Contents

Chapter 4

Multi-channel Modulation

Multi-channel modulation methods are generally the best for data transmission channels with moderate or severe intersymbol interference. The modulation and equalization methods studied in Chapter 3 have only mildly considered the channel pulse response - that is, the modulator basis functions were fixed. Chapter 3 studied equalization for mitigation of distortion (ISI) that might not have been of concern in data transmission with carefully designed channel-dependent basis functions. In practice, a particular modem may have to work with acceptable performance over a range of variation in specific transmission channels. For instance, there are nearly a billion telephone lines worldwide that exhibit significant variation in channel response and noise power spectral density. Also, cable-television coaxial links can also exhibit significant variation from customer to customer. Large performance improvement can be attained in these applications when the modulator basis functions can be designed as a function of measured channel characteristics. Multi-channel modulation methods allow such modulator adaptation to channel.

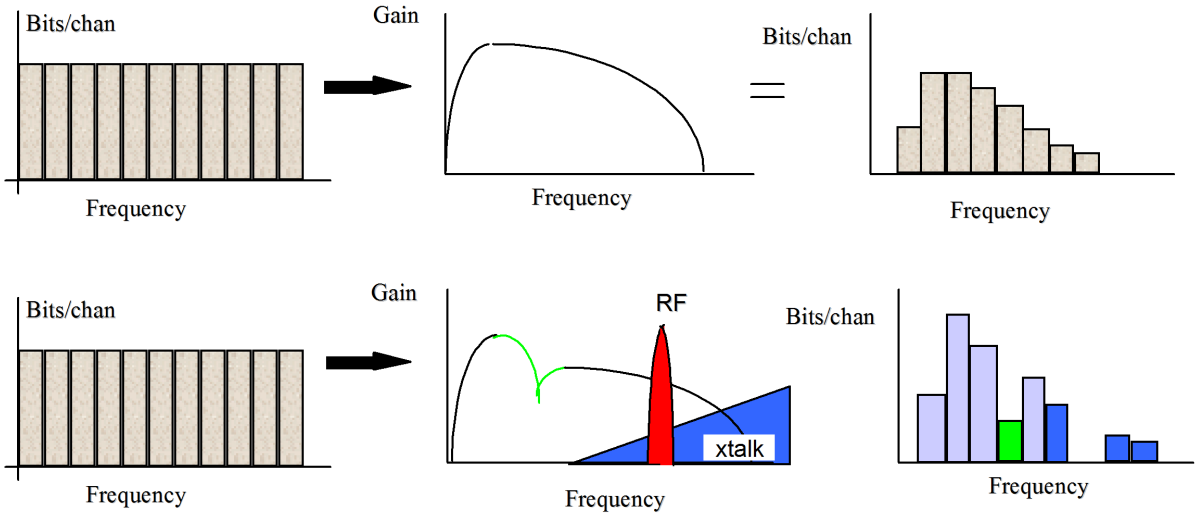


Figure 4.1: Basic concept of adaptive multi-channel ("multitone") transmission.

Figure 4.1 illustrates the basic operation of DSL circuits, which use multitone transmission. The basis functions can be viewed as a set of non-overlapping narrow QAM signals (i.e., "tones"). In the upper portion of Figure 4.1, the channel response is shown to remove low-frequencies (DC notching of transformer coupling to the telephone line) and also attenuates high-frequencies. The noise is presumed

AWGN. The energy and/or amount of information carried on each QAM channel roughly follows the channel transfer function. Multitone designs thus avoid need for equalization on any "tone" by making each sufficiently narrow that the associated channel response in that band appears "flat," so it satisfies the Nyquist criterion. The same DSL modem used on a different telephone line might see the same DC notch and attenuation with increasing frequency, but perhaps also a notch caused by a "bridged tap" (open-circuit unterminated extension-phone line that branches from the main line), AM radio noise (RF noise), and possibly crosstalk noise (which increases with frequency) from other DSLs sharing the same telephone cable. The transmit basis function are the same, but clearly carry a very different allocation of energy and information for this second example. Again though, equalization (which would have been very complex in this second case) is avoided. This chapter will develop a theory to analyze and ultimately find optimum the strategy basically outlined in Figure 4.1 on any stationary linear channel with additive Gaussian noise.

The origins of multi-channel modulation date to antiquity. Shannon's 1948 information-theory paper constructs capacity bounds for transmission on a AWGN channel with linear ISI, effectively using multitone modulation. The first uses of multi-channel modulation appear to be in the 1950's with many claiming invention. The 1958 Collins Kineplex modem, the very first voiceband data modem, used a rudimentary form of multitone modulation. This modem operated at data rates of about 3000 bps, more than 10 times faster than other modems in existence for the subsequent decade – an early testimonial to the potential of multitone modulation. Holsinger's 1964 MIT dissertation describes experiments on another modem with optimized transmit distributions that approximated Shannon's famous "water-filling" best adaptive spectrum, which was also more completely developed within that same dissertation. Gallager of MIT later added detailed mathematical rigor and popularized the term "water-filling" for the shape of the optimized transmit spectrum as a function of the channel and noise characteristics. An early problem was the stable implementation of multiple modulators in hardware, stifling wide use of the method for nearly 20 years.

A number of increasingly successful attempts at the use of multichannel modulation occurred over a 30-year period, including attempts by Chang and Saltzberg of Bell Labs (1967), Weinstein and Eber of Bell Labs (1973), Peled and Ruiz of IBM (1980), Keasler of U of Illinois (1970's, licensed by Gandolf), Baran of Telebit (1990's), Mallory and Chaffee of IMC (1990's), and Hirosaki of NEC (1987). Some of the early implementers mentioned became frustrated in their analog-signal-processing implementations and never recovered to later benefit from nor understand the merits of the approach in digital-signal-processing implementation. While the various wired-line[1] multichannel modems did often work better than other single-channel modems, a variety of practical problems had earned "multitone" a reputation slandered with cynicism and disdain by the early 1990's.

A final thrust at popularizing multitone transmission, this time addressing many of the practical problems inherent in the technique with digitally implementable solutions under the name "Discrete Multitone" (DMT) emanated from Stanford University[2]. With anti-multitone hostilities at extreme and ugly levels, the "academic" Stanford group was grudgingly allowed to compete with traditional equalized systems designed by three independent sets of recognized experts from some of the world's most reknown telecommunications companies in what has come to be known as the "Bellcore Transmission Olympics" in 1993 for DSL. The remaining world's transmission experts anticipated the final downfall of the multitone technique. In dramatic reversal of fortune, with measured 10-30 dB improvements over the best of the equalized systems on actual phone lines, the gold-medalist DMT system ressurected multitone transmission at 4:10 pm on March 10, 1993[3], when a US (and now world-wide) standard was set for transmitting data over nearly a billion phone lines using multichannel modulation – this is DSL[4]. At time of writing, over 200 million DMT components have been sold for DSL. Such credibility

[1]Wireless uses of multitone transmission are somewhat different and more recent and do not adapt the transmitter to the channel, instead using various codes that are pertinent to channels that vary so quickly that the transmitter cannot be adapted as suggested here in Chapter 4. Wireless multitone systems will be mentioned more in Section 4.5.

[2]Special thanks to former Stanford students and Amati employees Jim Aslanis, John Bingham, Jacky Chow, Peter Chow, Toni Gooch, Ron Hunt, Kim Maxwell, Van Whitis and even an MIT, Mark Flowers, and a Utah, Mark Mallory.

[3]An interesting and unexpected coincidence is that this formal decision time and the very first demonstration of an operating telephone ("Watson, Come here I need you," said A.G. Bell) are coincident to the minute 117 years apart, with Bell's famous demonstration occuring on March 10, 1876, also at 4:10 pm.

[4]It should be noted that a second Bellcore Olympics was held in 2003 for higher speed DSLs as the remnants of the old

established, it is now possible to develop the descriptions in this Chapter to the potential successful use by the reader.

At about the same time that wireline uses were succeeding, wireless use of multiple carriers became popular for digital audio and digital video broadcast systems. Clearly, these wireless transmission systems had no feedback path for an adaptive transmitter, that would of course be common to all users anyway and could not be adapted. Even if feedback were possible,time-variation of wireless channels is sufficiently rapid to overwhelm or at least significantly complicated any adaptive transmitters. Nonetheless, the concept of no equalizers (which otherwise would have to adapt rapidly and usually cannot do so in many time-varying wireless situations) had appeal. Codes (see chapters 10 and 11 for full treatment of coding) could be used however with equal information on all subchannels/tones of a multichannel system. The codes span the subchannels, allowing with their redundancy the recovery of any subchannels with poor probability of bit error – such systems are known as coded-OFDM. With the right code, such systems again can significantly outperform equalized systems simply because the equalizers cannot adapt quickly enough. The very first successful test of a digital broadcast audio system in late 1993 was actually performed with a specially retrofitted DSL modem with equal information and energy on all channels. The performance gain over a single wide-band equalized system in the tests sponsored by the United States Electronics Industry Association (EIA) in Ohio again repeated the enormous gains of multi-carrier over the single-carrier system. Digital broadcast audio and video systems throughout the world subsequently proceeded to use various multi-carrier designs that parallel the early DSL-based prototype's designs. Additionally, a wireless local-area-network (WLAN) development called "Hyperlan" subsequently used some of the digital broadcast concepts to specify a wireless multicarrier at carrier frequencies of roughly 5 GHz (so above the digital audio and video bands). Again, there was good success and this system's 54 Mbps transmission came to be standardized by the IEEE as IEEE 802.11(a) and then at a lower 2.4 GHz carrier frequency as the popular IEEE 802.11(g), and by comparison a single-carrier "equalized" system known as IEEE 802.11(b) was limited to 11 Mbps (both use the same bandwidth). These wireless successes, directly traced back to the Olympic prototype above, also then enabled the non-adaptive (but coded) OFDM systems to flourish also, quieting the single-carrier equalization supporters perhaps permanently.

Section 4.1 illustrates the basic multitone concept generically and then specifically through the use of the same $1 + .9D^{-1}$ channel that has been used throughout this text. Multichannel modulation forms a set of parallel independent transmission channels that are used in aggregate to carry a data bit stream. The generic parallel-channel situation is investigated in Section 4.2. The investigations include a review of the "SNR gap" of earlier chapters as well as an elaboration of Chapter 6's water-filling concepts in a design (rather than capacity theory) context. The water-fill concept is easily described mathematically and pictorially, but can require large complexity in implementation. Loading algorithms that optimize the transmit bit and energy distribution for the parallel channels are introduced in Section 4.3. Channel partitioning, the construction of parallel independent channels from a channel with intersymbol interference, is studied in Section 4.4. With isolated transmit symbols, achieved by waiting for trailing intersymbol interference to decay to zero, the optimum "vector coding" and suboptimum DMT methods are described and analyzed. These methods are found to converge to the same performance as the number of channels increases. The identification of the channel is very important to vector coding and DMT methods and specific channel identification methods are studied in Section 4.7. A more complete discussion of channel identification occurs in Chapter 13. The isolation of transmit symbols can be undesirable and was one of the problems poorly addressed in early multichannel efforts (leading to some of the cynicism). An improvement, but not the last improvement in the author's opinion, called a "TEQ" or "time-domain equalizer" is discussed in Section 4.8. Some other methods for practical isolation are suggested in Section 4.9, while some straightforward decoding and code-concatenation implementation issues are addressed in Section 4.10.

single-carrier component suppliers attempted to overturn the earlier results. The results were again the same, dramatic huge gains measured by a neutral lab of the DMT technique, and thus high speed DSLs known as VDSL are again DMT-based

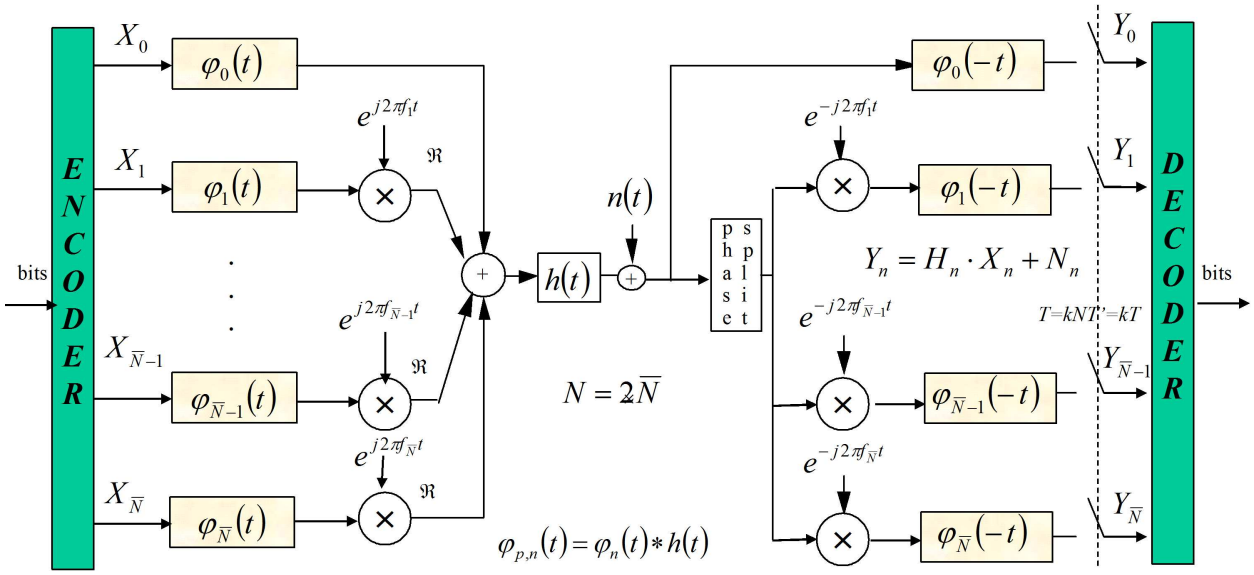


Figure 4.2: Basic multitone transmission.

4.1 Basic Multitone

Multitone transmission uses two or more coordinated passband (QAM or QAM-like) signals to carry a single bit stream over a communication channel. The passband signals are independently demodulated in the receiver and then re-multiplexed into the original bit stream. The heuristic motivation for multitone modulation is that if the bandwidth of each of the "tones" is sufficiently narrow, then no ISI occurs on any subchannel. The individual passband signals may carry data equally or unequally. Usually, the passband signal(s) with largest channel output signal-to-noise ratio carry a proportionately larger fraction of the digital information, consistent with Section 1.6's "gap approximation" that[5] $\bar{b}_n = \frac{1}{2}\log_2\left(1 + \frac{\mathrm{SNR}_n}{\Gamma}\right)$.

Figure 4.2 shows the simplest multitone system to understand. $\bar{N}$ QAM-like modulators, along with possibly one DC/baseband PAM modulator, transmit $\bar{N}+1$ sub-symbol components X_n, $n = 0, 1, ..., \bar{N}$.[6] ($\bar{N} \triangleq N/2$, and N is assumed to be even in this chapter.) This system is essentially the same as the general modulator of Chapter 1. X_0 and $X_{\bar{N}}$ are real one-dimensional **sub-symbols** while X_n, $n = 1, ..., \bar{N}-1$ can be two-dimensional ("complex") sub-symbols. Each sub-symbol represents one of 2^{b_n} messages that can be transmitted on subchannel n. The carrier frequencies for the corresponding subchannels are $f_n = n/T$, where T is again the symbol period. The baseband-equivalent basis functions are $\varphi_n(t) = \frac{1}{\sqrt{T}} \cdot \mathrm{sinc}\left(\frac{t}{T}\right)\ \forall\ n$.[7] The entire transmitted signal can be viewed as $\bar{N}+1$ independent transmission subchannels as indicated by the frequency bands of Figure 4.3.

The multitone modulated signal is transmitted over an ISI/AWGN channel with the corresponding demodulator structure also shown in Figure 4.2. Each subchannel is separately demodulated by first quadrature decoupling with a phase splitter and then baseband demodulating with a matched-

[5] Of importance here in the "gap approximation" is that the quantity Γ, that is the "gap," measures the proximity of data rates to a highest theoretically achievable data rate known as the channel capacity (see Chapter 8). When $\Gamma = 1$ (0 dB), the data rate is highest (and this is the lowest possible gap). Various coding methods in Chapters 10 and 11 have constant gaps across a wide range of possible numbers of bits per symbol. Thus the gap of 8.8 dB or 9.5 dB can be much smaller in complete designs using coding. Whatever the gap of a family of codes, if constant, it can be assumed then independent of b_n on any channel of a multi-channel design.

[6] Capital letters are used for symbol component elements rather than the usual lower case letters because these elements are frequency-domain elements, and frequency-domain quantities in this book are capitalized.

[7] The passband basis functions are actually the real and imaginary parts of $\sqrt{2/T}\mathrm{sinc}(\frac{t}{T}) \cdot e^{j2\pi f_n t}$ with receiver scaling tacitly moved to the transmitter as in Chapter 2.

