Chapter 4

Intersymbol Interference and Equalization

The all-pass assumption made in the AWGN (or non-dispersive) channel model is rarely practical. Due to the scarcity of the frequency spectrum, we usually filter the transmitted signal to limit its bandwidth so that efficient sharing of the frequency resource can be achieved. Moreover, many practical channels are bandpass and, in fact, they often respond differently to inputs with different frequency components, i.e., they are *dispersive*. We have to refine the simple AWGN (or non-dispersive) model to accurately represent this type of practical channels. One such commonly employed refinement is the *dispersive* channel model[1]:

$$r(t) = u * h_c(t) + n(t), \tag{4.1}$$

where $u(t)$ is the transmitted signal, $h_c(t)$ is the impulse response of the channel, and $n(t)$ is AWGN with power spectral density $N_0/2$. In essence, we model the dispersive characteristic of the channel by the linear filter $h_c(t)$. The simplest dispersive channel is the *bandlimited channel* for which the channel impulse response $h_c(t)$ is that of an ideal lowpass filter. This lowpass filtering smears the transmitted signal in time causing the effect of a symbol to spread to adjacent symbols when a sequence of symbols are transmitted. The resulting interference, *intersymbol interference (ISI)*, degrades the error performance of the communication system. There are two major ways to mitigate the detrimental effect of ISI. The first method is to design bandlimited transmission pulses which minimize the the

[1] For simplicity, all the signals considered in this chapter are real baseband signals. All the developments here can, of course, be generalized to bandpass signals using either the real bandpass representation or the complex baseband representation.

effect of ISI. We will describe such a design for the simple case of bandlimited channels. The ISI-free pulses obtained are called the *Nyquist pulses*. The second method is to filter the received signal to cancel the ISI introduced by the channel impulse response. This approach is generally known as *equalization*.

4.1 Intersymbol Interference

To understand what ISI is, let us consider the transmission of a sequence of symbols with the basic waveform $u(t)$. To send the nth symbol b_n, we send $b_n u(t - nT)$, where T is the symbol interval. Therefore, the transmitted signal is

$$\sum_n b_n u(t - nT). \tag{4.2}$$

Based on the dispersive channel model, the received signal is given by

$$r(t) = \sum_n b_n v(t - nT) + n(t), \tag{4.3}$$

where $v(t) = u * h_c(t)$ is the received waveform for a symbol. If a single symbol, say the symbol b_0, is transmitted, the optimal demodulator is the one that employs the matched filter, i.e., we can pass the received signal through the matched filter $\tilde{v}(t) = v(-t)$ and then sample the matched filter output at time $t = 0$ to obtain the decision statistic. When a sequence of symbols are transmitted, we can still employ this matched filter to perform demodulation. A reasonable strategy is to sample the matched filter output at time $t = mT$ to obtain the decision statistic for the symbol b_m. At $t = mT$, the output of the matched filter is

$$\begin{aligned} z_m &= \sum_n b_n v * \tilde{v}(mT - nT) + n_m \\ &= b_m \|v\|^2 + \sum_{n \neq m} b_n v * \tilde{v}(mT - nT) + n_m, \end{aligned} \tag{4.4}$$

where n_m is a zero-mean Gaussian random variable with variance $N_0\|v\|^2/2$. The first term in (4.4) is the desired signal contribution due to the symbol b_m and the second term contains contributions from the other symbols. These unwanted contributions from other symbols are called *intersymbol interference (ISI)*.

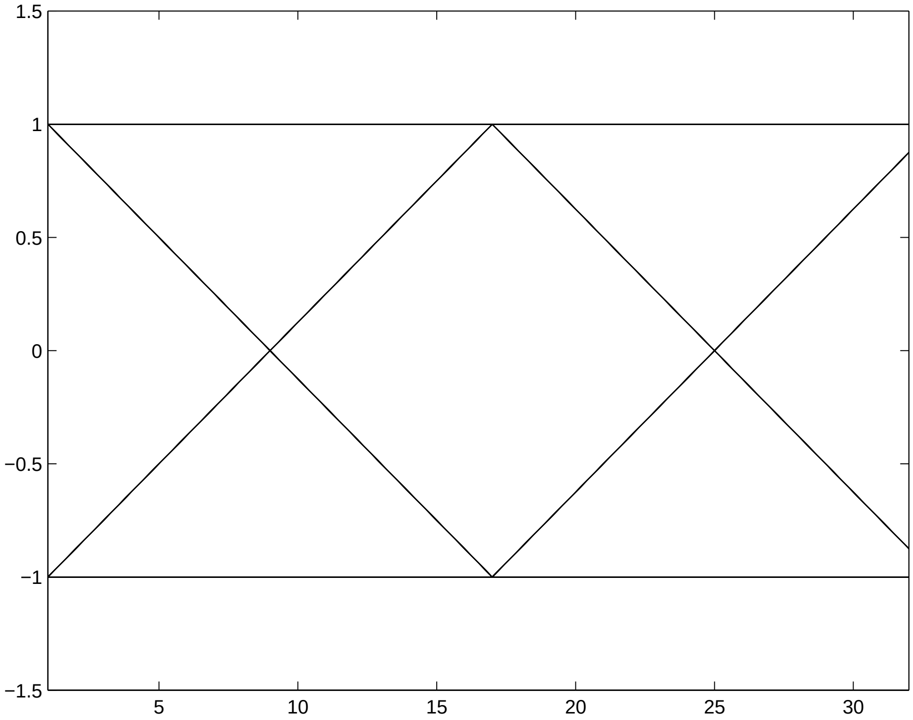

Figure 4.1: Eye diagram of a BPSK signal with no ISI

Suppose $v(t)$ were timelimited, i.e., $v(t) = 0$ except for $0 \leq t \leq T$. Then it is easy to see that $v*\tilde{v}(t) = 0$ except for $-T < t < T$. Therefore, $v*\tilde{v}(mT-nT) = 0$ for all $n \neq m$, and there is no ISI. As a result, the demodulation strategy above can be interpreted as matched filtering for each symbol. Unfortunately, a timelimited waveform is never bandlimited. Therefore, for a bandlimited channel, $v(t)$ and, hence, $v * \tilde{v}(t)$ are not timelimited and hence ISI is, in general, present. One common way to observe and measure (qualitatively) the effect of ISI is to look at the *eye diagram* of the received signal. The effect of ISI and other noises can be observed on an oscilloscope displaying the output of the matched filter on the vertical input with horizontal sweep rate set at multiples of $1/T$. Such a display is called an eye diagram. For illustration, let us consider the basic waveform $u(t)$ is the rectangular pulse $p_T(t)$ and binary signaling is employed. The eye diagrams for the cases where the channel is all-pass (no ISI) and lowpass (ISI present) are shown in Figures 4.1 and 4.2, respectively. The effect of ISI is to cause a reduction in the eye opening by reducing the peak as well as causing ambiguity in the timing information.

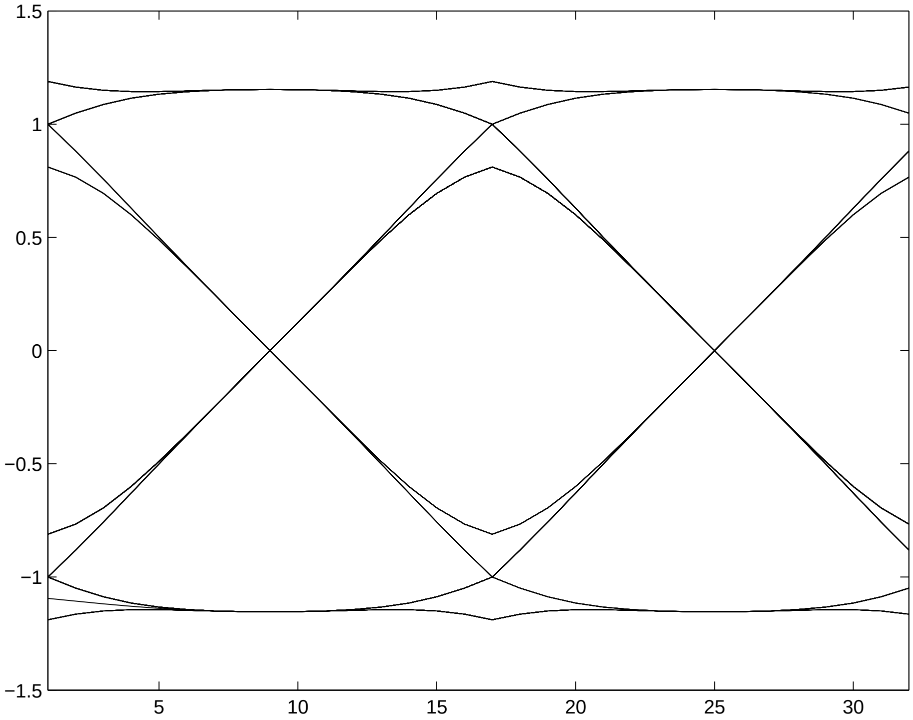

Figure 4.2: Eye diagram of a BPSK signal with ISI

4.2 Nyquist Pulses

A careful observation on (4.4) reviews that it is possible to have no ISI even if the $v(t)$ is bandlimited, i.e., the basic pulse shape $u(t)$ and/or the channel is bandlimited. More precisely, letting $x(t) = v*\tilde{v}(t)$, we can rewrite the decision statistic z_m in (4.4) as:

