1}^{P} A^{(i)}[c_x(m_1-i,m_2)-c_x(m_1+i,m_2+i)] \\ + \sum_{i=1}^{Q} B^{(i)}[c_x(m_1-i,m_2-i)-c_x(m_1+i,m_2)] \approx -m_1 c_x(m_1,m_2) \tag{15.150}$$

Equation (15.150) can be used to solve for the cepstral parameters $A^{(m)}$ and $B^{(m)}$.

Tri-Cepstrum

The tri-cepstrum of a signal $y(m)$ is defined as the inverse Fourier transform of the tri-spectrum:

$$y_c(m_1,m_2,m_3) = IDFT_3[\ln C_Y(\omega_1,\omega_2,\omega_3)] \tag{15.151}$$

where $IDFT_3[\cdot]$ denotes the three-dimensional inverse discrete Fourier transform. The tri-spectra of the input and output of the linear system are related by

$$C_Y(\omega_1,\omega_2,\omega_3)=H(\omega_1)H(\omega_2)H(\omega_3)H^*(\omega_1+\omega_2+\omega_3)C_X(\omega_1,\omega_2,\omega_3) \tag{15.152}$$

Assuming that the channel input $x(m)$ is uncorrelated, Equation (15.152) becomes

$$C_Y(\omega_1,\omega_2,\omega_3)=\frac{\gamma_x^{(4)}G^4}{(2\pi)^3}H(\omega_1)H(\omega_2)H(\omega_3)H^*(\omega_1+\omega_2+\omega_3) \tag{15.153}$$

where $\gamma_x^{(4)}/(2\pi)^3$ is the fourth-order cumulant of the input signal. Taking the logarithm of the tri-spectrum gives

$$\begin{aligned}\ln C_Y(\omega_1,\omega_2,\omega_3)=&\frac{\gamma_x^{(4)}G^4}{(2\pi)^3}+\ln H_{\min}(\omega_1)+\ln H_{\max}(-\omega_1)+\ln H_{\min}(\omega_2)+\ln H_{\max}(-\omega_2)\\&+\ln H_{\min}(\omega_3)+\ln H_{\max}(-\omega_3)+\ln H^*_{\min}(\omega_1+\omega_2+\omega_3)+\ln H^*_{\max}(-\omega_1-\omega_2-\omega_3)\end{aligned} \tag{15.154}$$

From Equations (15.151) and (15.154), we have

$$y_c(m_1,m_2,m_3)=\begin{cases}\ln A, & m_1=m_2=m_3=0\\ -A^{(m_1)}/m_1, & m_1>0, m_2=m_3=0\\ -A^{(m_2)}/m_2, & m_2>0, m_1=m_3=0\\ -A^{(m_3)}/m_3, & m_3>0, m_1=m_2=0\\ B^{(-m_1)}/m_1, & m_1<0, m_2=m_3=0\\ B^{(-m_2)}/m_2, & m_2<0, m_1=m_3=0\\ B^{(-m_3)}/m_3, & m_3<0, m_1=m_2=0\\ -B^{(m_2)}/m_2, & m_1=m_2=m_3>0\\ A^{(m_2)}/m_2, & m_1=m_2=m_3<0\\ 0 & \text{otherwise}\end{cases} \tag{15.155}$$

where $A=\gamma_x^{(4)}G^4/(2\pi)^3$. Note from Equation (15.155) that the maximum-phase information $B^{(m)}$ and the minimum-phase information $A^{(m)}$ are separated and appear in different regions of the tri-cepstrum indices m_1, m_2 and m_3.

Calculation of Equalizer Coefficients from the Tri-cepstrum

Assuming that the channel z-transfer function can be described by Equation (15.134), the inverse channel can be written as

$$H^{\text{inv}}(z)=\frac{1}{H(z)}=\frac{1}{H_{\min}(z)H_{\max}(z^{-1})}=H^{\text{inv}}_{\min}(z)H^{\text{inv}}_{\max}(z^{-1}) \qquad (15.156)$$

where it is assumed that the channel gain G is unity. In the time domain Equation (15.156) becomes

$$h^{\text{inv}}(m)=h^{\text{inv}}_{\min}(m)*h^{\text{inv}}_{\max}(m) \qquad (15.157)$$