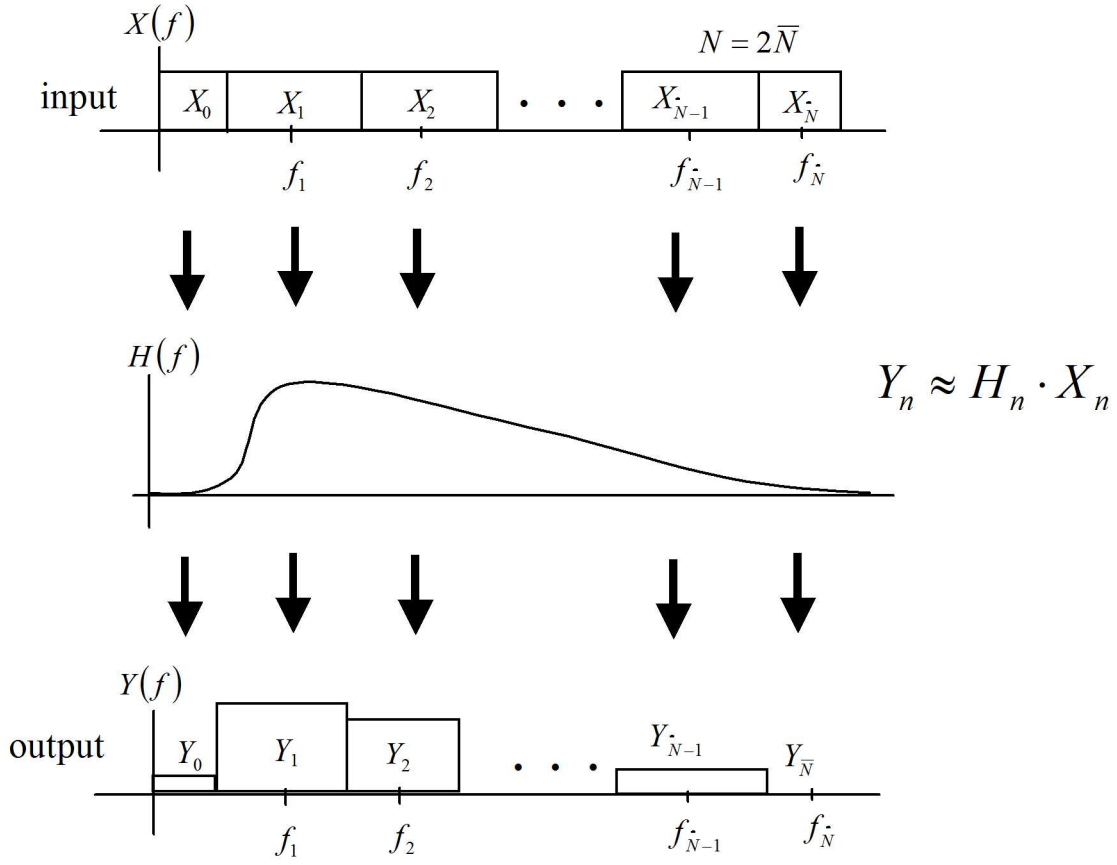


Figure 4.3: Illustration of frequency bands for multitone transmission system.

filter/sampler combination. With this particular ideal choice of basis functions, the channel output basis functions $\varphi_{p,n}(t)$ are an orthogonal set. There is thus no interference between the subchannels. Each subchannel may have ISI, but as $N \to \infty$, this ISI vanishes. Thus, (sub)symbol-by-(sub)symbol detection independently applied to each subchannel implements an overall ML detector. No equalizer is then necessary (large N) to implement the maximum-likelihood detector. Thus, maximum-likelihood detection is thus more easily achieved with multitone modulation on an ISI channel than it is on a single QAM or PAM signal. Again, equalization is unnecessary if the bandwidth of each tone is sufficiently narrow to make the ISI on that subchannel negligible. Thus a high-speed modulator and equalizer are replaced by $\bar{N}+1$ low-speed modulators and demodulators in parallel.

Multitone modulation typically uses a value for $\bar{N}$ that ensures that the pulse response of the ISI channel appears almost constant at $H(n/T) \triangleq H_n = H(f)$ for $|f - n/T| < 1/2T$. In practice, this means that the symbol period T greatly exceeds the length of the channel pulse response. The (scaled) matched filters then simply become the bandpass filters $\varphi_{p,n}(t) = \varphi_n(t) = 1/\sqrt{T} \cdot \text{sinc}(t/T) \cdot e^{-\jmath(2\pi/T)nt}$ and the sampled outputs become

$$Y_n \approx H_n \cdot X_n + N_n \quad . \tag{4.1}$$

The accuracy of this approximation becomes increasingly exact as $N \to \infty$. Figure 4.3 illustrates the scaling of H_n at the channel output on each of the subchannels. Each subchannel scales the input X_n by the pulse-response gain H_n. Each subchannel has an AWGN with power-spectral density σ_n^2. Thus, each subchannel has $\text{SNR}_n = \frac{\bar{\mathcal{E}}_n \cdot |H_n|^2}{\sigma_n^2}$.

Each subchannel in the multitone system carries b_n bits per symbol or $\bar{b}_n$ bits per dimension (note both $n = 0$ and $n = \bar{N}$ are by definition one-dimensional subchannels with subchannel $\bar{N}$ viewed as single-side-band modulation)[8]. The total number of bits carried by the multitone system is then

$$b = \sum_{n=0}^{\bar{N}} b_n \tag{4.2}$$

and the corresponding data rate is then

$$R = \frac{b}{T} = \sum_{n=0}^{\bar{N}} R_n \quad , \tag{4.3}$$

where $R_n \triangleq b_n/T$. Thus the aggregate data rate R is divided, possibly unequally, among the "subchannels."

With sufficiently large N, an optimum maximum-likelihood detector is easily implemented as $\bar{N}+1$ simple symbol-by-symbol detectors. This detector need not search all combinations of $M = 2^b$ possible transmit symbols. Each subchannel is optimally symbol-by-symbol detected. The reason this maximum-likelihood detector is so easily constructed is because of the choice of the basis functions: multitone basis functions are generically well-suited to transmission over ISI channels. Broader bandwidth basis functions, as might be found in a wider bandwidth single-carrier (for instance QAM) system, are simply not well suited to transmission on this channel, thus inevitably requiring suboptimum detectors for reasons of complexity. Chapter 5 investigates the concept of canonical transmission and finds a means for generalizing QAM transmission so that it is effectively just as good using a device known as the Generalized Decision Feedback Equalizer or GDFE, which can in some trivial cases look like the usual DFE of Chapter 3, but in most cases significantly differs in its generalized implementation, both at transmitter and receiver.

This chapter later trivially shows that multitone modulation, when combined with a code of gap Γ on each AWGN subchannel, comes as close to its theoretical capacity as that same code would come to capacity on an AWGN. That is, for sufficiently large N, multitone basis functions are indeed optimum for transmission on an ISI channel.

EXAMPLE 4.1.1 $(1 + .9D^{-1})$ This book has repeatedly used the example of a channel with (sampled) pulse response $1 + .9D^{-1}$ to compare performance of various ISI-mitigation

[8] Typically, subchannel 0 and subchannel $\bar{N}$ are not used in actual systems

$1 + .9D^{-1}$ Gains and SNR's

n	$\mathcal{E}_n$	H_n	SNR_n	b_n	arg Q-func (dB)
0	8/7	1.9	22.8 (13.6 dB)	1.6	9.2
1	16/7	$1 + .9e^{j\pi/4} = 1.76\angle 21.3^o$	19.5 (12.9 dB)	2×1.5	9.2
2	16/7	$1 + .9e^{j\pi/2} = 1.35\angle 42.0^o$	11.4 (10.6 dB)	2×1.2	9.1
3	16/7	$1 + .9e^{j3\pi/4} = .733\angle 60.25^o$	3.4 (5.3 dB)	$2 \times .5$	10

Table 4.1: Subchannel SNR's for the $1 + .9D$ channel with gap at 4.4 dB.

structures. Multitone modulation on this example, as well as on any linear ISI channel, achieves the highest performance. In order to have a continuous channel for multitone, we will write that $H(f) = 1+.9e^{j2\pi f}$ ($T = N$). This channel increasingly attenuates with higher frequencies, so multitone modulation will be used with subchannel $n = 4$ silenced ($X_4 = 0$). $N = 8$ is not sufficiently large for the approximation in (4.1) to hold, but larger N would make this example intractable for the reader. So, we assume $N = 8$ is sufficiently large for easy illustration of concept.

Table 4.1 summarizes the subchannel gains and SNR's for each of the subchannels in this example: For multitone, the phase choice of the channel is irrelevant to achievable performance, and the example would be the same for $1 + .9D^{-1}$ as for $1 + .9D$.

The SNR's in the above table are computed based on the assumption that $\frac{\mathcal{N}_0}{2} = .181$ – that is, $\text{SNR}_{\text{MFB}} = 10$ dB if binary modulation is used (as in previous studies with this example). Of course, SNR_{MFB} is a function of the modulated signals and in this example has meaning only in the context of performance comparisons against previous invocations of this example. The energy per subchannel has been allocated so that equal energy per dimension is used on the 7 available dimensions. With $T = N = 8$, this means that the transmit power of 1 for previous invocations of this example (that is one unit of energy per dimension and symbol period 1) is maintained here because the transmit power is $\sum_n \mathcal{E}_n/T = 8/8 = 1$. For this simple example, 4 symbol-by-symbol detectors operating in parallel exceeded the performance of a MMSE-DFE, coming very close to the MFB for binary PAM transmission. The data rate remains constant at $\bar{b} = 1$, and the power also is constant at $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$, so comparison against the DFE of Chapter 3 on this example is fair. The above table's Q-function arguments are found by using the QAM formula $d_{\min}/2\sigma = \sqrt{\frac{3}{M-1}\text{SNR}}$ for $n = 1, 2, 3$ and the PAM formula $d_{\min}/2\sigma = \sqrt{\frac{3}{M^2-1}\text{SNR}}$ with $M = 2^{b_n}$ – this approximation can be made very accurate by careful design of the signal sets for each subchannel.

For this example, $N = 8$ is not sufficiently high for all approximations to hold and a more detailed analysis of each subchannel's exact SNR might lead to slightly less optimistic performance projection. On the other hand, the energy per subchannel and the bit distribution in this example are not optimized, but rather simply guessed by the author. One finds for a large number of subchannels that performance will come down to 8.8 dB when the tones are sufficiently narrow for all approximations to hold accurately. An infinite-length MMSE-DFE, at best, achieves 8.4 dB in Chapter 3 – with Tomlinson precoding, the actual performance at the same data rate of $\bar{b} = 1$ was 1.3 dB lower or 7.1 dB. Thus a MMSE-DFE would perform about 1.7 dB worse than multitone on this channel. (The GDFE of Chapter 5 allows most of this 1.7 dB to be recovered, but also uses several subchannels.)

This example illustrates that there is something to be gained from multitone. An interesting aspect is also the parallelism inherent in the multitone receiver and transmitter, allowing for the potential of a very high-speed implementation on a number of parallel processing devices. The $1 + .9D^{-1}$ channel has mild ISI (but is easy to work with mathematically without computers) – the gain of 1.7 dB can be much larger on channels with severe ISI.

4.2 Parallel Channels

The multitone transmission system of Section 4.1 inherently exhibits $\bar{N}+1$ independent subchannels. Generally, multichannel systems can be construed as having N subchannels (where in the previous multitone system, several pairs of these subchannels have identical gains and can be considered as two-dimensional subchannels). This section studies the general case of a communication channel that has been decomposed into an equivalent set of N real parallel subchannels. Of particular importance is performance analysis and optimization of performance for the entire set of subchannels, which this section presents.

The concept of the SNR gap for a single channel is crucial to some of the analysis, so Subsection 4.2.1 reviews a single channel and then Subsections 4.2.2 and 4.2.3 use these single-channel results to establish results for parallel channels.

4.2.1 The single-channel gap analysis

The so-called "gap" analysis of Chapter 1 is reviewed here briefly in preparation for the study of parallel channels and loading algorithms.

On an AWGN channel, with gain P_n and noise power spectral density σ_n^2, the SNR_n is $\frac{\mathcal{E}_\cdot |P_n|^2}{\sigma_n^2}$. This channel has maximum data rate or capacity[9] of

$$\bar{c}_n = \frac{1}{2}\log_2(1+\text{SNR}_n) \tag{4.4}$$

bits/dimension. More generally,

$$\bar{b}_n = \frac{1}{2}\log_2(1+\frac{\text{SNR}_n}{\Gamma}) \quad . \tag{4.5}$$

Any reliable and implementable system must transmit at a data rate below capacity. The gap is a convenient mechanism for analyzing systems that transmit with $\bar{b} < \bar{c}$. Most practical signal constellations in use (PAM, QAM) have a constant gap for all $\bar{b} \geq .5$. For $\bar{b}_n < .5$, one can construct systems of codes to be used that exhibit the same constant gap as for $\bar{b} \geq .5$.[10] For any given coding scheme and a given target probability of symbol error P_e, an SNR gap is defined according to

$$\Gamma \triangleq \frac{2^{2\bar{c}}-1}{2^{2\bar{b}}-1} = \frac{\text{SNR}}{2^{2\bar{b}}-1} \quad . \tag{4.6}$$

To illustrate such a constant gap, Table 4.2.1 lists achievable $\bar{b}$ for uncoded QAM schemes using square constellations. For $\bar{b} < 1$, maintenance of this constant gap concept allows mild additional coding (see Chapters 10 and 11) in the "uncoded" system to obtain small additional coding gain when $b = .5$.

For uncoded QAM or PAM and $P_e = 10^{-6}$, the SNR gap Γ is constant at 8.8 dB. With uncoded QAM or PAM and $P_e = 10^{-7}$, the gap would be fixed instead at about 9.5 dB. The use of codes, say trellis or turbo coding and/or forward error correction (see Chapters 10 and 11) reduces the gap. A very well-coded system may may have a gap as low as .5 dB at $P_e \leq 10^{-6}$. Figure 4.4 plots $\bar{b}$ versus SNR for various gaps. Smaller gap indicates stronger coding. The curve for $\Gamma = 9$ dB approximates uncoded PAM or QAM transmission at probability of symbol error $P_e \approx 10^{-6}$. A gap of 0 dB means the theoretical maximum bit rate has been achieved, which requires infinite complexity and delay - see Chapter 8.) For the remainder of this chapter and the next, the same constant gap is assumed for the coding of each and every subchannel.

For a given coding scheme, practical transmission designs often mandate a specific value for b, or equivalently a fixed data rate. In this case, the design is not for $\bar{b}_{max} = \frac{1}{2}\log_2(1+\text{SNR}/\Gamma)$, but rather for b. The **margin** measures the excess SNR for that given bit rate.

[9] Capacity in Volume I is simply defined as the data rate with a code achieving a gap of $\Gamma = 1$ (0 dB). This is the lowest possible gap theoretically and can be approached by sophisticated codes, as in Volume II. Codes exist for which $\bar{P}_e \to 0$ if $\bar{b} \leq \bar{c}$.

[10] Some of the algorithms in Section 4.3 do not require the gap be constant on each tone, and instead require only tabular knowledge of the exact b_n versus required $\mathcal{E}_n$ relationship for each subchannel.

$\bar{b}$	.5	1	2	3	4	5
SNR for $P_e = 10^{-6}$ (dB)	8.8	13.5	20.5	26.8	32.9	38.9
$2^{2\bar{b}} - 1$ (dB)	0	4.7	11.7	18.0	24.1	30.1
Γ (dB)	8.8	8.8	8.8	8.8	8.8	8.8

Table 4.2: Table of SNR Gaps for $\bar{P}_e = 10^{-6}$.

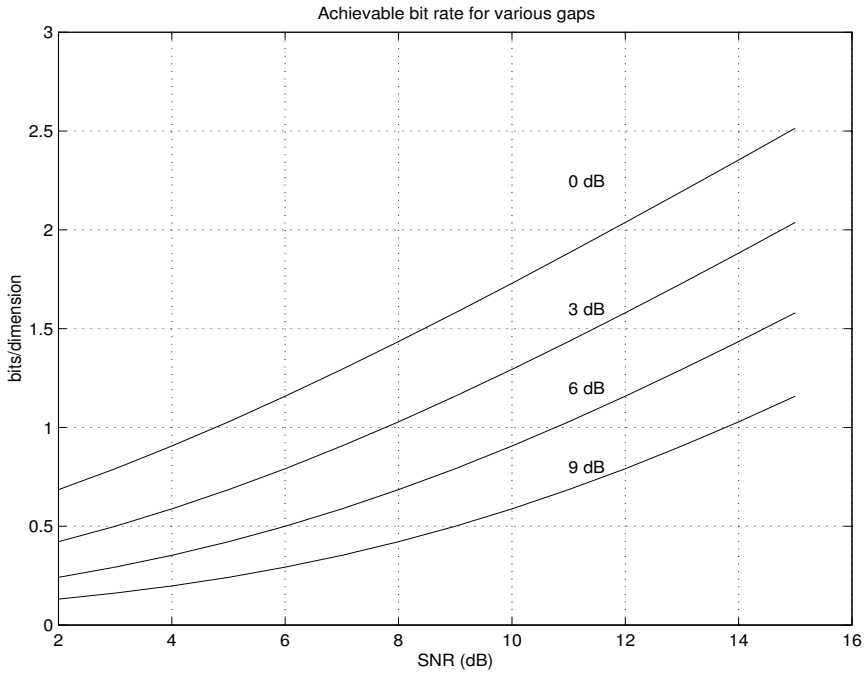


Figure 4.4: Illustration of bit rates versus SNR for various gaps.

(repeated from Section 1.5)

Definition 4.2.1 *The* **margin**, γ_m, *for transmission on an AWGN (sub)channel with a given SNR, a given number of bits per dimension* $\bar{b}$, *and a given coding-scheme/target-*$\bar{P}_e$ *with gap* Γ *is the amount by which the SNR can be reduced (increased for negative margin in dB) and still maintain a probability of error at or below the target* P_e.