$$z_m = b_m x(0) + \sum_{n \neq m} b_n x(mT - nT) + n_m. \tag{4.5}$$

There is no ISI if the *Nyquist condition* is satisfied:

$$x(nT) = \begin{cases} c & \text{for} \quad n = 0, \\ 0 & \text{for} \quad n \neq 0, \end{cases} \tag{4.6}$$

where c is some constant and, without loss of generality, we can set $c = 1$. The Nyquist condition in this form is not very helpful in the design of ISI-free pulses. It turns out that it is more illustrative to restate the Nyquist condition in frequency domain. To do so, first let

$$x_\delta(t) = \sum_{n=-\infty}^{\infty} x(nT)\delta(t - nT). \tag{4.7}$$

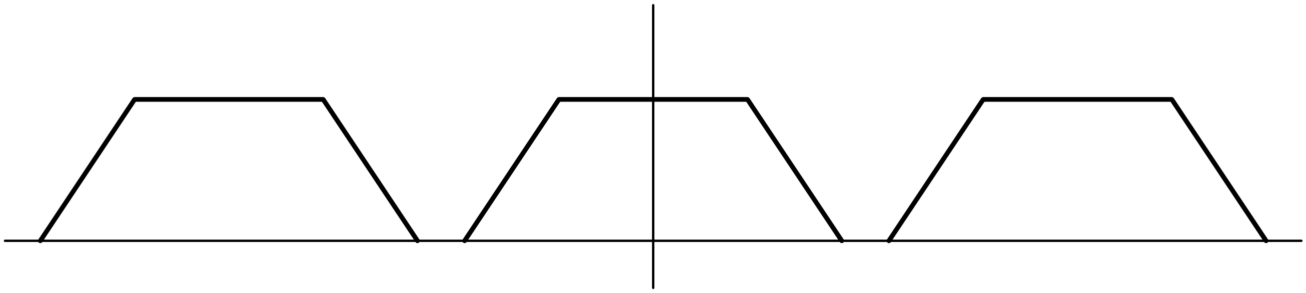

Figure 4.3: Case $\frac{1}{T} > 2W$: non-overlapping spectrum

Taking Fourier transform,

$$X_\delta(f) = \frac{1}{T}\sum_{n=-\infty}^{\infty} X\Big(f - \frac{n}{T}\Big), \tag{4.8}$$

where $X(f)$ is the Fourier transform of $x(t)$. The Nyquist condition in (4.6) is equivalent to the condition $x_\delta(t) = \delta(t)$ or $X_\delta(f) = 1$ in frequency domain. Now, by employing (4.8), we get

$$\sum_{n=-\infty}^{\infty} X\Big(f - \frac{n}{T}\Big) = T. \tag{4.9}$$

This is the equivalent Nyquist condition in frequency domain. It says that the folded spectrum of $x(t)$ has to be flat for not having ISI.

When the channel is bandlimited to WHz, i.e., $X(f) = 0$ for $|f| > W$, the Nyquist condition has the following implications:

- Suppose that the symbol rate is so high that $\frac{1}{T} > 2W$. Then, the folded spectrum $\sum_{n=-\infty}^{\infty} X(f - \frac{n}{T})$ looks like the one in Figure 4.3. There are gaps between copies of $X(f)$. No matter how $X(f)$ looks, the Nyquist condition cannot be satisfied and ISI is inevitable.

- Suppose that the symbol rate is slower so that $\frac{1}{T} = 2W$. Then, copies of $X(f)$ can just touch their neighbors. The folded spectrum $\sum_{n=-\infty}^{\infty} X(f - \frac{n}{T})$ is flat if and only if

$$X(f) = \begin{cases} T & \text{for} \quad |f| < W, \\ 0 & \text{for} \quad \text{otherwise}. \end{cases} \tag{4.10}$$

The corresponding time domain function is the sinc pulse

$$x(t) = \text{sinc}(\pi t/T). \tag{4.11}$$

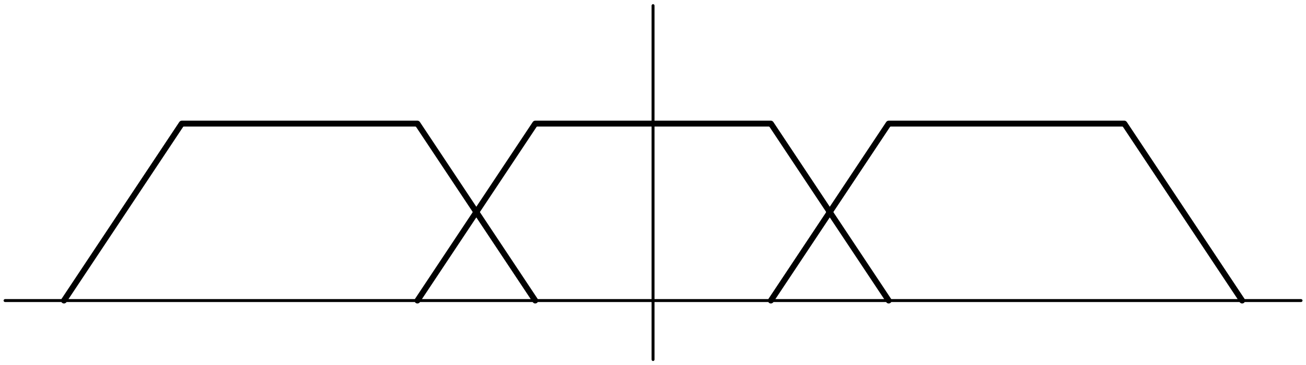

Figure 4.4: Case $\frac{1}{T} < 2W$: overlapping spectrum

We note that the sinc pulse is not timelimited and is not causal. Therefore, it is not physically realizable. A truncated and delayed version is used as an approximation. The critical rate $\frac{1}{T} = 2W$ above which ISI is unavoidable is known as the *Nyquist rate*.

- Suppose that the symbol rate is even slower so that $\frac{1}{T} < 2W$. Then, copies of $X(f)$ overlap with their neighbors. The folded spectrum $\sum_{n=-\infty}^{\infty} X(f - \frac{n}{T})$ can be flat with many different choices of $X(f)$. An example is shown in Figure 4.4. Therefore, we can design an ISI-free pulse shape which gives a flat folded spectrum.

When the symbol rate is below the Nyquist rate, a widely used ISI-free spectrum is the raised-cosine spectrum (Figure 4.5):

$$X(f) = \begin{cases} T & \text{for} \quad 0 \le |f| \le \frac{1-\alpha}{2T}, \\ \frac{T}{2}\left[1 + \cos\frac{\pi T}{\alpha}(|f| - \frac{1-\alpha}{2T})\right] & \text{for} \quad \frac{1-\alpha}{2T} \le |f| \le \frac{1+\alpha}{2T}, \\ 0 & \text{for} \quad |f| > \frac{1+\alpha}{2T}, \end{cases} \tag{4.12}$$

where $0 \le \alpha \le 1$ is called the *roll-off factor*. It determines the *excess bandwidth*[2] beyond $\frac{1}{2T}$. The corresponding time domain function (Figure 4.6) is:

$$x(t) = \frac{\sin(\pi t/T)}{\pi t/T} \frac{\cos(\pi \alpha t/T)}{1 - 4\alpha^2 t^2/T^2}. \tag{4.13}$$

When $\alpha = 0$, it reduces to the sinc function. We note that for $\alpha > 0$, $x(t)$ decays as $1/t^3$ while for $\alpha = 0$, $x(t)$ (sinc pulse) decays as $1/t$. Hence, the raised cosine spectrum gives a pulse that is much

[2] Another way to interpret this is that to fit $X(f)$ into a channel bandlimited to W, $\frac{1+\alpha}{2T} < W$. Therefore, the larger α is, the larger T has to be and the slower is the symbol rate.