Pan and Nikias (1988) describe an iterative algorithm for estimation of the truncated impulse response of the maximum-phase and the minimum-phase factors of the inverse channel transfer function. Let $\hat{h}^{\text{inv}}_{\min}(i,m)$, $\hat{h}^{\text{inv}}_{\max}(i,m)$ denote the estimates of the m^{th} coefficients of the maximum-phase and minimum-phase parts of the inverse channel at the i^{th} iteration . The Pan and Nikias algorithm is the following:

(a) Initialisation

$$\hat{h}^{\text{inv}}_{\min}(i,0)=\hat{h}^{\text{inv}}_{\max}(i,0)=1 \qquad (15.158)$$

(b) Calculation of the minimum-phase polynomial

$$\hat{h}^{\text{inv}}_{\min}(i,m)=\frac{1}{m}\sum_{k=2}^{m+1}\hat{A}^{(k-1)}\hat{h}^{\text{inv}}_{\min}(i,m-k+1) \qquad i=1, ..., P_1 \qquad (15.159)$$

(c) Calculation of the maximum-phase polynomial

$$\hat{h}^{\text{inv}}_{\max}(i,m)=\frac{1}{m}\sum_{k=m+1}^{0}\hat{B}^{(1-k)}\hat{h}^{\text{inv}}_{\max}(i,m-k+1) \qquad i=-1, ..., -P_2 \qquad (15.160)$$

The maximum-phase and minimum-phase components of the inverse channel response are combined in Equation (15.157) to give the inverse channel equalizer.

15.7 Summary

In this chapter, we considered a number of different approaches to channel equalization. The chapter began with an introduction to models for channel distortions, the definition of an ideal channel equalizer, and the problems that arise in channel equalization due to noise and possible non-invertibility of the channel. In some problems, such as speech recognition or restoration of distorted audio signals, we are mainly interested in restoring the magnitude spectrum of the signal, and phase restoration is not a primary objective. In other applications, such as digital telecommunication the restoration of both the amplitude and the timing of the transmitted symbols are of interest, and hence we need to equalise for both the magnitude and the phase distortions.

In Section 15.1, we considered the least square error Wiener equalizer. The Wiener equalizer can only be used if we have access to the channel input or the cross-correlation of the channel input and output signals.

For cases where a training signal cannot be employed to identify the channel response, the channel input is recovered through a blind equalization method. Blind equalization is feasible only if some statistics of the channel input signal are available. In Section 15.2, we considered blind equalization using the power spectrum of the input signal. This method was introduced by Stockham for restoration of the magnitude spectrum of distorted acoustic recordings. In Section 15.3, we considered a blind deconvolution method based on the factorisation of a linear predictive model of the convolved signals.

Bayesian inference provides a framework for inclusion of the statistics of the channel input and perhaps also those of the channel environment. In Section 15.4, we considered Bayesian equalization methods, and studied the case where the channel input is modelled by a set of hidden Markov models. Section 15.5 introduced channel equalization methods for removal of intersymbol interference in digital telecommunication systems, and finally in Section 15.6, we considered the use of higher-order spectra for equalization of non-minimum-phase channels.

Bibliography

BENVENISTE A., GOURSAT M. and RUGET G. (1980) Robust Identification of a Non-minimum Phase System: Blind Adjustment of Linear Equalizer in Data Communications. IEEE Trans, Automatic Control, **AC-25**, pp. 385–399.

BELLINI S. (1986) Bussgang Techniques for Blind Equalization. IEEE GLOBECOM Conf. Rec., pp. 1634–1640.

BELLINI S. and ROCCA F. (1988) Near Optimal Blind Deconvolution. IEEE Proc. Int. Conf. Acoustics, Speech, and Signal Processing. ICASSP-88, pp. 2236–2239.

BELFIORE C.A. and PARK J.H. (1979) Decision Feedback Equalization. Proc. IEEE, 67, pp. 1143–1156.

GERSHO A. (1969) Adaptive Equalization of Highly Dispersive Channels for Data Transmission. Bell System Technical Journal, **48**, pp. 55–70.

GODARD D.N. (1974) Channel Equalization using a Kallman Filter for Fast Data Transmission. IBM J. Res. Dev., **18**, pp. 267–273.