Margin is accurately approximated through the use of the gap formula by

$$\gamma_m = \frac{2^{2\bar{b}_{max}} - 1}{2^{2\bar{b}} - 1} = \frac{\text{SNR}/\Gamma}{2^{2\bar{b}} - 1} \quad . \tag{4.7}$$

The margin is the amount by which the SNR on the channel may be lowered before performance degrades to a probability of error greater than the target P_e used in defining the gap. A negative margin in dB means that the SNR must improve by the magnitude of the margin before the P_e is achieved. The margin relation can also be written as

$$\bar{b} = .5 \log_2 \left(1 + \frac{\text{SNR}}{\Gamma \cdot \gamma_m} \right) \quad . \tag{4.8}$$

EXAMPLE 4.2.1 (AWGN with SNR = 20.5 dB) An AWGN has SNR of 20.5 dB. The capacity of this channel is then

$$\bar{c} = .5 \log_2 (1 + \text{SNR}) = 3.5 \text{ bits/dim} \quad . \tag{4.9}$$

With a $P_e = 10^{-6}$ and 2B1Q,

$$\bar{b} = .5 \log_2 \left(1 + \frac{\text{SNR}}{10^{.88}} \right) = 2 \text{ bits/dim} \quad . \tag{4.10}$$

With concatenated trellis and forward error correction (or with "turbo codes" – see Chapters 10 and 11), a coding gain of 7 dB at $P_e = 10^{-6}$, then the achievable data rate leads to

$$\bar{b} = .5 \log_2 \left(1 + \frac{\text{SNR}}{10^{.18}} \right) = 3 \text{ bits/dim} \quad . \tag{4.11}$$

Suppose a transmission application requires $\bar{b} = 2.5$ bits/dimension, then the margin for the coded system is

$$\gamma_m = \frac{2^{2 \cdot 3} - 1}{2^{2 \cdot 2.5} - 1} = \frac{63}{31} \approx 3 \text{ dB} \quad . \tag{4.12}$$

This means the noise power would have to be increased (or the transmit power reduced) by more than 3 dB before the target probability of error of 10^{-6} is no longer met. Alternatively, suppose a design transmits 4-QAM over this channel and no code is used, then the margin is

$$\gamma_m = \frac{2^{2 \cdot 2} - 1}{2^{2 \cdot 1} - 1} = \frac{15}{3} \approx 7 \text{ dB} \quad . \tag{4.13}$$

4.2.2 A single performance measure for parallel channels - geometric SNR

In a multi-channel transmission system, it is usually desirable to have all subchannels with the same P_e. Otherwise, if one subchannel had significantly higher P_e than others, then it would dominate bit error rate. Constant P_e can occur when all subchannels use the same class of codes with constant gap Γ. In this case, a single performance measure characterizes a multi-channel transmission system. This measure is a geometric SNR that can be compared to the detection SNR of equalized transmission systems or to the SNR_{MFB}.

For a set of N (one-dimensional real) parallel channels, the aggregate number of bits per dimension is

$$\begin{aligned} \bar{b} &= (1/N) \cdot \sum_{n=1}^{N} \bar{b}_n = (1/N) \sum_{n=1}^{N} \frac{1}{2} \log_2 \left(1 + \frac{\text{SNR}_n}{\Gamma}\right) && (4.14) \\ &= \frac{1}{2N} \log_2 \left(\prod_{n=1}^{N} \left[1 + \frac{\text{SNR}_n}{\Gamma}\right] \right) && (4.15) \\ &\triangleq \frac{1}{2} \log_2 \left(1 + \frac{\text{SNR}_{m,u}}{\Gamma}\right) \quad . && (4.16) \end{aligned}$$

Definition 4.2.2 (Multi-channel SNR) *The* **multi-channel SNR** *for a set of parallel channels is defined by*

$$SNR_{m,u} \triangleq \left[\left(\prod_{n=1}^{N} \left[1 + \frac{SNR_n}{\Gamma}\right] \right)^{1/N} - 1 \right] \cdot \Gamma \quad . \tag{4.17}$$

The multi-channel SNR is a single SNR measure that characterizes the set of subchannels by an equivalent single AWGN that achieves the same data rate, as in Figure 4.5. The multichannel SNR in (4.17) can be directly compared with the detection SNR of an equalized single channel system. The bit rate is given in (4.16) as if the aggregate set of parallel channels were a single AWGN with $\text{SNR}_{m,u}$.

When the terms involving +1 can be ignored, the multichannel SNR is approximately the geometric mean of the SNR's on each of the subchannels

$$\text{SNR}_{m,u} \approx \text{SNR}_{geo} = \left(\prod_n \text{SNR}_n \right)^{1/N} \quad . \tag{4.18}$$

Returning to Example 4.1.1, the multi-channel SNR (with gap 0 dB) is

$$\text{SNR}_{m,u} = \left[(22.8+1)(19.5+1)^2(11.4+1)^2(3.4+1)^2\right]^{.125} - 1 = 8.8(dB) \quad . \tag{4.19}$$

This multichannel SNR with $\Gamma = 0$ is very close to the argument that was used for all the Q-functions. This value is actually the best that can be achieved.[11] $\text{SNR}_{m,u} = 9.7$ dB for $\Gamma = 8.8$ dB, very close to the value in the sample design of Example 4.1.1. This value is artifically high only because the approximation accruing to using too small a value of N for illustration purposes - if N were large, $\text{SNR}_{m,u}$ would be close to 8.8 dB even with $\Gamma > 1$, and also the gap approximation loses accuracy when $\bar{b} < 1$.

4.2.3 The Water-Filling Optimization

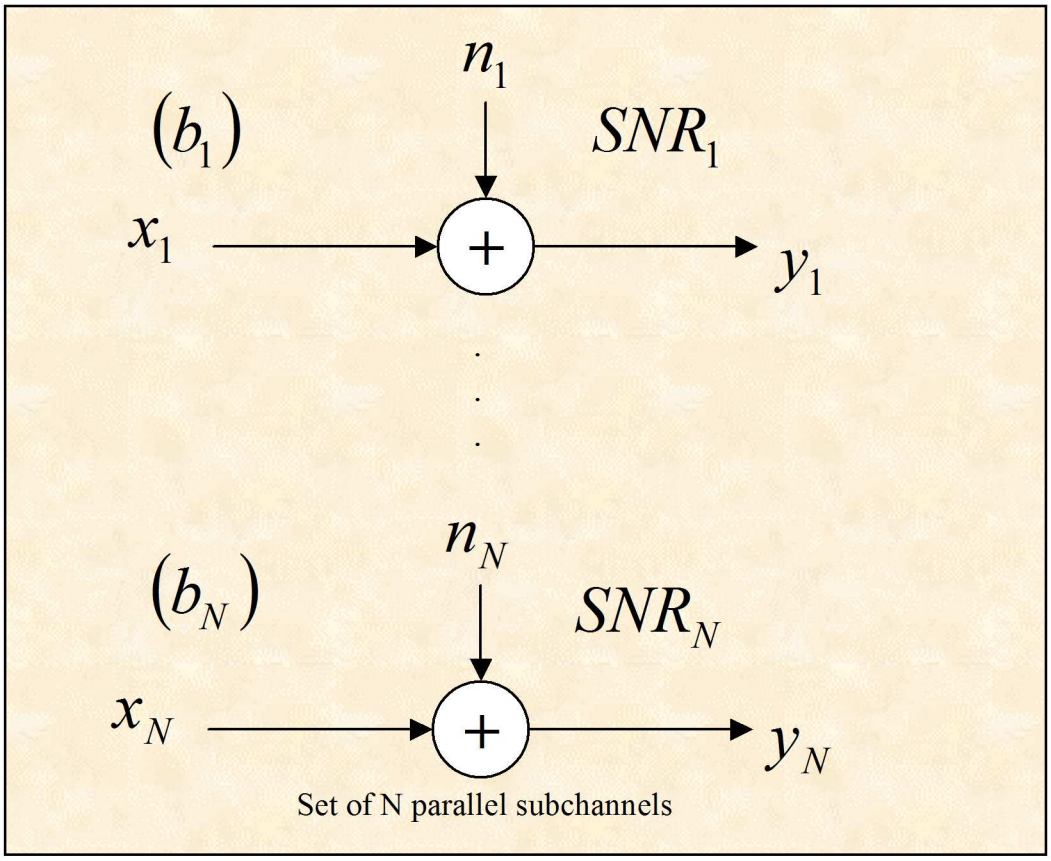
To maximize the data rate, $R = b/T$, for a set of parallel subchannels when the symbol rate $1/T$ is fixed requires maximization of the achievable $b = \sum_n b_n$ over b_n and $\mathcal{E}_n$. The largest number of bits that can be transmitted over a parallel set of subchannels must maximize the sum

$$b = \frac{1}{2} \sum_{n=1}^{N} \log_2(1 + \frac{\mathcal{E}_n \cdot g_n}{\Gamma}) \quad , \tag{4.20}$$

where g_n represents the subchannel signal-to-noise ratio when the transmitter applies unit energy to that subchannel. (For multitone, $g_n = |H_n|^2/(\sigma_n^2)$.) g_n is a fixed function of the channel, but $\mathcal{E}_n$ can be varied to maximize b, subject to an energy constraint that

$$\sum_{n=1}^{N} \mathcal{E}_n = N\bar{\mathcal{E}}_{\boldsymbol{x}} \quad . \tag{4.21}$$

[11] Somewhat by coincidence because the continuous optimum bandwidth from Chapter 8 is $.88\pi$. This example fortuitously can approximate $.88\pi$ very accurately only using 7/8 subchannels to get $.875\pi$. In other cases, a much larger number of subchannels may need to be used to approximate the optimum bandwidth.



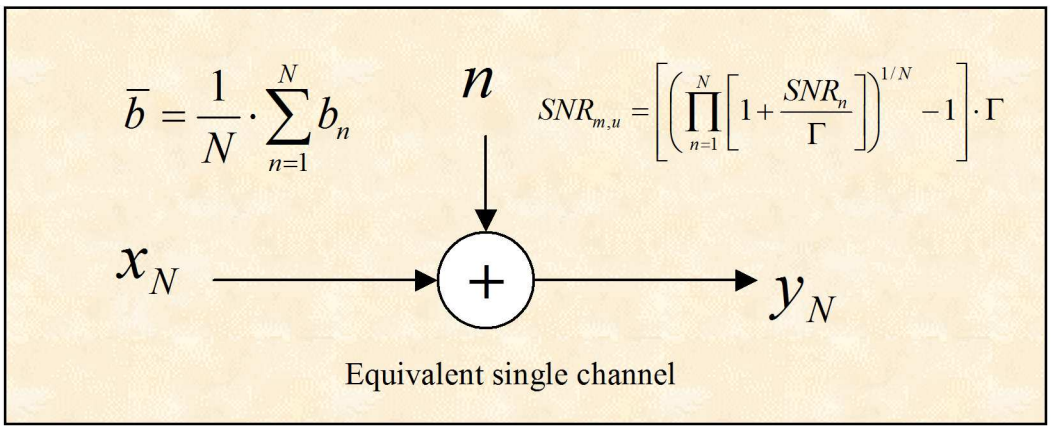


Figure 4.5: Equivalent single channel characterized by Multi-channel SNR.

Using Lagrange multipliers, the cost function to maximize (4.20) subject to the constraint in (4.21) becomes

$$\frac{1}{2\ln(2)}\sum_n \ln\left(1+\frac{\mathcal{E}_n\cdot g_n}{\Gamma}\right)+\lambda\left(\sum_n \mathcal{E}_n - N\bar{\mathcal{E}}_{\boldsymbol{x}}\right) \quad . \tag{4.22}$$

Differentiating with respect to $\mathcal{E}_n$ produces

$$\frac{1}{2\ln(2)}\frac{1}{\frac{\Gamma}{g_n}+\mathcal{E}_n} = -\lambda \quad . \tag{4.23}$$

Thus, (4.20) is maximized, subject to (4.21), when

$$\mathcal{E}_n + \frac{\Gamma}{g_n} = \text{ constant} \quad . \tag{4.24}$$

When $\Gamma = 1$ (0 dB), the maximum data rate or capacity of the parallel set of channels is achieved. The solution is called the "water-filling" solution because one can construe the solution graphically by thinking of the curve of inverted channel signal-to-noise ratios as being filled with energy (water) to a constant line as in Figure 4.6. (Figure 4.6 illustrates the discrete equivalent of water-filling for a set of 6 subchannels, which will be subsequently described further.) When $\Gamma \neq 1$, the form of the water-fill optimization remains the same (as long as Γ is constant over all subchannels). The scaling by Γ makes the inverse channel SNR curve, Γ/g_n vs n, appear more steep with n, thus leading to a more narrow (fewer used subchannels) optimized transmit band than when capacity ($\Gamma = 1$) is achieved. The number of bits on each subchannel is then

$$b_n = .5\cdot\log_2\left(1+\frac{\mathcal{E}_n\cdot g_n}{\Gamma}\right) \tag{4.25}$$

For the example of multi-tone, the optimum water-filling transmit energies then satisfy

$$\mathcal{E}_n + \Gamma\cdot\frac{\sigma_n^2}{|H_n|^2} = \text{ constant} \quad . \tag{4.26}$$

Figure 4.6 depicts water-filling for a transmission system with 6 subchannels with $g_n = \frac{|H_n|^2}{\sigma_n^2}$. Equation (4.26) is solved for the constant when $\sum_n \mathcal{E}_n = N\bar{\mathcal{E}}_{\boldsymbol{x}}$ and $\bar{\mathcal{E}}_{\boldsymbol{x}} \geq 0$. Four of the six subchannels in Figure 4.6 have positive energies, while 2 were eliminated for having negative energy, or equivalently having a noise power that exceeded the constant "water" line of water filling. The 4 used subchannels have energy that makes the sum of normalized noise and transmit energy constant. For methods of computing the water-fill solution, see the next section. The term "water-filling" arises from the analog of the curve of Γ/g_n being a bowl into which water (energy) is poured, filling the bowl until there is no more energy to use. The water will rise to a constant flat level in the bowl. The amount of water/energy in any subchannel is the depth of the water at the corresponding point in the bowl.

The water-filling solution is unique because the function being minimized is convex, so there is a unique optimum energy distribution (and corresponding set of subchannel data rates) for each ISI channel with multi-channel modulation.

4.2.4 Margin Maximization (or Power Minimization)

For many transmission systems, variable data rate is not desirable. In this case, the best design will maximize the performance margin at a given fixed data rate. To maximize fixed-rate margin, the designer equivalently minimizes the total energy (which is also called **fixed-margin** loading)

$$\sum_{n=1}^{N}\mathcal{E}_n \tag{4.27}$$

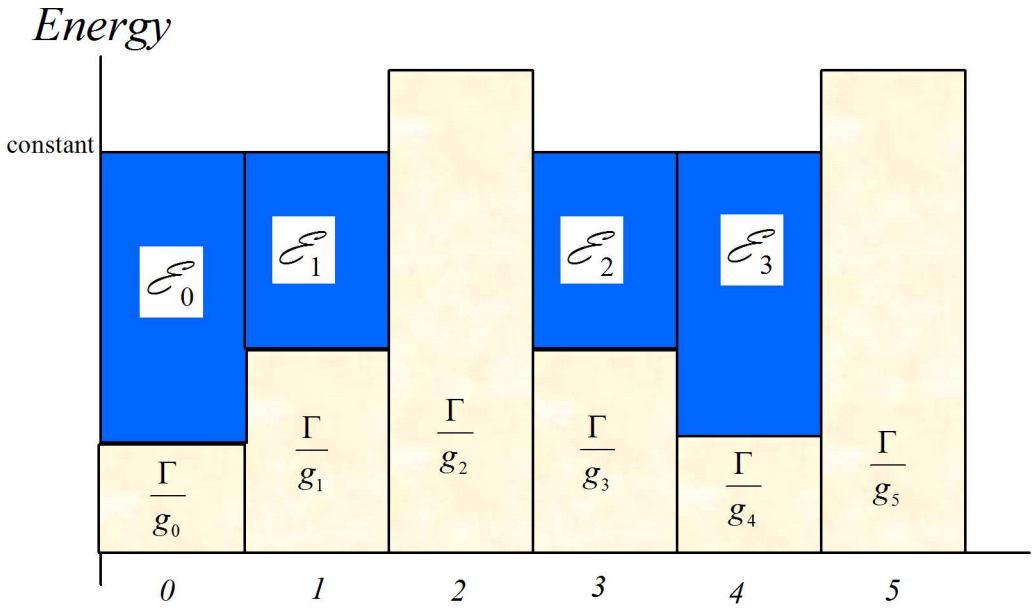


Figure 4.6: Illustration of discrete water-filling for 6 subchannels.

subject to the data rate and margin being fixed according to

$$\sum_{n=1}^{N} \frac{1}{2} \log_2 \left(1 + \frac{\mathcal{E}_n \cdot g_n}{\Gamma} \right) = b \quad . \tag{4.28}$$

The best margin will then be

$$\gamma_m = \frac{N \bar{\mathcal{E}}_{\boldsymbol{x}}}{\sum_{n=1}^{N} \mathcal{E}_n} \quad . \tag{4.29}$$

The solution to the energy minimization problem after correct differentiation is again the water-filling solution given by

$$\mathcal{E}_n + \frac{\Gamma}{g_n} = \text{ constant} \quad . \tag{4.30}$$

However, in this case water/energy is poured only until the number of bits per symbol (which is computed at each subchannel according to (4.25) and then summed over all subchannels) is equal to the given fixed rate in (4.28). Then the margin is computed according to (4.29). Each subchannel has its energy increased uniformly with respect to that which minimized the sum of energies by a factor of the maximum margin. Sometimes in multi-user channels, see Chapter 12, the solution with minimum energy for a given data rate and given margin is of interest in what is called "iterative water-filling." This energy-minimizing form of the loading problem is known in mathematics as "the dual form" of the rate-maximizing formulation.

Energy minimization is equivalent to margin maximization because any other bit distribution would have required more energy to reach the same given data rate. That greater energy would then clearly leave less energy to be distributed (uniformly) among the remaining subchannels if the new distribution used the same or a larger number of subchannels. If the new distribution used less subchannels than the water-fill, the amount by which the new energy distribution's total energy increased still exceeds the

improvement caused by extra energy being distributed over the smaller number of subchannels – for if this were not the case, then $\bar{b}$ could have been achieved more efficiently with the new distribution (a contradiction).