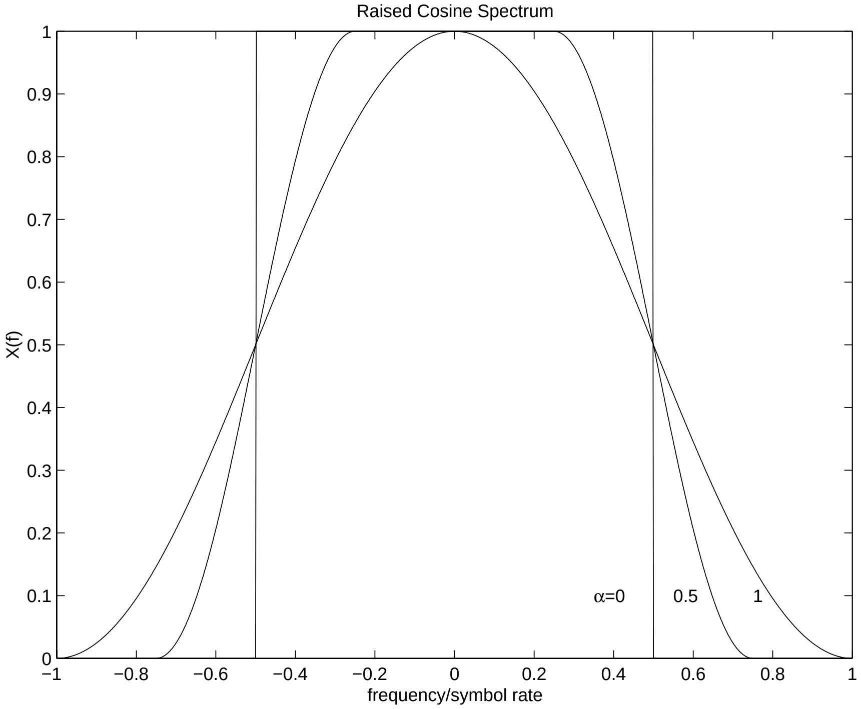

Figure 4.5: Raised-cosine spectrum

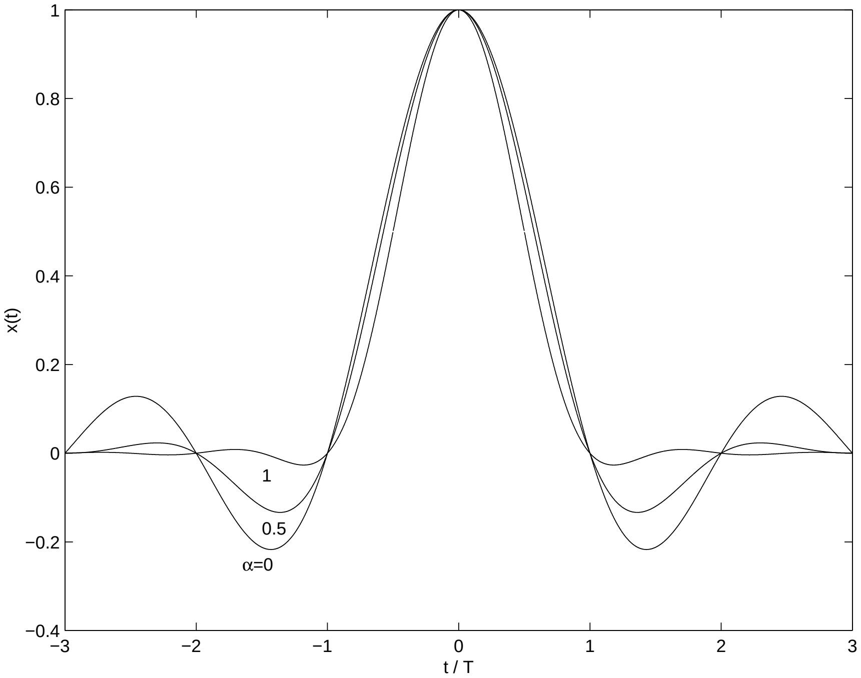

Figure 4.6: Time domain function of the raised-cosine spectrum

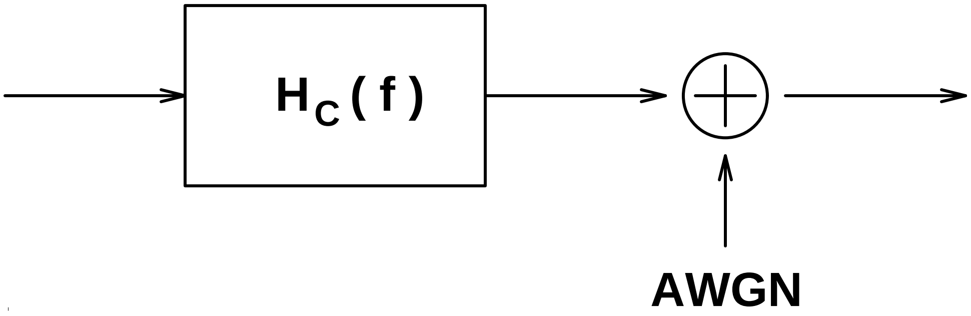

Figure 4.7: Dispersive channel model

less sensitive to timing errors than the sinc pulse. Just like all other bandlimited pulses, $x(t)$ from the raised cosine spectrum is not timelimited. Therefore, truncation and delay is required for realization.

Finally, recall that $x(t)$ is the overall response of the transmitted pulse passing through the bandlimited channel and the receiving filter. Mathematically,

$$\begin{aligned} X(f) &= |V(f)|^2 \\ &= |H_c(f)U(f)|^2, \end{aligned} \tag{4.14}$$

where $U(f)$, $V(f)$, and $H_c(f)$ are the Fourier transform of $u(t)$, $v(t)$, and $h_c(t)$, respectively. Given that an ISI-free spectrum $X(f)$ is chosen, we can employ (4.14) to obtain the simple case of a bandlimited channel, i.e., the channel does not introduce any distortion within its passband, we can simply choose $U(f)$ to be $\sqrt{X(f)}$. Then the Fourier transform the transfer function of the matched filter is also $\sqrt{X(f)}$. For example, if the raised-cosine spectrum is chosen, the resulting ISI-free pulse $u(t)$ is called the *square-root raised-cosine pulse*. Of course, suitable truncation and delay are required for physical realization.

4.3 Equalization

For many physical channels, such as telephone lines, not only are they bandlimited, but they also introduce distortions in their passbands. Such a channel can be modeled by an LTI filter followed by an AWGN source as shown in Figure 4.7. This is the dispersive channel model we describe before. In general, ISI is often introduced. For a communication system system employing a linear modulation, such as BPSK, through a dispersive channel, the whole system can be described the conceptual model

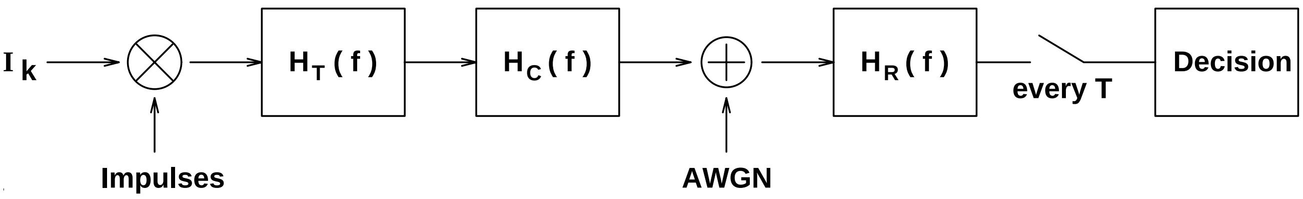

Figure 4.8: Continuous-time communication model over a dispersive channel

in Figure 4.8, in which the sequence of information symbols is denoted by $\{I_k\}$ and $H_T(f)$, $H_C(f)$, and $H_R(f)$ are the transfer functions of the transmission (pulse-shaping) filter, the dispersive channel and the receiving filter[3], respectively. The Nyquist condition for no ISI developed in Section 4.2 can be easily generalized to the above communication system. Letting $X(f) = H_T(f)H_C(f)H_R(f)$, the condition for no ISI is that the folded spectrum $X(f)$, is constant for all frequencies, i.e.,

$$\sum_{n=-\infty}^{\infty} X\left(f - \frac{n}{T}\right) = T. \tag{4.15}$$

One method to achieve the Nyquist condition is to fix the receiving filter to be the matched filter, i.e., set $H_R(f) = H_T^*(f)H_C^*(f)$, and choose the transmission filter so that (4.15) is satisfied. This is the Nyquist pulse design method described in Section 4.2. The major disadvantage of this pulse-shaping method is that it is in general difficult to construct the appropriate analog filters for $H_T(f)$ and $H_R(f)$ in practice. Moreover, we have to know the channel response $H_c(f)$ in advance to construct the transmission and receiving filters.

An alternative method is to fix the transmission filter[4] and choose the receiving filter $H_R(f)$ to satisfy the condition in (4.15). As for the previous method, it is also difficult to build the appropriate analog filter $H_R(f)$ to eliminate ISI. However, notice that what we want eventually are the samples at intervals T at the receiver. Therefore, we may choose to build a simpler (practical) filter $H_R(f)$, take samples at intervals T, and put a digital filter, called *equalizer*, at the output to eliminate ISI as shown below in Figure 4.9. This approach to remove ISI is usually known as *equalization*. The main advantage of this approach is that a digital filter is easy to build and is easy to alter for different equalization schemes, as well as to fit different channel conditions.

[3] If the matched filter demodulation technique is employed, the receiving filter $H_R(f)$ is chosen to be the complex conjugate of $H_T(f)H_C(f)$.

[4] Of course, we should choose one that can be easily built.