GODARD D.N. (1980) Self-recovering Equalization and Carrier Tracking in a Two-Dimensional Data Communication System. IEEE Trans. Comm., **COM-28**, pp. 1867-75.

HANSON B.A. and APPLEBAUM T.H. (1993) Subband or Cepstral Domain Filtering for Recognition of Lombard and Channel-Distorted Speech. IEEE Int. Conf. Acoustics, Speech and Signal Processing, pp. 79–82.

HARIHARAN S. and CLARK A.P. (1990) HF Channel Estimation using a Fast Transversal Filter Algorithm. IEEE Trans. Acoustics, Speech and Signal Processing, **38**, pp. 1353–1362.

HATZINAKO S.D.(1990) Blind Equalization Based on Polyspectra. Ph.D. Thesis, Northeastern University, Boston, MA.

HERMANSKY H. and MORGAN N. (1992) Towards Handling the Acoustic Environment in Spoken Language Processing. Int. Conf. on Spoken Language Processing Tu.fPM.1.1, pp. 85–88.

LUCKY R.W. (1965) Automatic Equalization of Digital Communications. Bell System technical Journal, **44**, pp. 547–588.

LUCKY R.W. (1965) Techniques for Adaptive Equalization of Digital Communication Systems. Bell System Technical Journal, **45**, pp. 255–286.

MENDEL J.M. (1990) Maximum Likelihood Deconvolution: A Journey into Model Based Signal Processing. Springer-Verlag, New York.

MENDEL J.M. (1991) Tutorial on Higher Order Statistics (Spectra) in Signal Processing and System Theory: Theoretical results and Some Applications. Proc. IEEE, **79**, pp. 278–305.

MOKBEL C., MONNE J. and JOUVET D. (1993) On-Line Adaptation of A Speech Recogniser to Variations in Telephone Line Conditions, Proc. 3rd European Conf. On Speech Communication and Technoplogy. EuroSpeech-93, **2**, pp. 1247-1250.

MONSEN P. (1971) Feedback Equalization for Fading Dispersive Channels. IEEE Trans. Information Theory, **IT-17**, pp. 56–64.

NIKIAS C.L. and CHIANG H.H. (1991) Higher-Order Spectrum Estimation via Non-Causal Autoregressive Modeling and Deconvolution. IEEE Trans. Acoustics, Speech and Signal Processing, **ASSP-36**, pp. 1911–1913.

NOWLAN S.J. and HINTON G.E. (1993) A Soft Decision-Directed Algorithm for Blind Equalization IEEE Transactions on Communications., **41**, No. 2, pp. 275–279.

PAN R. and NIKIAS C.L. (1988) Complex Cepstrum of Higher Order Cumulants and Non-minimum Phase Identification. IEEE Trans. Acoustics, Speech and Signal Processing, **ASSP-36**, pp. 186–205.

PICCHI G. and PRATI G. (1987) Blind Equalization and Carrier Recovery using a Stop-and-Go Decision-Directd Algorithm, IEEE Trans. Commun, **COM-35**, pp. 877–887.

RAGHUVEER M.R. and NIKIAS C.L. (1985) Bispectrum Estimation: A Parameteric Approach. IEEE Trans. Acoustics, Speech, and Signal Processing, **ASSP-33**, **5**, pp. 35–48.

ROSENBLATT M. (1985) Stationary Sequences and Random Fields. Birkhauser, Boston, MA.

SPENCER P.S. and RAYNER P.J.W. (1990) ,Separation of Stationary and Time-Varying Systems and Its Applications to the Restoration of Gramophone Recordings. Ph.D. Thesis, Cambridge University.

STOCKHAM T.G., CANNON T.M. and INGEBRETSEN R.B (1975) Blind Deconvolution Through Digital Signal Processing. IEEE Proc., **63**, **4**, pp. 678-92.

QURESHI S.U. (1985) Adaptive Equalization. IEEE Proc. Vol 73, No. 9, pp. 1349–1387.

UNGERBOECK G. (1972) Theory on the Speed of Convergence in Adaptive Equalizers for Digital Communication. IBM J. Res. Dev., **16**, pp. 546-555.