4.3 Loading Algorithms

Loading algorithms compute values for the bits b_n and energy $\mathcal{E}_n$ for each and every subchannel in a parallel set of subchannels. One example of a loading algorithm is the optimum water-filling algorithm of Section 4.2 that solves a set of linear equations with boundary constraints, as described further in this section. The solution of these water-filling equations for large N may produce b_n that have fractional parts or that are very small. Such small or fractional b_n can complicate encoder and decoder implementation. Alternative suboptimal loading algorithms approximate the water-fill solution, but constrain b_n to integer values, as also described later in this section.

There are two types of loading algorithms – those that try to maximize data rate and those that try to maximize performance at a given fixed data rate. Both are studied in this section.

Definition 4.3.1 (Rate-Adaptive (RA) loading criterion) *A* **rate-adaptive** *loading procedure maximizes (or approximately maximizes) the number of bits per symbol subject to a fixed energy constraint:*

$$\max_{\mathcal{E}_n} b = \sum_{n=1}^{N} \frac{1}{2} \log_2 \left(1 + \frac{\mathcal{E}_n \cdot g_n}{\Gamma}\right) \tag{4.31}$$

$$\textit{subject to: } N\bar{\mathcal{E}}_{\boldsymbol{x}} = \sum_{n=1}^{N} \mathcal{E}_n \tag{4.32}$$

Definition 4.3.2 (Margin-Adaptive (MA) loading criterion) *A* **margin-adaptive** *loading procedure minimizes (or approximately minimizes) the energy subject to a fixed bits/symbol constraint:*

$$\min_{\mathcal{E}_n} \mathcal{E}_{\boldsymbol{x}} = \sum_{n=1}^{N} \mathcal{E}_n \tag{4.33}$$

$$\textit{subject to: } b = \sum_{n=1}^{N} \frac{1}{2} \log_2 \left(1 + \frac{\mathcal{E}_n \cdot g_n}{\Gamma}\right) \tag{4.34}$$

The consequent maximum margin is then

$$\gamma_{max} = \frac{N\bar{\mathcal{E}}_{\boldsymbol{x}}}{\mathcal{E}} \quad . \tag{4.35}$$

When $\bar{\mathcal{E}}_{\boldsymbol{x}}$ is left at its minimized value (and the gap includes any margin and coding gain by definition) in the solution of Equations (4.31) and (4.32) and energy is then minimized for the given data rate b, the loading is called **Fixed Margin (FM)**.

FM loading is useful in transmission designs for which power savings is important.

4.3.1 Computing Water Filling for RA loading

The set of linear equations that has the water-fill distribution as its solution is

$$\mathcal{E}_1 + \Gamma/g_1 = K \tag{4.36}$$

$$\mathcal{E}_2 + \Gamma/g_2 = K \tag{4.37}$$

$$\vdots = \vdots \tag{4.38}$$

$$\mathcal{E}_N + \Gamma/g_N = K \tag{4.39}$$

$$\mathcal{E}_1 + ... + \mathcal{E}_N = N\bar{\mathcal{E}}_{\boldsymbol{x}} \quad . \tag{4.40}$$

There are a maximum of $N+1$ equations in $N+1$ unknowns The unknowns are the energies $\mathcal{E}_n, n = 1, ..., N$ and the constant K. The solution can produce negative energies. If it does, the equation with

the smallest g_n should be eliminated, and the corresponding $\mathcal{E}_n$ should be zeroed. The remaining sets of equations are solved recursively by eliminating the smallest g_n and zeroing $\mathcal{E}_n$, until the first solution with no negative energies occurs.

In matrix form, the equations become

$$\begin{bmatrix} 1 & 0 & 0 & 0 & ... & -1 \\ 0 & 1 & 0 & 0 & ... & -1 \\ \vdots & \vdots & \ddots & \ddots & \ddots & -1 \\ 1 & 1 & 1 & ... & 1 & 0 \end{bmatrix} \begin{bmatrix} \mathcal{E}_1 \\ \mathcal{E}_2 \\ \vdots \\ \mathcal{E}_N \\ K \end{bmatrix} = \begin{bmatrix} -\Gamma/g_1 \\ -\Gamma/g_2 \\ \vdots \\ -\Gamma/g_N \\ N\bar{\mathcal{E}}_{\boldsymbol{x}} \end{bmatrix} \quad , \tag{4.41}$$

which are readily solved by matrix inversion. Matrix inversions of size $N+1$ down to 1 may have to be performed iteratively until nonnegative energies are found for all subchannels.

An alternative solution sums the first N equations to obtain:

$$K = \frac{1}{N}\left[N\bar{\mathcal{E}}_{\boldsymbol{x}} + \Gamma \cdot \sum_{n=1}^{N} \frac{1}{g_n}\right] \quad , \tag{4.42}$$

and

$$\mathcal{E}_n = K - \Gamma/g_n \ \ \forall\, n = 1, ..., N \ . \tag{4.43}$$

If one or more of $\mathcal{E}_n < 0$, then the most negative is eliminated, and (4.42) and (4.43) are solved again with $N \to N-1$ and the corresponding g_n term eliminated. The equations preferably are preordered in terms of signal-to-noise ratio with $n = 1$ corresponding to the largest-SNR subchannel $g_1 = \max_n g_n$ and $n = N$ corresponding to the smallest-SNR subchannel $g_N = \min_n g_n$. Equation (4.42) then becomes at the i^{th} step of the iteration $(i = 0, ..., N)$

$$K = \frac{1}{N-i}\left[\mathcal{E}_{\boldsymbol{x}} + \Gamma \cdot \sum_{n=1}^{N-i} \frac{1}{g_n}\right] \quad , \tag{4.44}$$

which culminates with $N^* = N - i$ for the first value of i that does not cause a negative energy on $\mathcal{E}_i$, then leaving the energies as

$$\mathcal{E}_n = K - \Gamma/g_n \ \ \forall\, n = 1, ..., N^* = N - i \ . \tag{4.45}$$

The water-filling algorithm is illustrated by the flow chart of Figure 4.7.

The sum in the expression for K in (4.42) is always over the used subchannels. The following simplified formulae, apparently first observed by Aslanis, then determine the number of bits and number of bits per dimension as

$$b_n = \frac{1}{2}\log_2\left(\frac{K \cdot g_n}{\Gamma}\right) = \frac{1}{2}\log_2\left(1 + \frac{\mathcal{E}_n \cdot g_n}{\Gamma}\right) \quad . \tag{4.46}$$

and

$$\bar{b} = \frac{1}{2}\log_2\left(\frac{K^{N^*/N} \cdot (\prod_{n=1}^{N^*} g_n)^{1/N}}{\Gamma^{N^*/N}}\right) = \frac{1}{N}\sum_{n=1}^{N^*} b_n \tag{4.47}$$

$$= \frac{1}{2}\log_2\left(1 + \frac{\mathrm{SNR}_{m,u}}{\Gamma}\right) \quad , \tag{4.48}$$

respectively. The SNR for the set of parallel channels is therefore

$$\mathrm{SNR}_{m,u} = \Gamma\left\{\left(\frac{K}{\Gamma}\right)^{N^*/N}\left(\prod_{n=1}^{N^*} g_n^{1/N}\right) - 1\right\} \quad . \tag{4.49}$$

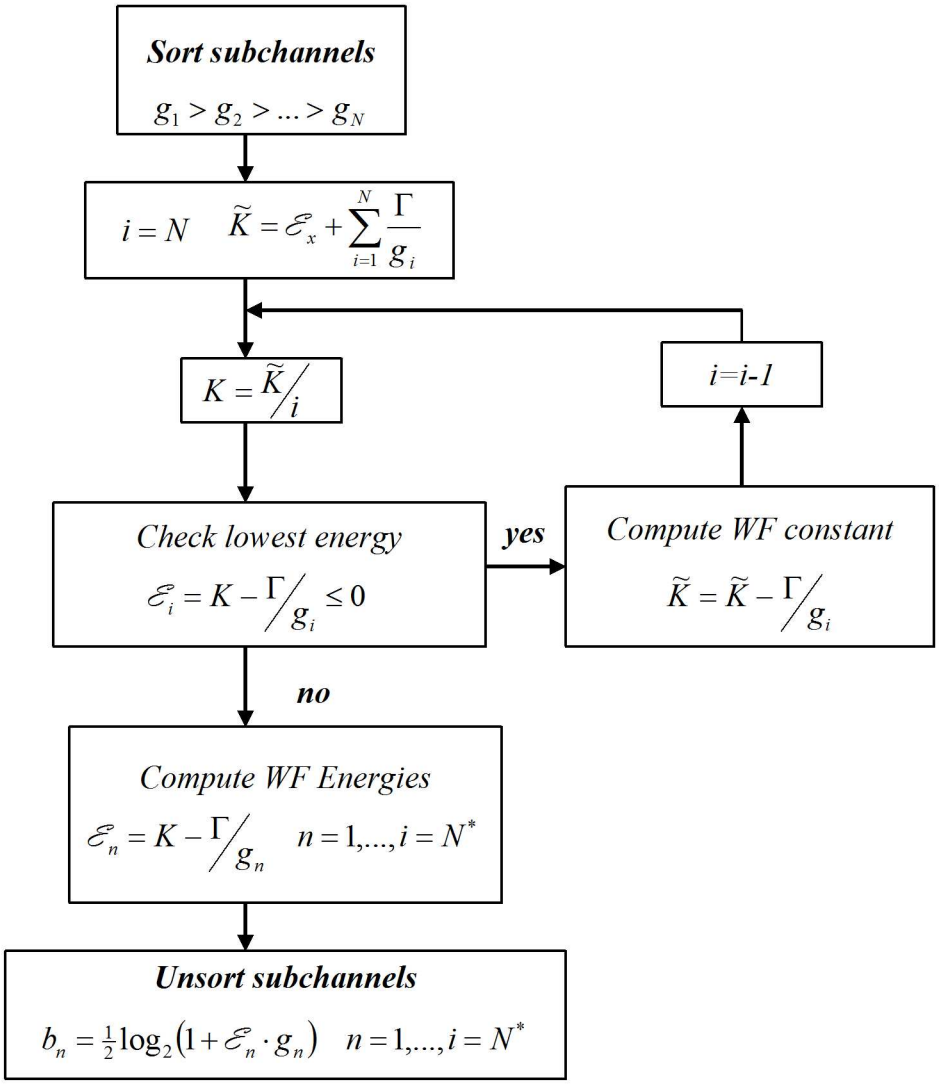


Figure 4.7: Flow chart of Rate-Adaptive water-filling algorithm.

The water-filling constant K becomes the ratio

$$K = \Gamma \cdot \left(\frac{2^{2b}}{\prod_{n=1}^{N^*} g_n} \right)^{1/N^*} \quad . \tag{4.50}$$

EXAMPLE 4.3.1 ($1 + .9D^{-1}$ again) The water-fill solution (for $\Gamma = 1$) can be found on the $1 + .9D^{-1}$ example to compute capacity. First, the subchannels are characterized by

$$g_0 = \frac{1.9^2}{.181} = 19.94 \tag{4.51}$$

$$g_1 = \frac{1.7558^2}{.181} = 17.03 \tag{4.52}$$

$$g_2 = \frac{1.3454^2}{.181} = 10.00 \tag{4.53}$$

$$g_3 = \frac{.7329^2}{.181} = 2.968 \tag{4.54}$$

$$g_4 = \frac{.1^2}{.181} = .0552 \tag{4.55}$$

Using (4.42) with all subchannels ($N = 8$ or actually 5 subchannels since 3 are complex with identical g's on real and imaginary dimensions), one computes

$$K = \frac{1}{8}\left[8 + 1 \cdot (\frac{1}{19.94} + \frac{2}{17.03} + \frac{2}{10.0} + \frac{2}{2.968} + \frac{1}{.0552})\right] = 3.3947 \quad , \tag{4.56}$$

and thus $\mathcal{E}_4 = 3.3947 - 1/.05525 = -14.7 < 0$. So the last subchannel should be eliminated and the equations propagated again. For $N = 7$,

$$K = \frac{1}{7}\left[8 + 1 \cdot (\frac{1}{20} + \frac{2}{17} + \frac{2}{9.8} + \frac{2}{3})\right] = 1.292 \quad , \tag{4.57}$$

The corresponding subchannel energies, $\bar{\mathcal{E}}_n$, on the subchannels are 1.24, 1.23, 1.19, and .96, which are all positive, meaning the solution with positive energy on all subchannels has been found. The corresponding number of bits per each subchannel is computed as $\bar{b}_n = .5 \log_2(K g_n)$, which has values $\bar{b}$=2.3, 2.2, 1.83, and .98 respectively. The sum is very close to 1.54 bits/dimension, almost exactly the known capacity of this channel[12]. Figure 4.8 illustrates the water-fill energy distribution for this channel.

The capacity (when $\Gamma = 1$ or 0 dB) exceeds 1 bit/dimension, implying that codes exist that allow arbitrarily small probability of error for the selected transmission rate in earlier uses of this example of 1 bit/dimension.

The accuracy of this computation of capacity can be increased somewhat by further increasing N – the eventual large-N-asymptotic value is approximately 1.55 bits/dimension. Then, $\text{SNR}_{m,u} = 2^{2(1.54)} - 1 = 8.8$ dB. This is the highest detection-point SNR that any receiver can achieve. This is better than the MMSE-DFE by .4 dB (or by 1.7 dB with error propagation/precoding effects included). On more severe-ISI channels ($1 + .9D^{-1}$ is easy for computation by hand but is not really a very difficult transmission channel), the difference between multitone and the MMSE-DFE can be very large. Chapter 5 shows how to correct the MMSE-DFE's transmit filter and symbol rate in some situations to improve the MMSE-DFE's detection SNR to the level of multitone, but this is not always possible.

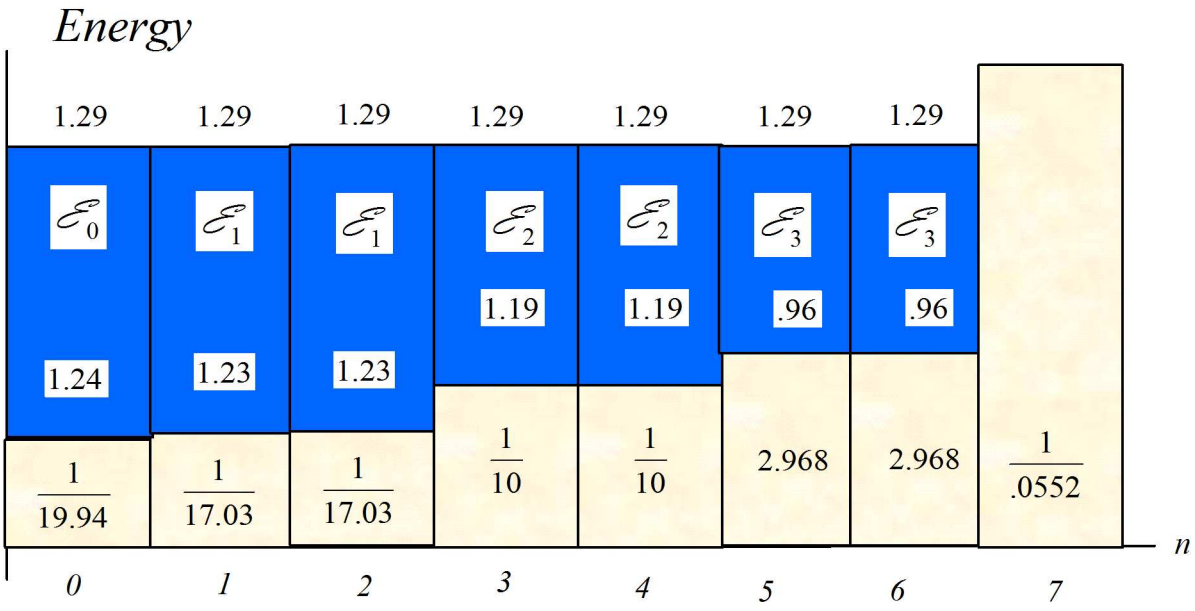


Figure 4.8: Water-filling energy distribution for $1 + .9D^{-1}$ channel with $N = 8$.

4.3.2 Computing Water-Filling for MA loading

MA water-filling will have a constant K_{ma} that satisfies

$$\mathcal{E}_n = K_{ma} - \frac{\Gamma}{g_n} \tag{4.58}$$

for each used subchannel. The bit-rate constraint then becomes

$$b = \frac{1}{2}\sum_{n=1}^{N^*} \log_2\left(1 + \frac{\mathcal{E}_n \cdot g_n}{\Gamma}\right) \tag{4.59}$$

$$= \frac{1}{2}\sum_{n=1}^{N^*} \log_2\left(\frac{K_{ma} \cdot g_n}{\Gamma}\right) \tag{4.60}$$

$$= \frac{1}{2}\log_2\left(\prod_{n=1}^{N^*} \frac{K_{ma} \cdot g_n}{\Gamma}\right) \tag{4.61}$$

$$2^{2b} = \left(\frac{K_{ma}}{\Gamma}\right)^{N^*} \cdot \prod_{n=1}^{N^*} g_n \tag{4.62}$$

Thus, the constant for MA loading is

$$K_{ma} = \Gamma\left(\frac{2^{2b}}{\prod_{n=1}^{N^*} g_n}\right)^{1/N^*} . \tag{4.63}$$

MA water-fill algorithm

1. Order g_n largest to smallest, $n = 1,..., N$, and set $i = N$.

[12] This channel is later shown to have a capacity of 1.55 bits/dimension, so not yet "known" to the reader upon initial reading.