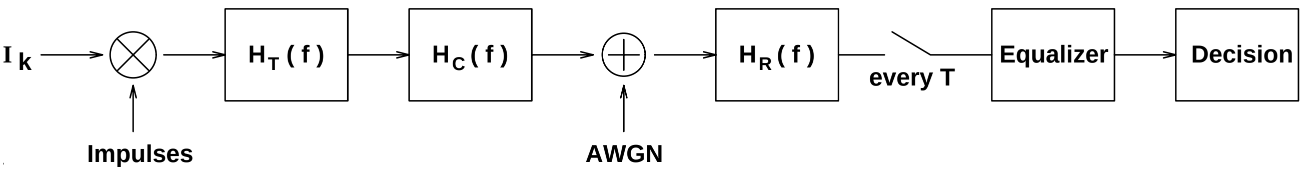

Figure 4.9: Communication system with equalizer

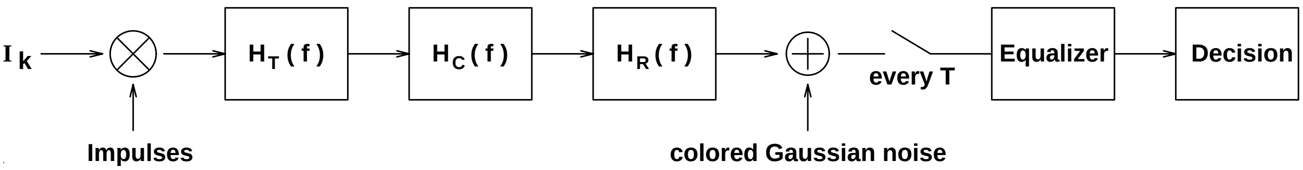

Figure 4.10: Equivalent communication system with colored Gaussian noise

4.3.1 Equivalent discrete-time model

Our goal is to design the equalizer which can remove (or suppress) ISI. To do so, we translate the continuous-time communication system model in Figure 4.9 to an equivalent discrete-time model that is easier to work with. The following steps describe the translation process:

- Instead of considering AWGN being added before the receiving filter $H_R(f)$, we can consider an equivalent colored Gaussian noise being added after $H_R(f)$ when we analyze the system. The equivalent colored noise is the output of $H_R(f)$ due to AWGN. The resulting model is shown in Figure 4.10.

- We input a bit or a symbol to the communication system every T seconds, and get back a sample at the output of the sampler every T seconds. Therefore, we can represent the communication system in Figure 4.10 from the information source to the sampler as a digital filter. Since $H_T(f)$, $H_C(f)$ and $H_R(f)$ are LTI filters, they can be combined and represented by an equivalent digital LTI filter. Denote its transfer function by $H(z)$ and its impulse response by $\{h_k\}_{k=-\infty}^{\infty}$. The result is the *discrete time-linear filter model* shown in Figure 4.11, in which the output sequence I'_k is given by

$$\begin{aligned} I'_k &= \sum_j I_j h_{k-j} + n_k \\ &= I_k h_0 + \sum_{j\neq k} I_j h_{k-j} + n_k. \end{aligned} \tag{4.16}$$

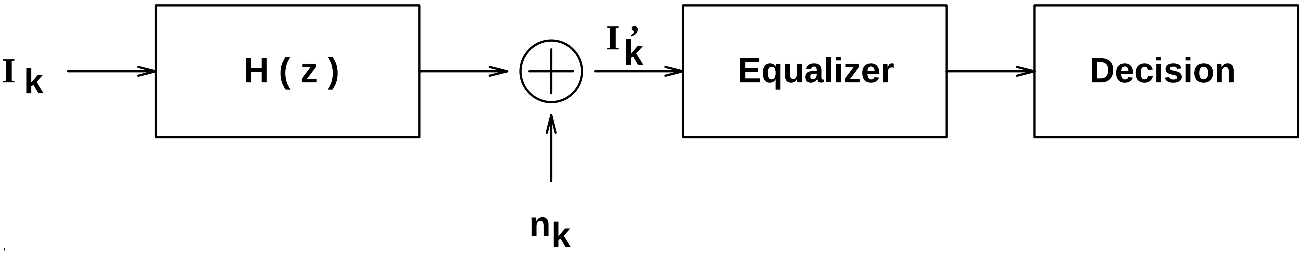

Figure 4.11: Equivalent discrete-time communication system model with colored noise

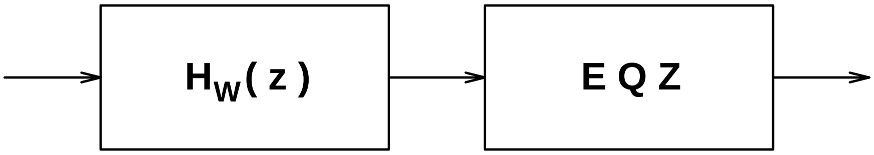

Figure 4.12: Typical equalizer

In general, $h_j \neq 0$ for some $j \neq 0$. Therefore, ISI is present. Notice that the noise sequence $\{n_k\}$ consists of samples of the colored Gaussian noise (AWGN filtered by $H_R(f)$), and is not white in general.

- Usually, the equalizer consists of two parts, namely, a *noise-whitening* digital filter $H_W(z)$ and an equalizing circuit that equalizes the noise-whitened output as shown in Figure 4.12. The effect of $H_W(z)$ is to "whiten" the noise sequence so that the noise samples are uncorrelated. Notice that $H_W(z)$ depends only on $H_R(f)$, and can be determined a prior according to our choice of $H_R(f)$. At the output of $H_W(z)$, the noise sequence is white. Therefore, equivalently, we can consider the equivalent discrete-time model shown in Figure 4.13, in which $\{\tilde{n}_k\}$ is an AWGN sequence.

- Let $G(z) = H(z)H_W(z)$. The communication system from the information source to the output of the noise whitening filter can now be represented by the *discrete-time white-noise linear filter model* in Figure 4.14. The output sequence $\tilde{I}_k$ is given by

$$\begin{aligned} \tilde{I}_k &= \sum_j I_j g_{k-j} + \tilde{n}_k \\ &= I_k g_0 + \sum_{j \neq k} I_j g_{k-j} + \tilde{n}_k, \end{aligned} \tag{4.17}$$

where $\{g_k\}$ is the impulse response corresponding to the transfer function $G(z)$, and $\{\tilde{n}_k\}$ is an

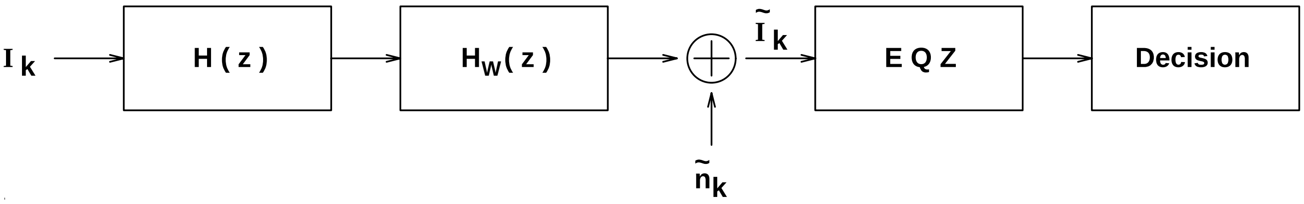

Figure 4.13: Equivalent discrete-time communication system model with white noise

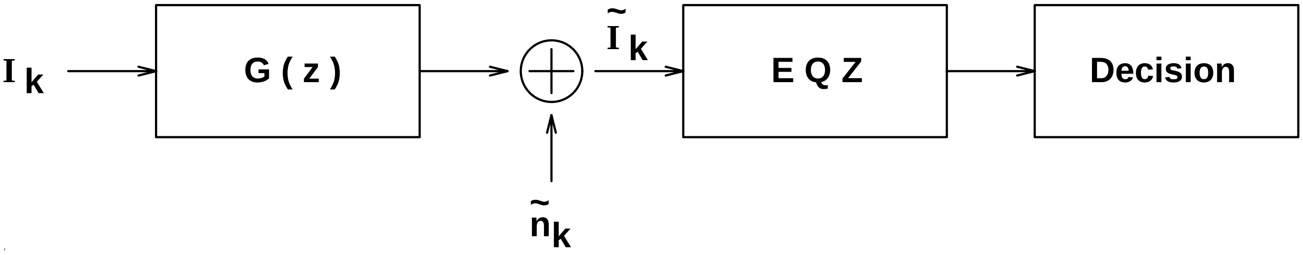

Figure 4.14: Equivalent discrete-time white-noise linear filter model

AWGN sequence. We will work with this discrete-time model in all the following sections.

- Finally, the equalizing circuit (we simply call it the equalizer from now on) attempts to remove ISI from the output of $G(z)$. The focus of our coming discussion is the design of this equalizer. Suppose that the equalizer is also an LTI filter with transfer function $H_E(z)$ and corresponding impulse response $\{h_{E,j}\}$. Then the output of the equalizer is given by

$$\hat{I}_k = \sum_j \tilde{I}_{k-j} h_{E,j}. \tag{4.18}$$

Ideally, $\hat{I}_k$ contains only contributions from the current symbol I_k and the AWGN sequence with small variance.

4.3.2 Zero-forcing equalizer

First, let us consider the use of a linear equalizer, i.e., we employ an LTI filter with transfer function $H_E(z)$ as the equalizing circuit. The simplest way to remove the ISI is to choose $H_E(z)$ so that the output of the equalizer gives back the information sequence, i.e., $\hat{I}_k = I_k$ for all k if noise is not present. This can be achieved by simply setting the transfer function $H_E(z) = 1/G(z)$. This method is called *zero-forcing* equalization since the ISI component at the equalizer output is forced to zero.