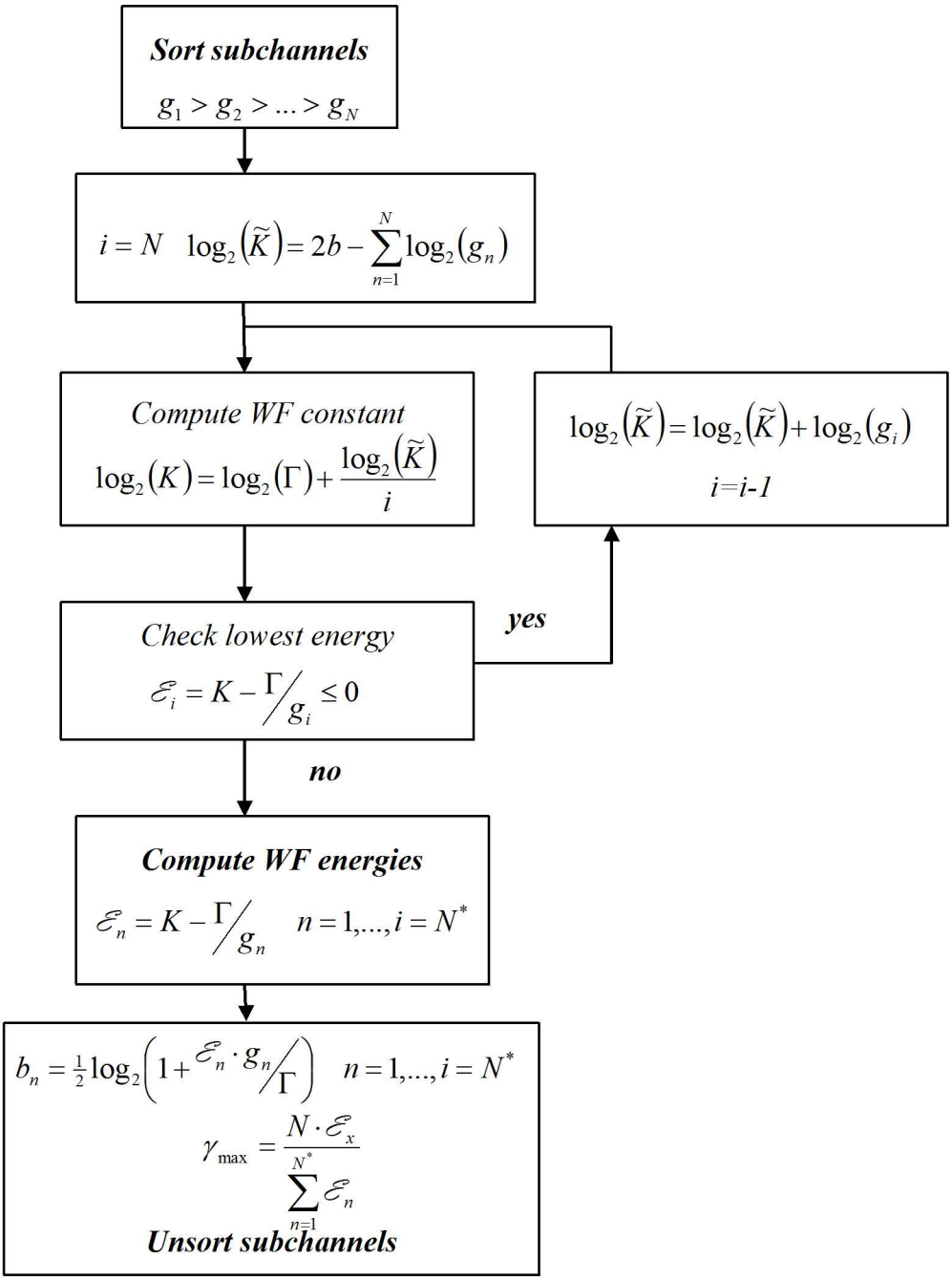


Figure 4.9: Flow chart of margin-adaptive water filling.

2. Compute $K_{ma} = \Gamma \left(\frac{2^{2b}}{\prod_{n=1}^{i} g_n} \right)^{1/i}$ (this step requires only one multiply for $i \leq N-1$).

3. Test subchannel energy $\mathcal{E}_i = K_{ma} - \frac{\Gamma}{g_i} < 0$? If yes, then $i \leftarrow i-1$, go to step 2, otherwise continue

4. Compute solution with $n = 1, ..., N^* = i$

$$\mathcal{E}_n = K_{ma} - \frac{\Gamma}{g_n} \tag{4.64}$$

$$b_n = \frac{1}{2} \log_2 \left(\frac{K_{ma} \cdot g_n}{\Gamma} \right) \tag{4.65}$$

5. Compute margin $\gamma_{max} = \frac{N\bar{\mathcal{E}}_{\boldsymbol{x}}}{\sum_{n=1}^{N^*} \mathcal{E}_n}$.

Figure 4.9 illustrates margin-adaptive water-filling. Figure 4.9 recognizes that the numerator and denominator terms in Equation (4.63) could be quite large while the quotient remains reasonable, so logarithms are used to reduce the precisional requirements inside the loop.

EXAMPLE 4.3.2 Returning to the $1 + .9D^{-1}$ example, the MA water-fill iterations for a gap of $\Gamma = 8.8$ dB for a $P_e = 10^{-6}$ are:

1. iteration $i = 5$ subchannels

$$K_{ma} = 10^{.88} \cdot \left(\frac{2^{16}}{19.945 \cdot 17.032^2 \cdot 10^2 \cdot 2.968^2 \cdot .05525} \right)^{.125} = 6.3228 \tag{4.66}$$

$$\mathcal{E}_4 = 6.3228 - 10^{.88}/.0552 = -131 < 0 \quad i \leftarrow i \rightarrow 4 \tag{4.67}$$

2. step $i = 4$ subchannels

$$K_{ma} = 10^{.88} \cdot \left(\frac{2^{16}}{19.845 \cdot 17.032^2 \cdot 10^2 \cdot 2.968^2} \right)^{1/7} = 4.0727 \tag{4.68}$$

$$\bar{\mathcal{E}}_3 = 4.0727 - 10^{.88}/2.968 = 1.5169 > 0 \;\; \mathcal{E}_3 = 3.0337 \tag{4.69}$$

$\mathcal{E}_2 = 6.6283$, $\mathcal{E}_1 = 7.2547$, $\mathcal{E}_0 = 3.6827$.

$$\gamma_{max} = \frac{8}{3.0337 + 6.6283 + 7.2547 + 3.6827} = (1/2.57) = -4.1 \text{ dB} \tag{4.70}$$

The negative margin indicates what we already knew, this channel needs coding or more energy (at least 4.1 dB more) to achieve a $P_e = 10^{-6}$, even with multitone. However, gap/capacity arguments assume that a code with nearly 9 dB exists, so it is possible with codes of gain greater than 4.1 dB, or equivalently gaps less than 4.7 dB, to have acceptable performance on this channel. The amount of additional coding necessary is a minimum with multi-tone (when $N \rightarrow \infty$).

4.3.3 Loading with Discrete Information Units

Water-filling algorithms have bit distributions where b_n can be any real number. Realization of bit distributions with non-integer values can be difficult. Alternative loading algorithms allow the computation of bit distributions that are more amenable to implementation with a finite granularity:

Definition 4.3.3 (Information Granularity) *The* **granularity** *of a multichannel transmission system is the smallest incremental unit of information that can be transmitted, β. The number of bits on any subchannel is then given by*

$$b_n = B_n \cdot \beta \quad , \tag{4.71}$$

where $B_n \geq 0$ is an integer.

Typically, β, takes values such as .25, .5, .75, 1, or 2 bit(s) with fractional bit constellations being discussed in both Chapters 10 and 11.

There are two approaches to finite granularity. The first computes a bit distribution by rounding of approximate water-fill results, and is called **Chow's algorithm**. The second is based on greedy methods in mathematics and first suggested by Hughes-Hartog; however, considerable significant improvements, a complete and mathematically verifiable framework, as well as an approach that circumvents many drawbacks in the original Hughes-Hartog methods was developed independently by Jorge Campello de Souza[13] and Howard Levin and are now known as **Levin-Campello (LC) Algorithms**. These algorithms solve the finite-β loading problem directly, which is not a water-filling solution. This section first describes Chow's algorithm, and then proceeds to the LC methods.

Chow's Algorithm

Peter Chow studied a number of transmission media with severe intersymbol interference, in particular telephone line digital transmission known generally as Digital Subscriber Line (DSL), in his 1993 Stanford Dissertation. Chow was able to verify that an "on/off" energy distribution, *as long as it used the same or nearly the same transmission band as water-filling*, exhibits negligible loss with respect to the exact water-filling shape. The reader is cautioned not to misinterpret this result as saying that flat energy distribution is as good as water-filling, but rather "on/off"; where "on" means flat energy distribution in the same used bands as water filling and "off" means zero energy distribution where water-filling would be zero.

Chow's algorithms take as much or more computation than water-fill methods, rasing the issue of their utility even if close to optimum. The reason for the use of "on/off" energy distributions is that practical systems usually have a constraint on the maximum power spectral density along with a total energy/power constraint. Such power-spectral-density constraints are typically flat over the region of bandwidth considered for transmission. Such "on/off' energy distribution comforts those concerned with the effect of emissions from a transmission channel on other transmission systems or electronic devices.

Using this observation, Chow was able to devise algorithms that approximate water-filling with finite granularity in the bit loading b_n and equal energy $\mathcal{E}_n = \frac{N}{N^*}\bar{\mathcal{E}}_{\boldsymbol{x}}$. Both the RA and MA problems can be solved with Chow's algorithms, which in both cases begin with **Chow's Primer** below:

Chow's "on/off" Loading Primer:

1. Re-order the g_n so that g_1 is the largest
2. Set $b_{temp}(N+1) = 0$, $i = N$. ($b_{temp}(i)$ and i are the tentative total bits and number of used subchannels, respectively.)
3. Set equal energy on used tones as $\mathcal{E}_n = N\bar{\mathcal{E}}_{\boldsymbol{x}}/i$ for $n = 1, ..., i$, and then $\text{SNR}_n = g_n\mathcal{E}_n$. (When power-spectral density limits would be exceeded - the energy level is instead set at the power-spectal-density limits and the number of used subchannels i is reset to $i+1$, making one last pass through steps 4 and 5).
4. Compute $b_{temp}(i) = \sum_{n=1}^{i} \frac{1}{2}\log_2(1 + \text{SNR}_n/\Gamma)$.
5. If $b_{temp}(i) < b_{temp}(i+1)$, then keep the bit distribution corresponding to $b_{temp}(i+1)$ - otherwise $i \leftarrow i-1$ and go to step 3.

The bit distribution produced by Chow's Primer will likely contain many subchannels with non-integer numbers of bits.

EXAMPLE 4.3.3 ($1 + .9D^{-1}$) The algorithm above can be executed for the $(1 + .9D^{-1})$ example with $\Gamma = 0$ dB. The 3 passes necessary for solution are:

[13] 379C students take heart, it was his course project!

Pass i=4 $\bar{\mathcal{E}}_n = 1,\ n = 0, 1, 2, 3, 4.$

$$\begin{aligned} \mathrm{SNR}_0 &= \frac{1.9^2}{.181} = 19.9448 && (4.72)\\ \mathrm{SNR}_1 &= \frac{1.7558^2}{.181} = 17.032 && (4.73)\\ \mathrm{SNR}_2 &= \frac{1.3454^2}{.181} = 10.000 && (4.74)\\ \mathrm{SNR}_3 &= \frac{.7392^2}{.181} = 2.968 && (4.75)\\ \mathrm{SNR}_4 &= \frac{.1^2}{.181} = .0552 && (4.76) \end{aligned}$$

$$b_{temp}(4) = .5{\cdot}\log_2(1{+}19.9448){+}\log_2(1{+}17.032){+}\log_2(1{+}10){+}\log_2(1{+}2.968){+}.5{*}\log_2(1.0552) = 11.8533\ . \quad (4.77)$$

Pass i=3 $\bar{\mathcal{E}}_n = 8/7,\ n = 0, 1, 2, 3.$

$$\begin{aligned} \mathrm{SNR}_0 &= \frac{8 \cdot 1.9^2}{7 \cdot .181} = 22.794 && (4.78)\\ \mathrm{SNR}_1 &= \frac{8 \cdot 1.7558^2}{7 \cdot .181} = 19.4651 && (4.79)\\ \mathrm{SNR}_2 &= \frac{8 \cdot 1.3454^2}{7 \cdot .181} = 11.4286 && (4.80)\\ \mathrm{SNR}_3 &= \frac{8 \cdot .7392^2}{7 \cdot .181} = 3.3920 && (4.81) \end{aligned}$$

$$b_{temp}(3) = .5 \cdot \log_2(23.794) + \log_2(20.4651) + \log_2(12.42865) + \log_2(4.3920) = 12.4118\ . \quad (4.82)$$

Pass i=2 $\bar{\mathcal{E}}_n = 8/5,\ n = 0, 1, 2.$

$$\begin{aligned} \mathrm{SNR}_0 &= \frac{8 \cdot 1.9^2}{5 \cdot .181} = 31.9116 && (4.83)\\ \mathrm{SNR}_1 &= \frac{8 \cdot 1.7558^2}{5 \cdot .181} = 27.2512 && (4.84)\\ \mathrm{SNR}_2 &= \frac{8 \cdot 1.3454^2}{5 \cdot .181} = 16.000 && (4.85) \end{aligned}$$

$$b_{temp}(2) = .5 \cdot \log_2(32.9116) + \log_2(28.2512) + \log_2(17) = 11.428 < 12.4118\ . \quad (4.86)$$

STOP and use $i = 3$ result.

The algorithm has terminated with the computation of $b = 12.4118$ or $\bar{b} = 1.54$ bits/dimension, the known capacity of this channel for $\Gamma = 0$ to almost 3 digit accuracy, confirming on this example that on/off energy is essentially as good as water-filling energy shaping. The key in both Chow and Water-filling methods is that the tone at Nyquist is not used - it's use corresponds to about a 10% loss in data rate on this example.

To illustrate that one could reorder the tones from smallest to largest in terms of channel SNR, the 5 passes necessary for solution follow:

Pass 1 $\mathrm{SNR} = 160$
$b_{temp}(1) = .5 \log_2(1 + 160/1) = 3.7$
$\bar{b}_{temp} = b_{temp}/8 = .46$

Pass 2 $\mathrm{SNR}_0 = 53.2$
$\mathrm{SNR}_1 = 45.5$
$b_{temp}(2) = 2.9 + 2(2.75) = 8.4$
$\bar{b}_{temp} = b_{temp}/8 = 1.1$

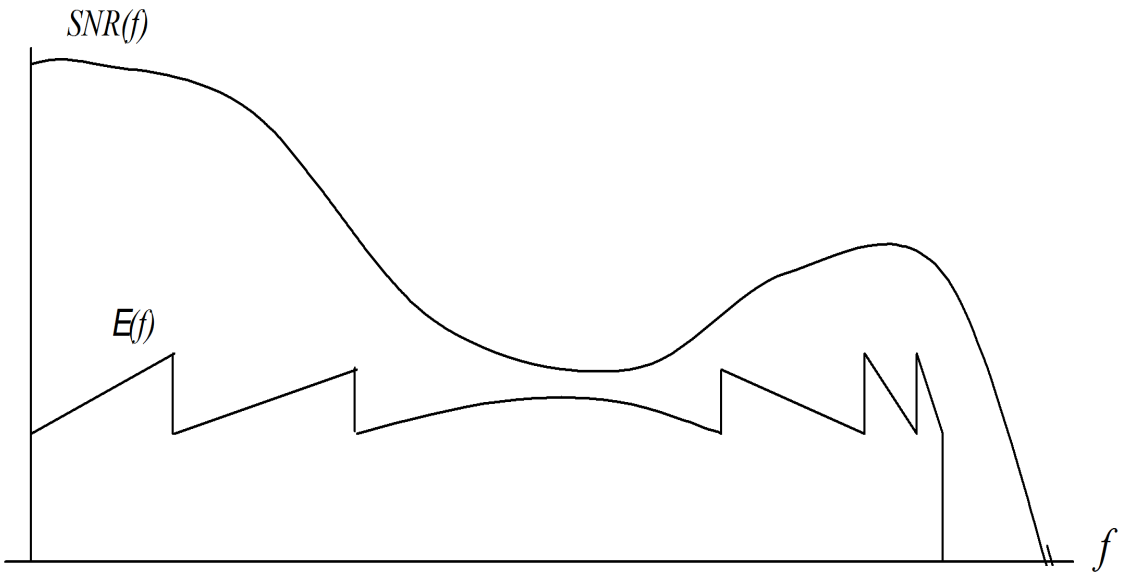


Figure 4.10: Illustration of sawtooth energy distribution characteristic of Chow's Algorithm.

Pass 3 $\text{SNR}_0 = 31.9$
$\text{SNR}_1 = 27.3$
$\text{SNR}_2 = 16.0$
$b_{temp}(3) = 2.5 + 2(2.4) + 2(2.0) = 11.3$
$\bar{b}_{temp} = b_{temp}/8 = 1.42$

Pass 4 $\text{SNR}_0 = 22.8$
$\text{SNR}_1 = 19.5$
$\text{SNR}_2 = 11.4$
$\text{SNR}_3 = 3.4$
$b_{temp}(4) = 2.3 + 2(2.1) + 2(1.8) + 2(1.05) = 12.4$
$\bar{b} = b_{temp}/8 = 1.55$

Pass 5 $\text{SNR}_0 = 20.0$
$\text{SNR}_1 = 17.1$
$\text{SNR}_2 = 10.0$
$\text{SNR}_3 = 3.0$
$\text{SNR}_4 = .055$
$B = 2.2 + 2(2.1) + 2(1.7) + 2(1) = 11.8$
$\bar{b}_{temp} = b_{temp}/8 = 1.48$

Again, the correct solution was obtained with a larger number of steps. Designers sometimes run the algorithm backwards if they expect less than half the subchannels to be used. Then, computation is saved.