In general, the corresponding impulse response $\{h_{E,k}\}$ can be an infinite length sequence. Suitable truncation and delay is applied to get an approximation.

We note that the effect of the equalizing filter on the noise is neglected in the development of the zero-forcing equalizer above. In reality, noise is always present. Although the ISI component is forced to zero, there may be a chance that the equalizing filter will greatly enhancing the noise power and hence the error performance of the resulting receiver will still be poor. To see this, let us evaluate the signal-to-noise ratio at the output of the zero-forcing equalizer when the transmission filter $H_T(f)$ is fixed and the matched filter is used as the receiving filter, i.e.,

$$H_R(f) = H_T^*(f)H_C^*(f). \tag{4.19}$$

In this case, it is easy to see that the digital filter $H(z)$ is given by

$$H\left(e^{\mathrm{j}2\pi fT}\right) = \frac{1}{T}\sum_{n=-\infty}^{\infty}\left|H_T\left(f-\frac{n}{T}\right)H_C\left(f-\frac{n}{T}\right)\right|^2, \tag{4.20}$$

and the PSD of the colored Gaussian noise samples n_k in Figure 4.11 is given by

$$\Phi_{n_k}\left(e^{\mathrm{j}2\pi fT}\right) = \frac{N_0}{2T}\sum_{n=-\infty}^{\infty}\left|H_T\left(f-\frac{n}{T}\right)H_C\left(f-\frac{n}{T}\right)\right|^2. \tag{4.21}$$

Hence, the noise-whitening filter $H_W(z)$ can be chosen as

$$H_W\left(e^{\mathrm{j}2\pi fT}\right) = \frac{1}{\sqrt{H\left(e^{\mathrm{j}2\pi fT}\right)}}, \tag{4.22}$$

and then the PSD of the whitened-noise samples $\tilde{n}_k$ is simply $N_0/2$. As a result, the overall digital filter $G(z)$ in Figure 4.14 is

$$G\left(e^{\mathrm{j}2\pi fT}\right) = H\left(e^{\mathrm{j}2\pi fT}\right)H_W\left(e^{\mathrm{j}2\pi fT}\right) = \sqrt{H\left(e^{\mathrm{j}2\pi fT}\right)}. \tag{4.23}$$

Now, we choose the zero-forcing filter $H_E(z)$ as

$$H_E\left(e^{\mathrm{j}2\pi fT}\right) = \frac{1}{G\left(e^{\mathrm{j}2\pi fT}\right)} = \frac{1}{\sqrt{H\left(e^{\mathrm{j}2\pi fT}\right)}}. \tag{4.24}$$

Since the zero-forcing filter simply inverts the effect of the channel on the original information symbols I_k, the signal component at its output should be exactly I_k. If we model the I_k as iid random variables with zero mean and unit variance, then the PSD of the signal component is 1 and hence the signal energy at the output of the equalizer is just $\int_{-1/2T}^{1/2T} df = 1/T$. On the other hand, the PSD of the noise component at the output of the equalizer is $\frac{N_0}{2}\left|H_E\left(e^{\mathrm{j}2\pi fT}\right)\right|^2$. Hence the noise energy at the equalizer output is $\int_{-1/2T}^{1/2T}\frac{N_0}{2}\left|H_E\left(e^{\mathrm{j}2\pi fT}\right)\right|^2 df$. Defining the SNR as the ratio of the signal energy to the noise energy, we have

$$\text{SNR} = \left\{\frac{N_0T^2}{2}\int_{-1/2T}^{1/2T}\left[\sum_{n=-\infty}^{\infty}\left|H_T\left(f-\frac{n}{T}\right)H_C\left(f-\frac{n}{T}\right)\right|^2\right]^{-1} df\right\}^{-1}. \tag{4.25}$$

Notice that the SNR depends on the folded spectrum of the signal component at the input of the receiver. If there is a certain region in the folded spectrum with very small magnitude, then the SNR can be very poor.

4.3.3 MMSE equalizer

The zero-forcing equalizer, although removes ISI, may not give the best error performance for the communication system because it does not take into account noises in the system. A different equalizer that takes noises into account is the minimum mean square error (MMSE) equalizer. It is based on the mean square error (MSE) criterion.

Without knowing the values of the information symbols I_k beforehand, we model each symbol I_k as a random variable. Assume that the information sequence $\{I_k\}$ is WSS. We choose a linear equalizer $H_E(z)$ to minimize the MSE between the original information symbols I_k and the output of the equalizer $\hat{I}_k$:

$$\text{MSE} = \text{E}\left[e_k^2\right] = \text{E}\left[(I_k-\hat{I}_k)^2\right]. \tag{4.26}$$

Let us employ the FIR filter of order $2L+1$ shown in Figure 4.15 as the equalizer. We note that a delay of L symbols is incurred at the output of the FIR filter. Then

$$\begin{aligned}\text{MSE} &= \text{E}\left[\left(I_k-\sum_{j=-L}^{L}\tilde{I}_{k-j}h_{E,j}\right)^2\right]\\ &= \text{E}\left[\left(I_k-\tilde{\mathbf{I}}_k^T\mathbf{h}_E\right)^2\right],\end{aligned} \tag{4.27}$$

where

$$\tilde{\mathbf{I}}_k = \left[\tilde{I}_{k+L},\ldots,\tilde{I}_{k-L}\right]^T, \tag{4.28}$$

$$\mathbf{h}_E = \left[h_{E,-L},\ldots,h_{E,L}\right]^T. \tag{4.29}$$

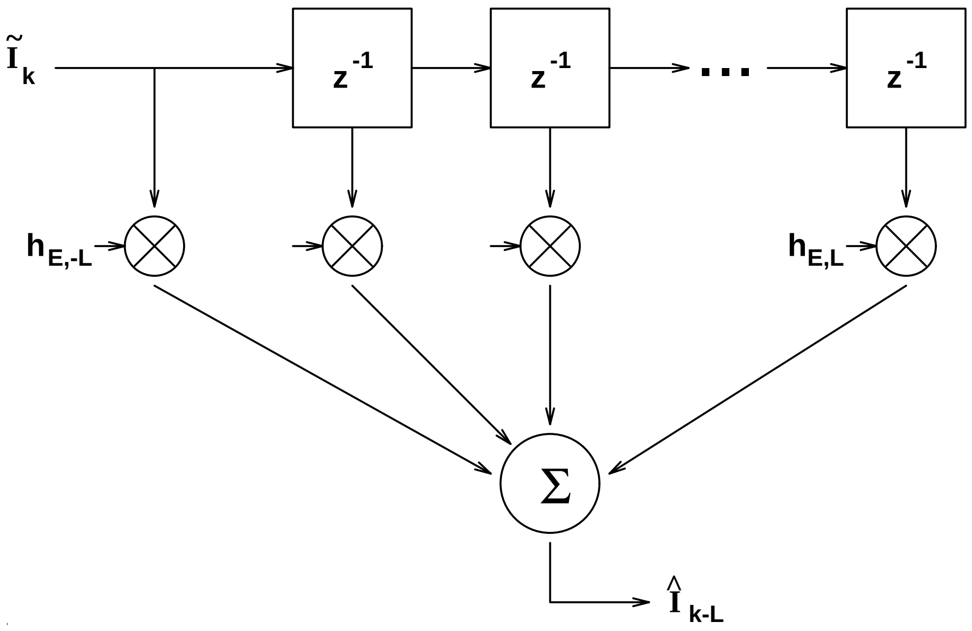

Figure 4.15: FIR filter as an MMSE equalizer

We want to minimize MSE by suitable choices of $h_{E,-L}, \ldots, h_{E,L}$. Differentiating with respect to each $h_{E,j}$ and setting the result to zero, we get

$$\mathrm{E}\left[\tilde{\mathbf{I}}_k\left(I_k - \tilde{\mathbf{I}}_k^T\mathbf{h}_E\right)\right] = \mathbf{0}. \tag{4.30}$$

Rearranging, we get

$$\mathbf{R}\mathbf{h}_E = \mathbf{d}, \tag{4.31}$$

where

$$\mathbf{R} = \mathrm{E}\left[\tilde{\mathbf{I}}_k\tilde{\mathbf{I}}_k^T\right], \tag{4.32}$$

$$\mathbf{d} = \mathrm{E}\left[I_k\tilde{\mathbf{I}}_k\right]. \tag{4.33}$$

If $\mathbf{R}$ and $\mathbf{d}$ are available, then the MMSE equalizer can be found by solving the linear matrix equation (4.31). It can be shown that the signal-to-noise ratio at the output of the MMSE equalizer is better than that of the zero-forcing equalizer.

The linear MMSE equalizer can also be found iteratively. First, notice that the MSE is a quadratic function of $\mathbf{h}_E$. The gradient of the MSE with respect to $\mathbf{h}_E$ gives the direction to change $\mathbf{h}_E$ for the largest increase of the MSE. In our notation, the gradient is $-2(\mathbf{d} - \mathbf{R}\mathbf{h}_E)$. To decrease the MSE, we

can update $\mathbf{h}_E$ in the direction opposite to the gradient. This is the *steepest descent algorithm*:
At the kth step, the vector $\mathbf{h}_E(k)$ is updated as

$$\mathbf{h}_E(k) = \mathbf{h}_E(k-1) + \mu[\mathbf{d} - \mathbf{R}\mathbf{h}_E(k-1)] \tag{4.34}$$

where μ is a small positive constant that controls the rate of convergence to the optimal solution.