Chow's RA Algorithm To solve the RA problem, the following 3 steps adjust the result of the Chow's Primer (with steps 6 and 7 being executed for all used subchannels):

6. If $\frac{b_n}{\beta} - \lfloor \frac{b_n}{\beta} \rfloor < .5\beta$, then round b_n to the largest integer multiple of β contained in b_n, $B_n = \lfloor b_n/\beta \rfloor$. Reduce energy on this subchannel so that the performance is the same as on all other subchannels – that is scale energy by $\frac{2^{2\beta\bar{B}_n}-1}{2^{2\bar{b}_{old,n}}-1}$ so that the new integer value then satisfies $b_n = \frac{1}{2}\log_2(1 + \mathrm{SNR}_n/\Gamma)$.
7. If $\frac{b_n}{\beta} - \lfloor \frac{b_n}{\beta} \rfloor > .5\beta$, then round to the next largest integer, $B_n = \lceil b_n/\beta \rceil$. Increase energy, $\mathcal{E}_n$, by the factor $\frac{2^{2\beta\bar{B}_n}-1}{2^{2\bar{b}_{old,n}}-1}$ so that performance is the same as other subchannels and then $b_n = \frac{1}{2}\log_2(1 + \mathrm{SNR}_n/\Gamma)$.
8. Compute total energy $\mathcal{E} = \sum_n \mathcal{E}_n$ and then scale all subchannels by the factor $N\bar{\mathcal{E}}_{\boldsymbol{x}}/\mathcal{E}$.

This procedure produces a characteristic "sawtooth" energy distribution that may deviate from a flat distribution by as much as $\pm\frac{6\cdot\beta}{\tilde{N}}$ dB, where $\tilde{N}$ is the dimensionality of the subchannel. Figure 4.10 illustrates the sawtooth energy produced by Chow's algorithm for the $SNR(f)$ curve simultaneously displayed. The discontinuities are points at which the number of bits on a subchannel changes by the smallest information unit β. An increase in b_n by β means more energy is necessary, while a decrease in b_n by β means less energy is necessary. The curve however, on rough average, approximates a flat straight line where energy is "on."

EXAMPLE 4.3.4 Returning to the earlier $1+.9D^{-1}$ example, rounding the bit distribution with $\beta = 1$ bit leads to

$$2 \leftarrow 2.28 = b_0 = .5 \cdot \log_2(23.794) \tag{4.87}$$
$$4 \leftarrow 4.355 = b_1 = \log_2(20.4651) \tag{4.88}$$
$$4 \leftarrow 3.6356 = b_2 = \log_2(12.4286) \tag{4.89}$$
$$2 \leftarrow 2.13 = b_3 = \log_2(4.3920) \tag{4.90}$$

The maximum data rate is then 12 bits per symbol or 1.5 bits/dimension. The new energy distribution is

$$.7520 \leftarrow \mathcal{E}_0 = \frac{8}{7} \cdot \frac{2^{2\cdot 2}-1}{2^{2\cdot 2.28}-1} \tag{4.91}$$
$$1.7614 \leftarrow \mathcal{E}_1 = \frac{16}{7} \cdot \frac{2^{4}-1}{2^{4.355}-1} \tag{4.92}$$
$$3.000 \leftarrow \mathcal{E}_2 = \frac{16}{7} \cdot \frac{2^{4}-1}{2^{3.6356}-1} \tag{4.93}$$
$$2.0215 \leftarrow \mathcal{E}_3 = \frac{16}{7} \cdot \frac{2^{2}-1}{2^{2.1349}-1} \tag{4.94}$$

The total energy is then $\mathcal{E}_{\boldsymbol{x}} = 7.5349$, which means that the distribution of energies may all be increased by the factor 8/7.5349 so that the total energy budget is maintained. The loss in performance caused by quantization in this example is

$$\underbrace{10 \cdot \log_{10}\left[(2^3-1)/(2^{3.1}-1\right]}_{\text{loss-in-data-rate term}} - \underbrace{10 \cdot \log_{10}(8/7.5349)}_{\text{energy-gain term}} \tag{4.95}$$

or -.08 dB, pretty small!

The example's use of $\beta = 1$ for the complex subchannels was essentially equivalent to a number of bits/dimension granularity of 1/2 bit. The granularity thus need not be the same on all subchannels as long as each subchannel is handled consistently within the constraints of the granularity of its specific encoding scheme.

EXAMPLE 4.3.5 (DSL rate adaption) A number of international DSL standards (ITU G.992.1 or ADSL1), ITU G.992.5 or ADSL2+ , and G.993.2 or VDSL2) use a form of multitone described more fully in Section 4.5. These allow high-speed digital transmission on

phone lines with one modem in the telephone company central office and the other at the customer. These all have a mode of operation for internet access where rate-adaptive loading is used. The symbol rate is effectively 4000 Hz in these methods, meaning the tone width is effectively 4000 Hz. The granularity in the standards is $\beta = 1$ bit per two-dimensional subchannel. The ADSL1 standard uses 256 tones in one direction of transmission known as "downstream" and 32 tones upstream, the asymmetry being a match to usually asymmetric file transfers and acknowledgment packet-length ratios in IP traffic for internet use. ADSL2+ double the number of tones in both directions, while VDSL2 can use up to 4096 tones in either direction (and allows doubling of tone width to 8 kHz).

Thus the data rate can be any multiple of 4kbps with RA loading. Usually in these systems, the gap is intentionally increased by 6 dB - this is equivalent to forcing 6 dB of margin when RA loading completes. With nominally 100-250 tones being used and average numbers of bits per tone of 2-9 bits, data rates from 1 to 8 Mbps are possible in ADSL1 (and up to 150 Mbps in VDSL2).

Chow's MA Loading algorithm Chow's MA algorithm first computes a margin, from the data rate given by the Chow Primer, with respect to the desired fixed rate. It then recomputes the bit distribution including that margin, which will make the number of bits exactly equal to the desired number. However, the number of bits will still not be integer multiples of β in general. The resultant bit distribution is then rounded as in the RA algorithm, but the sum of the number of bits is monitored and rounding is adjusted to make the sum exactly equal to the desired number of bits per symbol.

The steps that follow the Chow Primer to complete the MA solution are:

6. Compute

$$\mathrm{SNR}_{m,u} = \Gamma \cdot \left\{ \left(\prod_{n=1}^{i} \left[1 + \frac{\mathrm{SNR}_n}{\Gamma} \right] \right)^{1/i} - 1 \right\} = 2^{2\bar{b}_{max}} - 1 \quad , \tag{4.96}$$

7. Compute the tentative margin

$$\gamma_{temp,m} = \frac{\frac{\mathrm{SNR}_{m,u}}{\Gamma}}{2^{2\bar{b}} - 1} \quad n = 1, ..., N^* \quad , \tag{4.97}$$

8. Compute the updated bit distribution

$$B_{temp,n} = \frac{1}{2\beta} \cdot \log_2 \left(1 + \frac{\mathrm{SNR}_n}{\Gamma \cdot \gamma_{temp,n}} \right) \quad n = 1, ..., N^* \quad , \tag{4.98}$$

9. Round $B_{temp,n}$ to the nearest integer (if more than one beta is being used for different subchannels, then use the correct value for each subchannel) and recompute energy as

$$\mathcal{E}_{temp,n} = \frac{\Gamma \cdot \left(2^{B_n \cdot \beta} - 1 \right)}{g_n} \quad n = 1, ..., i \quad , \tag{4.99}$$

10. Compute

$$B_{temp} = \sum_{n=1}^{N^*} B_{temp,n} \quad ; \quad b_{temp} = B_{temp} \cdot \beta \tag{4.100}$$

and check to see if $b_{temp} = b$, the desired fixed number of bits per symbol. If not, then select subchannels that were rounded (perhaps by choosing those close to one-half information granularity unit) and round the other way until the data rate is correct, recomputing energies as in step 9.

11. Compute the energy scaling factor

$$f_e = \frac{N \bar{\mathcal{E}}_{\boldsymbol{x}}}{\sum_{n=1}^{N^*} \mathcal{E}_n} \tag{4.101}$$

and scale all subchannel energies to

$$\mathcal{E}_n \rightarrow f_e \cdot \mathcal{E}_n \quad , \tag{4.102}$$

and the maximum margin becomes

$$\gamma_{max} = \gamma_{temp,max} \cdot f_e \quad . \tag{4.103}$$

EXAMPLE 4.3.6 (MA loading for $1 + .9D^{-1}$ channel with Γ =0 dB) The first step in Chow's MA procedure finds $\text{SNR}_{m,u}$=8.8 dB and the margin with respect to transmission at $\bar{b} = 1$ is then $10^{.88}/(2^2 - 1) = 4.0$ dB. The updated bit distribution computed from step 8 of Chow's MA loading is

$$b_0 = \frac{1}{2}\log_2\left(1 + \frac{(8/7)\cdot 19.948}{10^{.4}}\right) = 1.7 \tag{4.104}$$

$$b_1 = \log_2\left(1 + \frac{(8/7)\cdot 17.032}{10^{.4}}\right) = 3.1 \tag{4.105}$$

$$b_2 = \log_2\left(1 + \frac{(8/7)\cdot 10}{10^{.4}}\right) = 2.47 \tag{4.106}$$

$$b_3 = \log_2\left(1 + \frac{(8/7)\cdot 2.968}{10^{.4}}\right) = 1.2 \quad . \tag{4.107}$$

For $\beta = 1$, this bit distribution rounds to $b_0 = 2$, $b_1 = 3$, $b_2 = 2$, and $b_3 = 1$. The number of bits sums to 8 exactly, and so the procedure concludes. The energy distribution then changes to

$$\mathcal{E}_0 \leftarrow 1.7939 = \frac{8}{7}\cdot\frac{2^4 - 1}{2^{1.7} - 1} \tag{4.108}$$

$$\mathcal{E}_1 \leftarrow 2.1124 = \frac{16}{7}\cdot\frac{2^3 - 1}{2^{3.1} - 1} \tag{4.109}$$

$$\mathcal{E}_2 \leftarrow 1.5102 = \frac{16}{7}\cdot\frac{2^2 - 1}{2^{2.4} - 1} \tag{4.110}$$

$$\mathcal{E}_3 \leftarrow 1.7618 = \frac{16}{7}\cdot\frac{2^1 - 1}{2^{1.2} - 1} \tag{4.111}$$

Total energy is then 7.1783 and all subchannels have an additional margin of of 8/7.1783 leading to an additional margin of about .5 dB, so that the total is then 4.5 dB, when a system with powerful code (say gap is 0 dB at $P_e = 10^{-6}$) is used to transmit on the $1 + .9D^{-1}$ channel at $\bar{b} = 1$ and the same P_e as the gap used in loading.

Had $\beta = .5$ for these subchannels, then the bit distribution would have been $b_0 = 1.5$, $b_1 = 3$, $b_2 = 2.5$, and $b_3 = 1$, which still adds to 8.

Optimum Discrete Loading Algorithms

Optimum discrete loading algorithms recognize the discrete loading problem as an instance of what is known as "greedy optimization" in mathematics. The basic concept is that each increment of additional information to be transported by a multi-channel transmission system is placed on the subchannel that would require the least incremental energy for such transport. Such algorithms are optimum for loading when the information granularity β is the same for all subchannels, which is usually the case.[14]

Several definitions are necessary to formalize discrete loading.

[14]The $1 + .9D^{-1}$ channel example has a PAM subchannel at DC for which this chapter has previously used a granularity of β=1 bit per dimension while the other subchannels were QAM and used granularity of 1 bit per two dimensions, or equivalently $\beta = 1/2$ bit per dimension – this anomaly rarely occurs in practice because DC is almost never passed in transmission channels and complex-baseband channels use a two-dimensional QAM subchannel at the baseband-equivalent of DC. Thus, for practical systems β is constant on all subchannels.

Definition 4.3.4 (Bit distribution vector) *The* **bit distribution vector** *for a set of parallel subchannels that in aggregate carry b total bits of information is given by*

$$\boldsymbol{b} = [b_1\ b_2\ ...\ b_N] \quad . \tag{4.112}$$

The sum of the elements in $\boldsymbol{b}$ is clearly equal to the total number of bits transmitted, b.

Discrete loading algorithms can use any monotonically increasing relation between transmit symbol energy and the number of bits transmitted on any subchannel that is not a function of other subchannel's energies[15]. This function can be different for each subchannel, and there need not be a constant gap used. In general, the concept of incremental energy is important to discrete loading:

Definition 4.3.5 (Incremental Energy and Symbol Energy) *The symbol energy $\mathcal{E}_n$ for an integer number of information units $b_n = B_n \cdot \beta$ on each subchannel can be notationally generalized to the* **energy function**

$$\mathcal{E}_n \rightarrow \mathcal{E}_n(b_n) \quad , \tag{4.113}$$

where the symbol energy's dependence on the number of information units transmitted, b_n, is explicitly shown. The **incremental energy** *to transmit b_n information units on a subchannel is the amount of additional energy required to send the B_n^{th} information unit with respect to the $(B_n - 1)^{th}$ information unit (that is, one more unit of β). The incremental energy is then*

$$e_n(b_n) \triangleq \mathcal{E}_n(b_n) - \mathcal{E}_n(b_n - \beta) \quad . \tag{4.114}$$

EXAMPLE 4.3.7 (Incremental Energies) As an example, uncoded SQ QAM with $\beta =$ 1 bit-per-two-dimensions, and minimum input distance between points on subchannel n being d_n, has energy function

$$\mathcal{E}(b_n) = \begin{cases} \frac{2^{b_n}-1}{6} d_n^2 & b_n \text{ even} \\ \frac{2^{b_n+1}-1}{12} d_n^2 & b_n \text{ odd} \end{cases} \quad . \tag{4.115}$$

The incremental energy is then

$$e_n(b_n) = \begin{cases} \frac{2^{b_n}-1}{12} d_n^2 & b_n \text{ even} \\ \frac{2^{b_n}+1}{12} d_n^2 & b_n \text{ odd} \end{cases} \quad . \tag{4.116}$$

For large numbers of bits, the incremental energy for SQ QAM is approximately twice the amount of energy needed for the previous constellation, or 3 dB per bit as is well known to the reader of Chapter 1. A coded system with $\beta \neq 1$ might have a more complicated exact expression that perhaps is most easily represented by tabulation in general. For instance, the table for the uncoded SQ QAM is

b_n	0	1	2	3	4	5	6	7	8
$\mathcal{E}_n(b_n)$	0	$\frac{d_n^2}{4}$	$\frac{d_n^2}{2}$	$\frac{5d_n^2}{4}$	$\frac{5d_n^2}{2}$	$\frac{21d_n^2}{4}$	$\frac{21d_n^2}{2}$	$\frac{85d_n^2}{4}$	$\frac{85d_n^2}{2}$
$e_n(b_n)$	0	$\frac{d_n^2}{4}$	$\frac{d_n^2}{4}$	$\frac{3d_n^2}{4}$	$\frac{5d_n^2}{2}$	$\frac{11d_n^2}{4}$	$\frac{21d_n^2}{4}$	$\frac{43d_n^2}{4}$	$\frac{85d_n^2}{4}$

The distance d in this table is selected for a given desired probability of error (and the gap approximation is thus avoided) since it is an approximation.[16]

Alternatively, an energy function for QAM with $\beta = 1$ could also be defined via the gap approximation as

$$\mathcal{E}_n(b_n) = 2 \cdot \frac{\Gamma}{g_n} \left(2^{b_n} - 1\right) \quad . \tag{4.117}$$

[15]Such systems can occur in multi-user systems that have essentially suchannels per user as well as frequency. Chapters 13-15 address loading for multi-user channels.

[16]The $\text{SNR}_{m.u}$ in this case of no gap is obtained by calculated the usual formula in (4.17) with $\Gamma = 1$.

The incremental energy with $\beta = 1$ is then

$$\begin{aligned} e_n(b_n) &= \frac{\Gamma}{g_n}\left(2 \cdot 2^{b_n} - 2^{b_n}\right) && (4.118)\\ &= \frac{\Gamma}{g_n} \cdot 2^{b_n}(2-1) && (4.119)\\ &= \frac{\Gamma}{g_n} 2^{b_n} && (4.120)\\ &= 2 \cdot e_n(b_n - 1) \quad , && (4.121) \end{aligned}$$

which is exactly 3 dB per incremental bit. If $\beta = .25$, then the gap-based incremental QAM energy formula would be

$$e_n(b_n) = \frac{\Gamma}{g_n} 2^{b_n+1}\left(2^{.25} - 1\right) \quad , \tag{4.122}$$

and in general for gap-based QAM

$$e_n(b_n) = \frac{\Gamma}{g_n} 2^{b_n+1}\left(2^{\beta} - 1\right) \quad . \tag{4.123}$$

Once an encoding system has been selected for the given fixed β and for each of the subchannels in a multichannel transmission system, then the incremental energy for each subchannel can be tabulated for use in loading algorithms.

Clearly, there are many possible bit distributions that all sum to the same total b. A highly desirable property of a distribution is efficiency:

Definition 4.3.6 (Efficiency of a Bit Distribution) *A bit distribution vector* $\boldsymbol{b}$ *is said to be* **efficient** *for a given granularity* β *if*

$$\max_n \left[e_n(b_n)\right] \leq \min_m \left[e_m(b_m + \beta)\right] \quad . \tag{4.124}$$

Efficiency means that there is no movement of a bit from one subchannel to another that reduces the symbol energy. An efficient bit distribution clearly solves the MA loading problem for the given total number of bits b. An efficient bit distribution also solves the RA loading problem for the energy total that is the sum of the energies $\mathcal{E}_n(b_n)$.