In many applications, we do not know $\mathbf{R}$ and $\mathbf{d}$ in advance. However, the transmitter can transmit a *training sequence* that is known *a priori* by the receiver. With a training sequence, the receiver can estimate $\mathbf{R}$ and $\mathbf{d}$. Alternatively, with a training sequence, we can replace $\mathbf{R}$ and $\mathbf{d}$ at each step in the steepest descent algorithm by the rough estimates $\tilde{\mathbf{I}}_k\tilde{\mathbf{I}}_k^T$ and $I_k\tilde{\mathbf{I}}_k$, respectively. The algorithm becomes:

$$\mathbf{h}_E(k) = \mathbf{h}_E(k-1) + \mu\left[I_k - \tilde{\mathbf{I}}_k^T\mathbf{h}_E(k-1)\right]\tilde{\mathbf{I}}_k. \tag{4.35}$$

This is a stochastic steepest descent algorithm called the *least mean square (LMS) algorithm*.

4.3.4 LS equalizer

In the training period for the MMSE equalizer, the "data" sequence, i.e., the training sequence is known to the equalizer. Instead of minimizing the MSE, which is a statistical average, we can actually minimize the sum of the square errors. This is called the *least squares (LS)* criterion. Suppose that the known sequence lasts for K symbols. Then the sum of the square errors is given by

$$\begin{aligned} e_K^2 &= \sum_{k=1}^{K}(I_k - \hat{I}_k)^2 \\ &= \sum_{k=1}^{K}\left[I_k - \tilde{\mathbf{I}}_k^T\mathbf{h}_E(K)\right]^2. \end{aligned} \tag{4.36}$$

Differentiating with respect to $\mathbf{h}_E(K)$ and setting the result to zero, we get

$$\mathbf{R}(K)\mathbf{h}_E(K) = \mathbf{d}(K). \tag{4.37}$$

This time,

$$\mathbf{R}(K) = \sum_{k=1}^{K}\tilde{\mathbf{I}}_k\tilde{\mathbf{I}}_k^T, \tag{4.38}$$

$$\mathbf{d}(K) = \sum_{k=1}^{K}I_k\tilde{\mathbf{I}}_k. \tag{4.39}$$

Suppose that we are given one more training symbol. Apparently, we have to recalculate $\mathbf{R}(K+1)$ and $\mathbf{d}(K+1)$, and solve the matrix equation all over again. However, actually, there is a more efficient approach. Assuming $\mathbf{R}(K)$ is non-singular, $\mathbf{h}_E(K) = \mathbf{R}^{-1}(K)\mathbf{d}(K)$. Notice that

$$\begin{aligned} \mathbf{R}(K+1) &= \mathbf{R}(K) + \tilde{\mathbf{I}}_{K+1}\tilde{\mathbf{I}}_{K+1}^T, && (4.40) \\ \mathbf{d}(K+1) &= \mathbf{d}(K) + I_{K+1}\tilde{\mathbf{I}}_{K+1}. && (4.41) \end{aligned}$$

Using the matrix inversion lemma[5], we get

$$\begin{aligned} \mathbf{R}^{-1}(K+1) &= \mathbf{R}^{-1}(K) - \frac{\mathbf{R}^{-1}(K)\tilde{\mathbf{I}}_{K+1}\tilde{\mathbf{I}}_{K+1}^T\mathbf{R}^{-1}(K)}{1+\tilde{\mathbf{I}}_{K+1}^T\mathbf{R}^{-1}(K)\tilde{\mathbf{I}}_{K+1}}, && (4.42) \\ \mathbf{h}_E(K+1) &= \mathbf{h}_E(K) + \frac{I_{K+1} - \tilde{\mathbf{I}}_{K+1}^T\mathbf{h}_E(K)}{1+\tilde{\mathbf{I}}_{K+1}^T\mathbf{R}^{-1}(K)\tilde{\mathbf{I}}_{K+1}}\mathbf{R}^{-1}(K)\tilde{\mathbf{I}}_{K+1}. && (4.43) \end{aligned}$$

The procedure is called the *recursive least squares (RLS) algorithm*. In many cases, the RLS algorithm converges much faster than the steepest descent algorithm at the expense of more complex computation.

4.3.5 Decision feedback equalizer

Recall from the equivalent discrete-time model in Figure 4.14 that

$$\tilde{I}_k = I_k g_0 + \sum_{j\neq k} I_j g_{k-j} + \tilde{n}_k. \qquad (4.44)$$

The current symbol we want to determine is I_k. If we had known the other symbols exactly, an obvious approach to eliminate ISI would be to subtract their effects off[6], i.e., the equalizer would give

$$\hat{I}_k = \tilde{I}_k - \sum_{j\neq k} I_j g_{k-j}. \qquad (4.45)$$

In general, we do not know all the symbols that are affecting the reception of the current symbol. However, it is possible to use previously decided symbols (output from the decision device) provided that we have made correct decisions on them. This approach is called *decision feedback equalization*. With decision feedback, we can think of the equalizer to contain two parts — a *feedforward* part and a

[5] Assuming all the required invertibilities, $(\mathbf{A}-\mathbf{BDC})^{-1} = \mathbf{A}^{-1} + \mathbf{A}^{-1}\mathbf{B}\left(\mathbf{D}^{-1} - \mathbf{CA}^{-1}\mathbf{B}\right)^{-1}\mathbf{CA}^{-1}$.

[6] Of course, we need to have knowledge of $G(z)$ to be able to do so.

feedback part. Suppose that the feedforward filter is of order $L_1 + 1$ and the feedback filter is of order L_2. Then

$$\hat{I}_k = \sum_{j=-L_1}^{0} \tilde{I}_{k-j} h_{E,j} + \sum_{j=1}^{L_2} I^d_{k-j} h_{E,j}, \tag{4.46}$$

where I^d_j are the decided symbols. Again, the filter coefficients $h_{E,j}$ can be found by minimizing the MSE. In general, significant improvement over linear equalizers can be obtained with the decision feedback equalizer.

Consider a DFE with a feedforward filter of order $L_1 + 1$ and a feedback filter of order L_2. Assume perfect decision feedback, i.e., $I^d_j = I_j$. Then

$$\hat{I}_k = \mathbf{I}_F^T \mathbf{h}_{E,F} + \mathbf{I}_B^T \mathbf{h}_{E,B} \tag{4.47}$$

where

$$\begin{aligned}
\mathbf{I}_F &= \left[\tilde{I}_{k+L_1}\ \tilde{I}_{k+L_1-1}\ \ldots\ \tilde{I}_k\right]^T, && (4.48)\\
\mathbf{I}_B &= \left[I_{k-1}\ I_{k-2}\ \ldots\ I_{k-L_2}\right]^T, && (4.49)\\
\mathbf{h}_{E,F} &= \left[h_{E,-L_1}\ h_{E,-L_1+1}\ \ldots\ h_{E,0}\right]^T, && (4.50)\\
\mathbf{h}_{E,B} &= \left[h_{E,1}\ h_{E,2}\ \ldots\ h_{E,L_2}\right]^T. && (4.51)
\end{aligned}$$

Further assume that the data symbols I_k are zero-mean unit-variance iid random variables. We seek the filters $\mathbf{h}_{E,F}$ and $\mathbf{h}_{E,B}$ that minimize the MSE given by

$$\mathrm{E}\left[(I_k - \hat{I}_k)^2\right] = \mathrm{E}\left[(I_k - \mathbf{I}_F^T \mathbf{h}_{E,F} - \mathbf{I}_B^T \mathbf{h}_{E,B})^2\right]. \tag{4.52}$$

Differentiating with respect to $\mathbf{h}_{E,F}$ and $\mathbf{h}_{E,B}$, we get

$$\begin{aligned}
\mathrm{E}\left[\mathbf{I}_F(I_k - \mathbf{I}_F^T \mathbf{h}_{E,F} - \mathbf{I}_B^T \mathbf{h}_{E,B})\right] &= \mathbf{0}, && (4.53)\\
\mathrm{E}\left[\mathbf{I}_B(I_k - \mathbf{I}_F^T \mathbf{h}_{E,F} - \mathbf{I}_B^T \mathbf{h}_{E,B})\right] &= \mathbf{0}. && (4.54)
\end{aligned}$$

Notice that $\mathrm{E}[I_k \mathbf{I}_B] = 0$ and $\mathrm{E}[\mathbf{I}_B \mathbf{I}_B^T] = \mathbf{I}_{L_2 \times L_2}$, i.e., the identity matrix. The equations for optimal $\mathbf{h}_{E,F}$ and $\mathbf{h}_{E,B}$ reduce to

$$\begin{aligned}
\mathrm{E}[\mathbf{I}_F \mathbf{I}_F^T]\mathbf{h}_{E,F} + \mathrm{E}[\mathbf{I}_F \mathbf{I}_B^T]\mathbf{h}_{E,B} &= \mathrm{E}[I_k \mathbf{I}_F], && (4.55)\\
\mathrm{E}[\mathbf{I}_B \mathbf{I}_F^T]\mathbf{h}_{E,F} + \mathbf{h}_{E,B} &= \mathbf{0}. && (4.56)
\end{aligned}$$

Solving these equations, we have

$$\begin{aligned} \mathbf{h}_{E,F} &= \left(\mathrm{E}[\mathbf{I}_F\mathbf{I}_F^T] - \mathrm{E}[\mathbf{I}_F\mathbf{I}_B^T]\mathrm{E}[\mathbf{I}_B\mathbf{I}_F^T]\right)^{-1}\mathrm{E}[I_k\mathbf{I}_F], && (4.57)\\ \mathbf{h}_{E,B} &= -\mathrm{E}[\mathbf{I}_B\mathbf{I}_F^T]\mathbf{h}_{E,F}. && (4.58)\end{aligned}$$

Similar to the case of the MMSE equalizer, we can also solve for the feedforward and feedback filters using the steepest descent approach. If we do not know the expectations of the matrices above *a priori*, we can send a training sequence to facilitate the estimation of them.

4.4 Maximum Likelihood Sequence Receiver

Whenever it is not practical to construct ISI-free transmission pulses, we can use an equalizer to eliminate (or reduce) ISI, and then make a decision on the current symbol based on the equalizer output. Although this approach is simple and practical, we have no idea whether it is optimal in terms of minimizing the average symbol error probability. In fact, it turns out that all the equalization methods discussed in the previous section are not optimal. Because of the fact that the effect of a symbol is spread to other symbols, it is intuitive that the optimal receiver should observe not only the segment of received signal concerning the desired symbol, but the whole received signal instead. As a matter of fact, this strategy is also employed in the equalization techniques described previously[7]. Using the whole received signal, we can employ the MAP principle to develop the optimal *symbol-by-symbol detector*, which decides one transmitted symbol at a time, to minimize the average symbol error probability. The development is rather involved and the optimal symbol-by-symbol detector is usually too complex to implement. Here, we opt for another possibility.

Instead of deciding a transmitted symbol at a time, we can consider to decide the whole transmitted symbol sequence simultaneously from the received signal. In this way, we aim at minimizing the probability of choosing the wrong sequence of symbols instead of the average symbol error probability. With this sequence detection approach, we can employ the ML principle to achieve our goal. The resulting “optimal” receiver is referred to as the *maximum likelihood sequence (MLS) receiver*.

[7] We assume the use of FIR filters for the MMSE, LS, and decision-feedback equalizers. In these cases, we are not strictly observing the whole received signal, but some long segments of it.

Now, let us develop the MLS receiver. Suppose we transmit a sequence of information bits $\{b_n\}_{n=0}^{\infty}$ with rate $1/T$ symbols per second and the transmission and channel filters are both causal. The received signal up to time $(K+1)T$, i.e., for $t < (K+1)T$, is

$$r(t) = \sum_{n=0}^{K} b_n v(t-nT) + n(t). \tag{4.59}$$

This is simply a rewrite of (4.3) with the above assumptions applied. As mentioned above, we try to decide the sequence of transmitted symbols $\{b_n\}_{n=0}^{K}$ based on our observation $r(t)$ up to time $(K+1)T$[8]. To do so, we treat the whole summation on the right hand side of (4.59) as our desired signal[9]. With this point of view, we are equivalently considering an M-ary communication system over the AWGN channel with $M = 2^K$ exhausting all the possible transmitted bit sequences. Each bit sequence is treated as an M-ary "symbol". This is the problem we have solved. Assuming all the bit sequences are equally probable, we can employ the ML receiver to minimize the probability of making an error when determining the transmitted sequence. From the results in Section 2.5.3, we know that the ML estimator of the bit sequence

$$\mathbf{b}_K = [b_K, b_{K-1}, \ldots, b_0] \tag{4.60}$$

up to time $(K+1)T$ is

$$\begin{aligned}
\hat{\mathbf{b}}_K &= \arg\min_{\mathbf{b}_K} \int_{-\infty}^{\infty} \left[r(t) - \sum_{n=0}^{K} b_n v(t-nT) \right]^2 dt \\
&= \arg\max_{\mathbf{b}_K} \left\{ \int_{-\infty}^{\infty} r(t) \sum_{n=0}^{K} b_n v(t-nT) dt - \frac{1}{2} \int_{-\infty}^{\infty} \left[\sum_{n=0}^{K} b_n v(t-nT) \right]^2 dt \right\}.
\end{aligned} \tag{4.61}$$

Defining $y_n = r * \tilde{v}(nT)$ and $x_n = x(nT) = v * \tilde{v}(nT)$, we can write the metric as

$$c_K(\mathbf{b}_K) = \sum_{n=0}^{K} b_n y_n - \frac{1}{2} \sum_{n=0}^{K} \sum_{m=0}^{K} b_n b_m x_{n-m}. \tag{4.62}$$

Then the MLS receiver, which is depicted in Figure 4.16, decides

$$\hat{\mathbf{b}}_K = \arg\max_{\mathbf{b}_K} c_K(\mathbf{b}_K). \tag{4.63}$$

In order to make the decision, the MLS receiver has to check all the 2^K sequences to find the one

[8] We cannot decide the future symbols until we have observed them.

[9] Ironically, although our goal is to combat ISI, there is no interference in this interpretation.

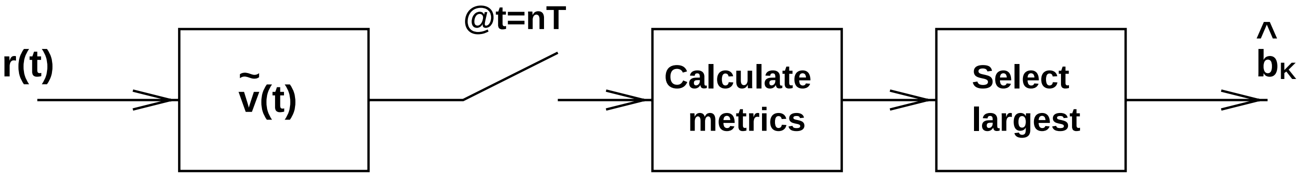

Figure 4.16: Maximum likelihood sequence receiver

that gives the largest metric. Hence its complexity increases exponentially with time. This is clearly impractical and this is why the relatively simple equalization techniques described in the previous section are more popular in practice.

When x_n is of finite-support[10], i.e., $x_n = 0$ for $|n| > L$, a significant simplification can be applied to the calculation of the maximum metric rendering the MLS receiver practical. First, we can update the metric in the following way:

$$c_K(\mathbf{b}_K) = c_{K-1}(\mathbf{b}_{K-1}) + b_K \left(y_K - \sum_{n=K-L}^{K-1} b_n x_{n-K} - \frac{1}{2} b_K x_0 \right). \tag{4.64}$$

We are going to make use of two important observations from (4.64) to simplify the maximization of the metric. The first observation is that the updating part (the second term on the right hand side) depends only on

$$\mathbf{s}_K = [b_{K-1}, b_{K-2}, \ldots, b_{K-L}]. \tag{4.65}$$

If we decompose $\mathbf{b}_K$ into $\mathbf{b}_K = [b_K, \mathbf{s}_K, \mathbf{b}_{K-L-1}]$, (4.64) can be rewritten as

$$c_K(\mathbf{b}_K) = c_{K-1}(\mathbf{s}_K, \mathbf{b}_{K-L-1}) + p(y_K, b_K, \mathbf{s}_K), \tag{4.66}$$

where

$$p(y_K, b_K, \mathbf{s}_K) = b_K \left(y_K - \sum_{n=K-L}^{K-1} b_n x_{n-K} - \frac{1}{2} b_K x_0 \right). \tag{4.67}$$

Recall our goal is to maximize $c_K(\mathbf{b}_K)$. From (4.66)

$$\begin{aligned} \max_{\mathbf{b}_K} c_K(\mathbf{b}_K) &= \max_{b_K, \mathbf{s}_K, \mathbf{b}_{K-L-1}} \left[c_{K-1}(\mathbf{s}_K, \mathbf{b}_{K-L-1}) + p(y_K, b_K, \mathbf{s}_K) \right] \\ &= \max_{b_K, \mathbf{s}_K} \left[m_K(\mathbf{s}_K) + p(y_K, b_K, \mathbf{s}_K) \right], \end{aligned} \tag{4.68}$$

[10] This is generally not true for a channel with a bandlimited response unless some form of transmission pulse design is applied. However, even when x_n is not strictly of finite-support, the simplification here can still be applied approximately provided x_n decays fast enough.

where

$$m_K(\mathbf{s}_K) = \max_{\mathbf{b}_{K-L-1}} c_{K-1}(\mathbf{s}_K, \mathbf{b}_{K-L-1}). \tag{4.69}$$

The second equality above is due to our second important observation that if $\mathbf{b}_K = [b_K, \mathbf{s}_K, \mathbf{b}_{K-L-1}]$ is the sequence that maximizes $c_K(\cdot)$, then the segment $\mathbf{b}_{K-L-1}$ must be the one that maximizes $c_{K-1}(\mathbf{s}_K, \cdot)$. Indeed, suppose that there is another segment $\mathbf{b}'_{K-L-1}$ such that $c_{K-1}(\mathbf{s}_K, \mathbf{b}'_{K-L-1}) > c_{K-1}(\mathbf{s}_K, \mathbf{b}_{K-L-1})$. Then it is clear[11] from (4.66) that $c_K(\mathbf{b}'_K) > c_K(\mathbf{b}_K)$, where $\mathbf{b}'_K = [b_K, \mathbf{s}_K, \mathbf{b}'_{K-L-1}]$. This contradicts with our original claim that $\mathbf{b}_K$ maximizes $c_K(\cdot)$. This observation reveals the important simplification in the maximization of $c_K(\cdot)$ that we do not need to test all the 2^K patterns of $\mathbf{b}_K$ as long as we know the values of $m_K(\mathbf{s}_K)$ for all the 2^L patterns of $\mathbf{s}_K$. The order of complexity of the maximization reduces dramatically from 2^K to 2^L, which does not grow with time.