Levin and Campello have independently formalized an iterative algorithm that will translate any bit distribution into an efficient bit distribution:

Levin-Campello (LC) "Efficientizing" (EF) Algorithm

1. $m \leftarrow \arg\left\{\min_{1 \leq i \leq N}\left[e_i(b_i + \beta)\right]\right\}$;
2. $n \leftarrow \arg\left\{\max_{1 \leq j \leq N}\left[e_j(b_j)\right]\right\}$;
3. While $e_m(b_m + \beta) < e_n(b_n)$ do
   (a) $b_m \leftarrow b_m + \beta$
   (b) $b_n \leftarrow b_n - \beta$
   (c) $m \leftarrow \arg\left\{\min_{1 \leq i \leq N}\left[e_i(b_i + \beta)\right]\right\}$;
   (d) $n \leftarrow \arg\left\{\max_{1 \leq j \leq N}\left[e_j(b_j)\right]\right\}$;

By always replacing a bit distribution with another distribution that is closer to efficient, and exhaustively searching all single-information-unit changes at each step, the EF algorithm will produce an efficient bit distribution if the energy function is monotonically increasing with information (as always the case in practical systems). Truncation of a water-filling distribution produces an efficient distribution.

EXAMPLE 4.3.8 (Return to $1 + .9D^{-1}$ for EF algorithm) By using the using the uncoded gap approximation at $P_e = 10^{-6}$, the PAM and SSB subchannels' energy is given by

$$\mathcal{E}_n(b_n) = \frac{10^{.88}}{g_n}\left(2^{2b_n} - 1\right) \tag{4.125}$$

and the QAM subchannels' energies are given by

$$\mathcal{E}_n(b_n) = 2 \cdot \frac{10^{.88}}{g_n}\left(2^{b_n} - 1\right) \quad . \tag{4.126}$$

The incremental energies $e_n(b_n)$ then are:

n	0	1	2	3	4
$e_n(1)$	1.14	.891	1.50	5.112	412.3
$e_n(2)$	4.56	1.78	3.03	10.2	–
$e_n(3)$	18.3	3.56	6.07	20.4	–
$e_n(4)$	–	7.13	12.1	–	–
$e_n(5)$	–	14.2	24.3	–	–

If the initial condition were $\boldsymbol{b} = [0\ 5\ 0\ 2\ 1]$, the sequence of steps would be:
1. $\boldsymbol{b} = [0\ 5\ 0\ 2\ 1]$
2. $\boldsymbol{b} = [1\ 5\ 0\ 2\ 0]$
3. $\boldsymbol{b} = [1\ 4\ 1\ 2\ 0]$
4. $\boldsymbol{b} = [1\ 4\ 2\ 1\ 0]$
5. $\boldsymbol{b} = [2\ 3\ 2\ 1\ 0]$.

The EF algorithm retains the total number of bits at 8 and converges to the best solution for that number of bits from any initial condition.

The Levin-Campello RA and MA loading algorithms both begin with the EF algorithm.

Levin-Campello RA Solution An additional concept of **E-tightness** is necessary for solution of the RA problem:

Definition 4.3.7 (E-tightness) *A bit distribution with granularity β is said to be* **E-tight** *if*

$$0 \le N\bar{\mathcal{E}}_{\boldsymbol{x}} - \sum_{n=1}^{N} E_n(b_n) \le \min_{1 \le i \le N}\left[e_i(b_i + \beta)\right] \quad . \tag{4.127}$$

E-tightness implies that no additional unit of information can be carried without violation of the total energy constraint.

To make a bit distribution E-tight, one uses the Levin-Campello E-tightening (ET) algorithm below:

1. Set $S = \sum_{n=1}^{N} \mathcal{E}_n(b_n)$;
2. WHILE $\left(N\bar{\mathcal{E}}_{\boldsymbol{x}} - S < 0\right)$ or $\left(N\bar{\mathcal{E}}_{\boldsymbol{x}} - S \ge \min_{1 \le i \le N}\left[e_i(b_i + \beta)\right]\right)$
   IF $\left(N\bar{\mathcal{E}}_{\boldsymbol{x}} - S < 0\right)$ THEN
   (a) $n \leftarrow \arg\left\{\max_{1 \le i \le N}\left[e_i(b_i)\right]\right\}$;
   (b) $S \leftarrow S - e_n(b_n)$
   (c) $b_n \leftarrow b_n - \beta$

   ELSE
   (a) $m \rightarrow \arg\left\{\min_{1 \le i \le N}\left[e_i(b_i + \beta)\right]\right\}$;
   (b) $S \leftarrow S + e_m(b_m + \beta)$
   (c) $b_m \leftarrow b_m + \beta$

The ET algorithm reduces the number of bits when the energy exceeds the limit. The ET algorithm adds additional bits in the least energy-consumptive places when energy is sufficiently below the limit.

EXAMPLE 4.3.9 (ET algorithm on EF output in example above) If the initial bit distribution vector for the above example were the EF solution $\boldsymbol{b} = [2\ 3\ 2\ 1\ 0]$, the sequence of ET steps lead to:
1. $\boldsymbol{b} = [2\ 3\ 2\ 0\ 0]$ with $\mathcal{E} \rightarrow 16.49$
2. $\boldsymbol{b} = [1\ 3\ 2\ 0\ 0]$ with $\mathcal{E} \rightarrow 11.92$
3. $\boldsymbol{b} = [1\ 2\ 2\ 0\ 0]$ with $\mathcal{E} \rightarrow 8.36$
4. $\boldsymbol{b} = [1\ 2\ 1\ 0\ 0]$ with $\mathcal{E} \rightarrow 5.32$

Thus, the data rate is halved in this example for E-tightness and the resultant margin is $8/5.32 = 1.8$dB.

If the ET algorithm is applied to an efficient distribution, the RA problem is solved:

The Levin-Campello RA Algorithm

1. Choose any $\boldsymbol{b}$
2. Make $\boldsymbol{b}$ efficient with the EF algorithm.
3. E-tighten the resultant $\boldsymbol{b}$ with the ET algorithm.

A program called LC.m for RA LC is available at the EE379C web site and listed here:

```
function [gn,En,bn,b_bar]=LC(P,SNRmfb,Ex_bar,Ntot,gap)

% Levin Campello's Method
%
% P is the pulse response
% SNRmfb is the SNRmfb in dB
% Ex_bar is the normalized energy
% Ntot is the total number of real/complex subchannels, Ntot>2
% gap is the gap in dB
%
% gn is channel gain
% En is the energy in the nth subchannel (PAM or QAM)
% bn is the bit in the nth subchannel (PAM or QAM)
% Nstar is the number of subchannel used
% b_bar is the bit rate
%
% The first bin and the last bin is PAM, the rest of them are QAM.

% dB into normal scale
Noise_var=Ex_bar*(norm(P)^2)/(10^(SNRmfb/10));
gap=10^(gap/10);

% initialization
En=zeros(1,Ntot/2+1);
bn=zeros(1,Ntot/2+1);
gn=zeros(1,Ntot/2+1);
Hn = zeros(1,Ntot/2+1);
decision_table=zeros(1,Ntot/2+1);

% subchannel center frequencies
f=0:1/Ntot:1/2;
```

```
% find Hn vector
for i=1:length(P)
Hn=Hn+P(i)*exp(j*2*pi*f*(i-1));
        % This value will be different depending if P represents
        % P(1) + P(2)*D^-1 + ....  or P(1) + P(2)*D^+1....,
        % but we'll get same gn, thus same waterfilling result.
        % (Note that both have the same magnitude response!)
end

% find gn vector
gn=abs(Hn).^2/Noise_var;

%debugging purpose
%plot(gn)

%%%%%%%%%%%%%%%%%%%%%%
% Now do LC %
%%%%%%%%%%%%%%%%%%%%%%

%initialization

%used energy so far
E_so_far=0;
%decision table - QAM and PAM
decision_table(2:Ntot/2)=2*gap./gn(2:Ntot/2);
if gn(1) ~= 0
    decision_table(1)=3*gap/gn(1);
else
    decision_table(1)=inf;
end
if gn(Ntot/2+1) ~=0
    decision_table(Ntot/2+1)=3*gap/gn(Ntot/2+1);
else
    decision_table(Ntot/2+1)=inf;
end

%decision_table: debugging purpose
while(1)
    [y,index]=min(decision_table);
    E_so_far=E_so_far+y;
    if E_so_far > Ex_bar*Ntot
        break;
    else

        En(index)=En(index)+y;
        bn(index)=bn(index)+1;

if (index ==1 | index == Ntot/2+1)
  decision_table(index)=4*decision_table(index);
else
  decision_table(index)=2*decision_table(index);
end
    end
```

```
end

% calculate b_bar
b_bar=1/Ntot*(sum(bn));

%>> [gn,En,bn,b_bar]=LC([1 0.9],10,1,8,0)
%
%gn = 19.9448   17.0320   10.0000   2.9680     0.0552
%
%En = 0.7521    1.7614    3.0000    2.0216     0
%
%bn = 2      4        4         2          0
%
%b_bar =    1.5000
```

The Levin-Campello MA Solution The dual of E-tightness for MA loading is B-tightness:

Definition 4.3.8 (B-tightness) *A bit distribution* $\boldsymbol{b}$ *with granularity* β *and total bits per symbol* b *is said to be* **B-tight** *if*

$$b = \sum_{n=1}^{N} b_n \quad . \tag{4.128}$$

B-tightness simply states the correct number of bits is being transmitted.

A B-tightening (BT) algorithm is

1. Set $\tilde{b} = \sum_{n=1}^{N} b_n$
2. WHILE $\tilde{b} \neq b$
   IF $\tilde{b} > b$
   (a) $n \leftarrow \arg\{\max_{1 \le i \le N} [e_i(b_i)]\}$;
   (b) $\tilde{b} \leftarrow \tilde{b} - \beta$
   (c) $b_n \leftarrow b_n - \beta$

   ELSE
   (a) $m \leftarrow \arg\{\min_{1 \le i \le N} [e_i(b_i + \beta)]\}$;
   (b) $\tilde{b} \leftarrow \tilde{b} + \beta$
   (c) $b_m \leftarrow b_m + \beta$

EXAMPLE 4.3.10 (Bit tightening on $1 + .9D^{-1}$ **channel)** Starting from a distribution of $\boldsymbol{b} = [0\ 0\ 0\ 0\ 0]$, the BT algorithm steps produce:
1. $\boldsymbol{b} = [0\ 0\ 0\ 0\ 0]$
2. $\boldsymbol{b} = [0\ 1\ 0\ 0\ 0]$
3. $\boldsymbol{b} = [1\ 1\ 0\ 0\ 0]$
4. $\boldsymbol{b} = [1\ 1\ 1\ 0\ 0]$
5. $\boldsymbol{b} = [1\ 2\ 1\ 0\ 0]$
6. $\boldsymbol{b} = [1\ 2\ 2\ 0\ 0]$
7. $\boldsymbol{b} = [1\ 3\ 2\ 0\ 0]$
8. $\boldsymbol{b} = [2\ 3\ 2\ 0\ 0]$
9. $\boldsymbol{b} = [2\ 3\ 2\ 1\ 0]$

The margin for this distribution has previously been found as -4.3 dB because it is the same distribution produced in the Chow method. The total energy is 21.6.

The BT algorithm reduces bit rate when it is too high in the place of most energy reduction per unit of information and increases bit rate when it is too low in the place of least energy increase per unit of information. If the BT algorithm is applied to an efficient distribution, the MA problem is solved.

The Levin-Campello MA Algorithm

1. Choose any $\boldsymbol{b}$
2. Make $\boldsymbol{b}$ efficient with the EF algorithm.
3. B-tighten the resultant $\boldsymbol{b}$ with the BT algorithm.

Some transmission systems may use coding, which means that there could be redundant bits. Ideally, one could compute the incremental energy table with the effect of redundancy included in the incremental-energy entries. This can be quite difficult to compute. Generally more redundant bits means higher gain but also allocation to increasingly more difficult subchannel positions, which at some number of redundant bits becomes a coding loss.

A program for MA LC is called MALC.m and at the EE379C website:

```
function [gn,En,bn,b_bar_check,margin]=MALC(P,SNRmfb,Ex_bar,b_bar,Ntot,gap)

%
% Levin Campello's Method
%
% P is the pulse response
% SNRmfb is the SNRmfb in dB
% Ex_bar is the normalized energy
% Ntot is the total number of real/complex subchannels, Ntot>2
% gap is the gap in dB
% b_bar is the bit rate

% gn is channel gain
% En is the energy in the nth subchannel (PAM or QAM)
% bn is the bit in the nth subchannel (PAM or QAM)
% b_bar_check is the bit rate for checking - this should be equal to b_bar
% margin is the margin

% The first bin and the last bin is PAM, the rest of them are QAM.

% dB into normal scale
Noise_var=Ex_bar*(norm(P)^2)/(10^(SNRmfb/10));
gap=10^(gap/10);

% initialization
En=zeros(1,Ntot/2+1);
bn=zeros(1,Ntot/2+1);
gn=zeros(1,Ntot/2+1);
Hn = zeros(1,Ntot/2+1);
decision_table=zeros(1,Ntot/2+1);

% subchannel center frequencies
f=0:1/Ntot:1/2;

% find Hn vector
for i=1:length(P)
Hn=Hn+P(i)*exp(j*2*pi*f*(i-1));
        % This value will be different depending if P represents
        % P(1) + P(2)*D^-1 + ....  or P(1) + P(2)*D^+1....,
        % but we'll get same gn, thus same waterfilling result.
        % (Note that both have the same magnitude response!)
```

```
end

% find gn vector
gn=abs(Hn).^2/Noise_var;

%debugging purpose
%plot(gn)

%%%%%%%%%%%%%%%%%%%%%%%
% Now do LC %
%%%%%%%%%%%%%%%%%%%%%%%

%initialization

%used energy so far
E_so_far=0;
%decision table - QAM and PAM
decision_table(2:Ntot/2)=2*gap./gn(2:Ntot/2);
if gn(1) ~= 0
    decision_table(1)=3*gap/gn(1);
else
    decision_table(1)=inf;
end
if gn(Ntot/2+1) ~=0
    decision_table(Ntot/2+1)=3*gap/gn(Ntot/2+1);
else
    decision_table(Ntot/2+1)=inf;
end

%decision_table: debugging purpose
while(1)
    [y,index]=min(decision_table);
    E_so_far=E_so_far+y;
    if sum(bn) >= Ntot*b_bar
        break;
    else
        En(index)=En(index)+y;
        bn(index)=bn(index)+1;
if (index ==1 | index == Ntot/2+1)
  decision_table(index)=4*decision_table(index);
else
  decision_table(index)=2*decision_table(index);
end
    end
end

% calculate b_bar
b_bar_check=1/Ntot*(sum(bn));

% check margin
margin=10*log10(Ntot*Ex_bar/sum(En));
```

4.3.4 Sorting and Run-time Issues

Campello in pursuit of the optimized discrete loading algorithm, studied further the issue of computational complexity in loading algorithms. Mathematically, a sort of the vector $\boldsymbol{g}$ (the vector $[g_1\ , ...,\ g_N]$) can be denoted

$$\boldsymbol{g}_{sort} = J\boldsymbol{g} \tag{4.129}$$

where J is a matrix with one and only one nonzero unit value in each and every row and column. For instance, the $N = 3$ matrix

$$J = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{4.130}$$

takes b_3 and moves it to the first position $3 \rightarrow 1$, moves the first position to the second position $1 \rightarrow 2$ and finally places the second position last $2 \rightarrow 3$. To unsort,

$$\boldsymbol{g} = J^*\boldsymbol{g}_{sort} \tag{4.131}$$

since $J^{-1} = J^*$ is also a sorting matrix.

Water-filling and Chow's methods begin with a sort of the channel SNR quantities g_n. Finding a maximum in a set of N numbers takes $N-1$ comparisons and then subsequently finding the minimum takes $N-2$ comparisons to order the g_n. Continuing in such fashion would require $N^2/2$ comparisons. Binary sort algorithms instead select an element at random from the set to be ordered and then partition this set into two subsets containing elements greater than and smaller than the first selected element. The process is then repeated recursively on each of the subsets with a minimum computation of approximately $Nlog_2(N)$ and a maximum still at $N^2/2$. The average over a uniform distribution of all possible starting orders of the vector $\boldsymbol{g}$ has computation proportional to $Nlog_2(N)$.

Campello noted that for water-filling and Chow's algorithms, absolute ordering is not essential. The important quantity is the position of the transition from "on" to "off' in energy. Rather than sort the SNR's initially, a random guess at the transition point can be made instead. Then, with $N-1$ additional comparisons, the quantities g_n can be sorted into two subsets greater than or less than or equal to the transition point. Then water-filling or Chow's algorithm can compute the constant K and test the transition point subchannel for negative energy. If negative, the set of larger g_n's again has contains the correct transition point. If positive, then the point is the transition point or it is in the set of lesser g_n's. The amount of computation is proportional to $NlogN$ comparisons.

The Levin-Campello EF algorithm routinely must also know the "position" bit (information unit) presently absorbing the most energy and the position of an additional bit that would require the least energy. At each step, the incremental energies would at least have to look at the two new incremental energies and place them accordingly (requiring perhaps up to $2N$ cmparisons). This number of comparisons could be quite high for each successive bit loaded. Campello further studied sets of tones, which he notes is an instance of the computer-scientists' concept of "heaps" in data bases. Essentially, there are two heaps that are indexed by two randomly selected incremental energies. The first heap contains incremental energies that are larger than that of its index. The second contains only incremental energies less than its index (which has index' energy less than the first heap). Bits are assigned to all the places in the second heap. If total energy (RA) or rate (MA) are exceeded, then the second group is recursively divided into two sub-heaps. If the limits are not exceeded, then the first heap is recursively divided into two heaps. Campello was able to show that the average complexity is proportional to N (that is less than Chow's $NlogN$ on average). An unfavorable selection could, however, have very high computation but with low probability.