However, there is still a problem needed to be solved before we can benefit from this large complexity reduction. We need to calculate $m_K(\mathbf{s}_K)$ for each of the 2^L patterns of $\mathbf{s}_K$. Fortunately, this task can be efficiently accomplished by an iterative algorithm known as the *Viterbi algorithm*. First, let each possible pattern of $\mathbf{s}_{K-1} = [b_{K-2}, b_{K-3}, \ldots, b_{K-L-1}]$ represents a state of the communication system at the time instant $t = (K-1)T$. There are altogether 2^L states. When the new input bit b_K arrives at time $t = KT$, the system changes state from $\mathbf{s}_{K-1}$ to $\mathbf{s}_K = [b_{K-1}, b_{K-2}, \ldots, b_{K-L}]$. When the received sample y_K is available at the output of the matched filter in Figure 4.16, the receiver calculates $c_K(\mathbf{b}_K)$ making use of (4.68). Hence, all the receiver needs to know is the value of $m_K(\mathbf{s}_K)$ for each state $\mathbf{s}_K$. But

$$\begin{aligned}
m_K(\mathbf{s}_K) &= \max_{\mathbf{b}_{K-L-1}} c_{K-1}(\mathbf{b}_{K-1}) \\
&= \max_{b_{K-L-1}, \mathbf{b}_{K-L-2}} [c_{K-2}(\mathbf{s}_{K-1}, \mathbf{b}_{K-L-2}) + p(y_{K-1}, b_{K-1}, \mathbf{s}_{K-1})] \\
&= \max_{b_{K-L-1}} \left[\max_{\mathbf{b}_{K-L-2}} c_{K-2}(\mathbf{s}_{K-1}, \mathbf{b}_{K-L-2}) + p(y_{K-1}, b_{K-1}, \mathbf{s}_{K-1}) \right] \\
&= \max_{b_{K-L-1}} \left[m_{K-1}(\mathbf{s}_{K-1}) + p(y_{K-1}, b_{K-1}, \mathbf{s}_{K-1}) \right]. \qquad (4.70)
\end{aligned}$$

Therefore, during the transition from state $\mathbf{s}_{K-1}$ to state $\mathbf{s}_K$, we can use (4.70) to update and store $m_K(\mathbf{s}_K)$ and the $\mathbf{b}_{K-L-1}$ that maximizes it:

$$\hat{\mathbf{b}}_{K-L-1}(\mathbf{s}_K) = \arg\max_{b_{K-L-1}} m_K(\mathbf{s}_K)$$

[11] The important observation here is that $p(y_K, b_K, \mathbf{s}_K)$ does not depends on $\mathbf{b}_{K-L-1}$.

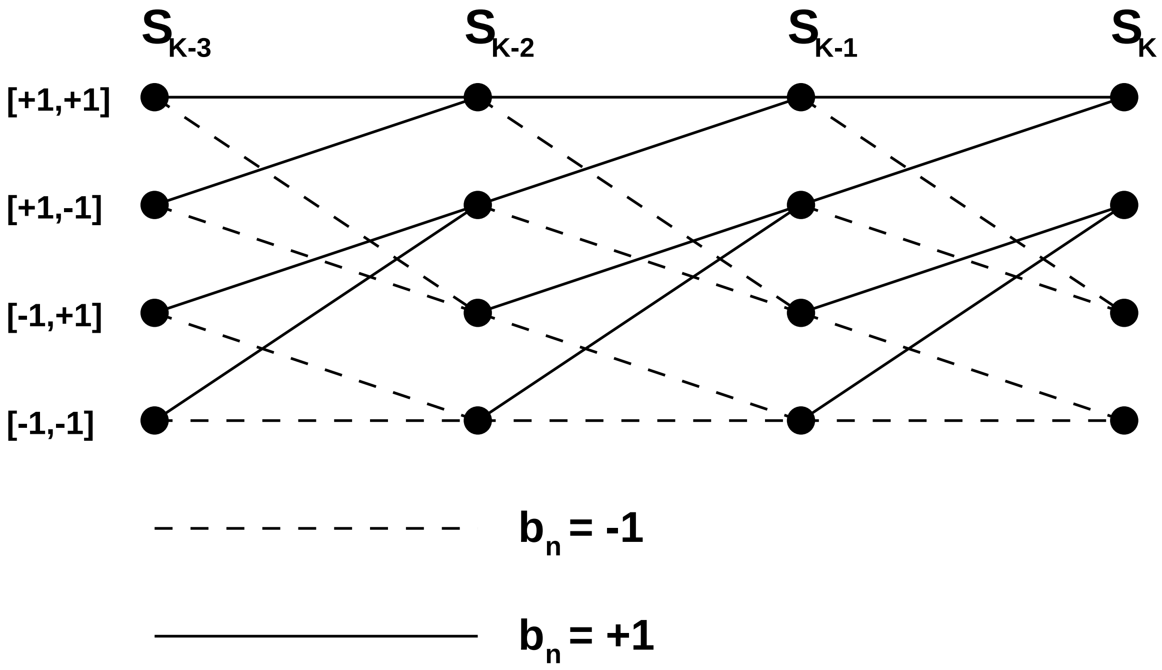

Figure 4.17: Trellis diagram for the case of $L = 2$

$$= \left[\arg\max_{b_{K-L-1}} \left\{m_{K-1}(\mathbf{s}_{K-1}) + p(y_{K-1}, b_{K-1}, \mathbf{s}_{K-1})\right\}, \hat{\mathbf{b}}_{K-L-2}(\mathbf{s}_{K-1})\right], \tag{4.71}$$

with $\hat{\mathbf{b}}_{K-L-2}(\mathbf{s}_{K-1})$ as the sequence $\mathbf{b}_{K-L-2}$ that maximizes $m_{K-1}(\mathbf{s}_{K-1})$. The other previous information can be discarded. This update is usually performed in a "backward" manner. For each state $\mathbf{s}_K = [b_{K-1}, b_{K-2}, \ldots, b_{K-L}]$, we know that the transition can only come from two previous states, namely $\mathbf{s}'_{K-1} = [b_{K-2}, b_{K-3}, \ldots, b_{K-L}, 1]$ and $\mathbf{s}''_{K-1} = [b_{K-2}, b_{K-3}, \ldots, b_{K-L}, -1]$. Therefore,

$$m_K(\mathbf{s}_K) = \max\left\{m_{K-1}(\mathbf{s}'_{K-1}) + p(y_{K-1}, b_{K-1}, \mathbf{s}'_{K-1}), m_{K-1}(\mathbf{s}''_{K-1}) + p(y_{K-1}, b_{K-1}, \mathbf{s}''_{K-1})\right\}. \tag{4.72}$$

If $\mathbf{s}'_{K-1}$ is the one chosen above, then $\hat{\mathbf{b}}_{K-L-1}(\mathbf{s}_K) = [1, \hat{\mathbf{b}}_{K-L-2}(\mathbf{s}'_{K-1})]$. Otherwise, $\hat{\mathbf{b}}_{K-L-1}(\mathbf{s}_K) = [-1, \hat{\mathbf{b}}_{K-L-2}(\mathbf{s}'_{K-1})]$. We perform this update for all the 2^L states $\mathbf{s}_K$ at state transition to make the needed values of $m_K(\mathbf{s}_K)$ for the determination of

$$\hat{\mathbf{b}}_K = \left[\arg\max_{b_K, \mathbf{s}_K}\{m_K(\mathbf{s}_K) + p(y_K, b_K, \mathbf{s}_K)\},\ \max_{\mathbf{s}_K} \hat{\mathbf{b}}_{K-L-1}(\mathbf{s}_K)\right]. \tag{4.73}$$

The update relations of the states are depicted by the *trellis diagram*. The trellis diagram for the case of $L = 2$ is given in Figure 4.17.

Let us end our discussion on the MLS receiver by making two remarks. First, the Viterbi algorithm described above can be easily generalized to the cases of non-binary symbols. Second, although the MLS receiver minimizes only the sequence error probability, its symbol error performance is usually better than that of the equalization techniques.