Generally, the sort is not an enormous complexity for initializing the multi-channel modem as it may be much less computation than other early signal-processing procedures. However, if the channel changes, an optimized design might need to repeat the procedure. This is the topic of the next subsection.

4.3.5 Dynamic Loading

In some applications, the subchannel SNR's, g_n, may change with time. Then, the bit and energy distribution may need corresponding change. Such variation in the bit and energy distribution is known

as **dynamic loading**, or more specifically **dynamic rate adaption (DRA)** or **dynamic margin adaption (DMA)**, the later which was once more colloquially called **"bit-swapping,"** by this author and others – that later name is in much wider use than "DMA".

It is conceivable that an entire loading algorithm could be run every time there is a change in the bit and/or energy distribution for a channel. However, if the channel variation has been small, then "reloading" might produce a bit distribution with only minor variation from the original. Generally, it is more efficient to update loading dynamically than to reload. Usually, dynamic loading occurs through the transmission of bit/energy-distribution-change information over a reverse-direction reliable low-bit-rate channel. The entire bit distribution need not be re-communicated, rather compressed incremental variation information is desirable to minimize the relative data rate of the reverse channel. Dynamic loading is not possible without a reverse channel, and so could thus not be used in broadcast TV or radio – it is also difficult to use on rapidly varying channels like mobile wireless channels (because the feedback channel is usually not faster than the rate of variation of the channel).

The LC algorithms lend themselves most readily to incremental variation. Specifically, for continuous solution of the MA problem, the previous bit distribution $\boldsymbol{b}$ can be input to the EF algorithm as an initial condition for the new table of incremental energies, $e_n(b_n)$, which is simply scaled by $g_{n,old}/g_{n,new}$ for any reasonable constellation. Steps 3(a) and 3(b) of the EF algorithm execute a "swap" whenever it is more efficient energy-wise to move a bit from one subchannel to another. As long as a sufficient number of swaps can occur before further variation in the channel, then the EF algorithm will continue to converge to the optimum finite-granularity solution. The incremental information is the location of the two subchannels for the "swap" and which is to be decremented. Such information has very low bit rate, even for large numbers of subchannels. The energy gain/loss on each of the two subchannels must also be transferred (from which the new incremental energy added/deleted from swapping subchannels can be computed) with use of the EF algorithm.

Both the transmitter and the receiver need not compute (use at transmitter) the same energy. Any deviation between the two effectively causes the channel to appear different. For example a transmitter using 3dB less energy that the receiver presumed would result in the receiver thining the channel had higher attenuation by that same factor. Loading algorithms may have a small error in this case relative to optimum. Of course, if the two have agreed on the exact transmit power level by standard or inital hand-shaking, then the loading is optimum. To avoid confusion between transmitter and receiver on absolute levels used, most bit-swapping procedures send relative gain (scale factors) from receiver back to the transmitter to increase/decrease transmitted energy on the corresponding subchannel (whatever that might have been initially) by the corresponding factor.

Bit-swapping systems in practice could also allow the addition of a bit or the deletion of a bit without a corresponding action on a second tone, allowing DRA. However, such a system would have continuously varying data rate. The instance of continuous bit-rate variation to correspond to channel change is thus rare (usually some higher-level communications authority prefers to arbitrate when bit rate changes are allowed to occur, rather than leave it to the channel). These systems allow such rate adaption between some finite rate range specified by a minimum and maximum data rate (to prevent buffer overflow or underflow in outer sytems, or equivalently to limit maximum delay), and this is sometimes called "seamless rate adaption." However, the consequent delay increase to accommodate such variation is often undesirable.

However, such true DRA can occur through the execution of the ET algorithm theoretically. That algorithm will change the number of bits total over the subchannels to conform to the best use of limited transmit energy. The efficiency of the resultant bit distribution will be maintained by the execution of the ET algorithm. The incremental information passed through the reliable reverse channel would be the index of the currently to-be-altered subchannel and whether a bit is to be added or deleted.

More practical is actually a dynamic fixed-margin adaption where the data rate is maintained, but power is reduced to the minimum necessary for a given fixed maximum margin. This saves power whenever channel conditions have less than the maximum noise (minimum SNR) present.

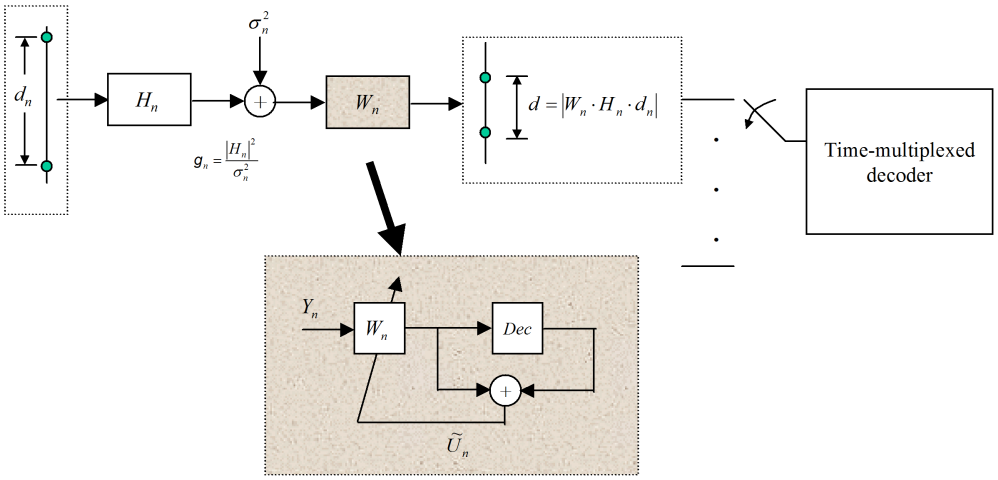


Figure 4.11: A multichannel normalizer for one subchannel, n.

Synchronization of dynamic loading

For maintenance of the channel, both transmitter and receiver need to know on which symbol a bit/energy distribution change has occurred. Thus two synchronized symbol counters must be maintained, one in the transmitter and one in the receiver. These counters can be circular with the same period, which is longer than the worst-case time that it takes for a bit-swap, bit-add, or bit-delete to occur. Upon receipt of a request for a change, a transmitter then acknowledges the change and returns a counter symbol number on which it intends to execute the change. Then both transmitter and receiver execute the change simultaneously on the symbol corresponding to the agreed counter value. Often extra redundancy is used on commands executing bit-swaps to ensure a higher reliability of the swap's correct synchronized implementation. However, a good receiver design (exploiting outer hard error-correcting codes, see Chapter 11) can determine rapidly without acknowledgement if a swap has been implemented, which allows a more rapid execution of multiple successive swaps (because the acknowledgements take time).

The multichannel normalizer and computing the new subchannel SNR's

Figure 4.11 illustrates a multi-channel normalizer for one subchannel. There is a normalizer for each and every subchannel. The multi-channel normalizer has no direct impact on performance because both signal and noise are scaled by the same value W_n. When multitone transmission is used, W_n is complex and often called a **Frequency EQualizer (FEQ)**. Nominally, the multi-channel normalizer allows a single decision device to be time-shared for decoding all the subchannel outputs. Such normalization of distance satisfies:

$$d = |W_n \cdot H_n \cdot d_n| \quad , \tag{4.132}$$

where d_n is the minimum distance between points on the n^{th} subchannel constellation on the transmitter side. The normalizer forces all subchannels to have output distance d, independent of the number of bits carried or signal to noise ratio. Clearly,

$$W_n = \frac{d}{H_n \cdot d_n} \quad . \tag{4.133}$$

However, since the subchannel may change its gain or SNR, the value for W_n is adaptively computed periodically (as often as every symbol is possible). The error, $\tilde{U}_n$ between the input and output of decision device shown in the enlarged normalizer box estimates the normalized noise (as long as the decision is correct). The updated value for W_n can be computed in a number of ways that average several successive normalized errors, of which the two most popular are the **zero-forcing algorithm**

$$W_{n,k+1} = W_{n,k} + \mu \tilde{U}_n X_n^* \quad , \tag{4.134}$$

which produces an unbiased estimate of the signal at the normalizer output and converges[17] when

$$0 \leq \mu < \frac{\mathcal{E}_x}{2} \quad , \tag{4.135}$$

equivalently $E[\tilde{U}_n/X_n]$ contains only noise; or the more common **MMSE algorithm**

$$W_{n,k+1} = W_{n,k} + \mu \tilde{U}_n Y_n^* \quad , \tag{4.136}$$

which instead produces a biased estimate and an optimistic SNR estimate of (Chapter 3 reader will recall $\text{SNR}_n + 1$). The ZF algorithm is preferred because there is no issue with noise enhancement with per-subchannel normalization. The factor μ is a positive gain constant for the adaptive algorithm selected to balance time-variation versus noise averaging.

The channel gain is estimated by

$$\hat{H}_n = \frac{d}{d_n \cdot W_n} \quad . \tag{4.137}$$

The mean square noise, assuming "ergodicity" at least over a reasonable time period, is estimated by time-averaging the quantity $|\tilde{U}_n|^2$,

$$\tilde{\sigma}^2_{n,k+1} = (1-\mu')\tilde{\sigma}^2_{n,k} + \mu' |\tilde{U}_n|^2 \quad , \tag{4.138}$$

where μ' is another averaging constant and

$$0 \leq \mu' < 2 \quad . \tag{4.139}$$

Since this noise was scaled by W_n also, then the estimate of g_n is then

$$\hat{g}_n = \frac{|H_n|^2}{\frac{\tilde{\sigma_n}^2}{|W_n|^2}} = \frac{d^2}{d_n^2} \cdot \frac{1}{\tilde{\sigma_n}^2} \quad . \tag{4.140}$$

Since the factor d^2/d_n^2 is constant and the loading algorithm need only know the relative change in incremental energy, that is the incremental energy table is scaled by

$$\frac{\hat{g}_{n,old}}{\hat{g}_{n,new}} = \frac{\tilde{\sigma}^2_{n,new}}{\tilde{\sigma}^2_{n,old}} \quad . \tag{4.141}$$

Thus the new incremental-energy entries can be estimated by simply scaling up or down the old values by the relative increase in the normalizer noise. Often the old value is an initial value stored at first initialization of the the transmission system. This reference to initial value then avoids a cascade of "finite-precision" errors.

4.3.6 ("bit-swapping")

If the loading algorithm moves or "swaps" a bit (or more generally information unit) from one tone to another, then the multi-channel transmitter and receiver should implement this "bit-swap" on the same symbol, as in the last subsection. A bit-swap is usually implemented by the receiver sending a pair of tone indices n (tone where bit is deleted) and m (tone where bit is added) through some reverse channel to the transmitter, as in Figure 4.12. There is often also an associated change in transmitted energy on each tone:

$$G_i = \frac{\mathcal{E}_i(new)}{\mathcal{E}_i(old)} \text{ for } i = m, n \quad . \tag{4.142}$$

[17] A convergence proof notes that with stationarity $E[W_n] = E[W_n] + \mu E\left[\left(X_n - W_{n,k}Y_{n,k}\right) \cdot X_n^*\right] = \left(1 - \mu H_n \mathcal{E}_\backslash\right) E[W_n] + \mu \mathcal{E}_n$, A simple first-order recursion in average FEQ value. This will converge to $E[W_n] = 1/H_n$ when the geometric ratio is less than 1.

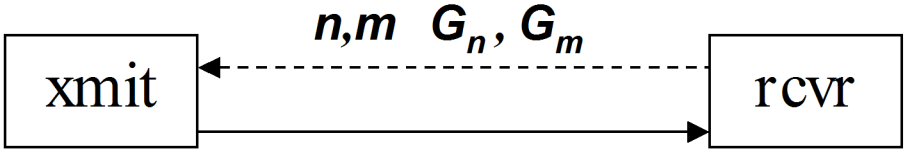


Figure 4.12: Illustration of reverse channel in bit-swapping.

The energy for the bit-swap tones $i = m$ or $i = n$ can be recomputed by summing the new increments in the table,

$$\mathcal{E}_i(new) = \frac{\hat{g}_{i,old}}{\hat{g}_{i,new}} \sum_{j=1}^{b_i} e_{i,old}(j) \quad . \tag{4.143}$$

Typically this factor is near unit value and usually less than a 3 dB change for 2-dimensional subchannels like those of MT (and less than 6 dB change for one-dimensional subchannels). The near-unity-value occurs because tone n's factor of decrease/increase in channel SNR $\frac{\hat{g}_{n,old}}{\hat{g}_{n,new}}$ is typically about 2 in order to cause the swap to be attractive, and this factor of 2 is then nearly the reciprocal of the constellation-distance increase/decrease when halving the constellation size on that tone n. However, it could be that the gain is not a factor of 2. Thus, a gain factor G_i should also be conveyed through the reverse channel to the transmitter. Some implementations thus ignore this factor if it is always close to one. This gain is typically relative to the initial value used by the transmitter during training (and not relative to the last value used, because such relative gain could lead to a chain of cascading finite-precision errors.

For RA or MA operation, the two gain factors should be selected to produce a total energy of the transmitter that is exactly equal to the allowed energy limit. Also, all the multi-channel normalizer coefficients need to be reduced by the inverse square-root of this factor, as follows: Equation (4.137) can be rewritten as

$$W_n = \frac{d}{d_n \cdot \hat{H}_n} \quad . \tag{4.144}$$

The distance d is constant in the receiver, but d_n has possibly chanbed. ($\hat{H}_n$ may have changed, but that change was already incorporated in the adaption of W_n that lead to a swap, so it is effectively viewed as constant.) The change in distance should be roughly a factor of 2, but is exactly known to the loading algorithm control as say

$$\Delta d_n = \frac{d_{n,new}}{d_{n,old}} \quad , \tag{4.145}$$

as is the gain factor G_n. When a swap is executed, the values for W_m and W_n and the corresponding noise estimates need scaling by an additional factor: The new values for W_n and $\tilde{\sigma}_n^2$ are

$$W_n(\text{new}) \rightarrow W_n(\text{old}) \cdot \frac{1}{\Delta d_n \cdot \sqrt{G_n}} \quad , \quad \tilde{\sigma}_n^2(\text{new}) \rightarrow \tilde{\sigma}_n^2(\text{old}) \cdot \frac{1}{(\Delta d_n)^2 \cdot G_n} \quad , \tag{4.146}$$

and similarly for tone index m. One channel change may produce several bit swaps.

The accuracy of channel estimation is important and discussed in Section 4.5. This section assumed all normalizer quantities have no error.

4.3.7 Gain Swapping

In addition to any energy changes associated with a bit-swap, gain may be changed in the absence of bit movement and is known as "gain swapping." While MA bit-swapping will continue to move bits from channels of higher energy cost to those of lower energy cost, there is still a barrier to moving a bit that is the difference between the cost to add a bit in the lowest cost position of the incremental-energy table relative to the savings of deleting that same bit on the highest-cost position in that same table. In the worst case, this cost can be equal to the cost of a unit of information. On a two-dimensional subchannel

with $\beta = 1$, this cost could be 3 dB of energy for a particular subchannel. This means that a probability of error of 10^{-6} on a particular subchannel could degrade by as much as roughly 3 orders of magnitude before the swap occurred. A particularly problematic situation would be a loss of for instance 2.9 dB that never grew larger and the swap never occurred. With a number of subchannels greater than 1000, the effect would be negligible, but as N decreases, the performance of a single subchannel can more dramatically effect the overall probability of error.

There are two solutions to this potential problem. The first is to reduce β, but this may be difficult. The second is alter the transmit energy to increase transmit energy on one subchannel and place it on another subchannel. Furthermore, the presence of a dramatically increased or increase noise (for instance a noise caused by another system turning on or off) might lead to a completely different distribution of energy when loading. Thus, the energy transmitted on a subchannel (or equivalently the relative factor applied to such a subchannel) may need to change significantly. Gain swapping (sending of the factor of increase/decrease of energy on a subchannel is thus commonly used. Usually by convention, a value of $b_n = 0$ on a subchannel means either no energy is to be transmitted or a very small value instead of the nominal level. If a gain-swap occurs, then the two columns of the incremental energy table needed to be scaled by the respective factors – and indeed a bit-swap may subsequently occur because of this change. One channel change may produce many gain swaps.

4.4 Channel Partitioning

Channel Partitioning methods divide a transmission channel into a set of parallel, and ideally independent, subchannels to which the loading algorithms of Section 4.3 can be applied. Multi-tone transmission (MT) is an example of channel partitioning where each subchannel has a frequency index and all subchannels are independent, as in Section 4.1 and Figure 4.2. The MT basis functions $\{\varphi_n(t)\}$ form an orthonormal basis while also exhibiting the desirable property that the set of channel-output functions $\{h(t) * \varphi_n(t)\}$ remains orthogonal for any $h(t)$. Reference to Chapter 8 finds that the MT basis functions, when combined with continuous-time water-filling, and $N \to \infty$ are optimum. For a finite N, the MT basis functions are not quite optimum basically because the channel shaping induced by $h(t)$ in each MT frequency band is not quite flat in general, and some equalization loss occurs on each tone. Furthermore, the MT basis functions exist over an infinite time interval, whereas practical implementation suggests restriction of attention to a finite observation interval. Thus, MT makes a useful example for the introduction of multi-channel modulation, but is not practical for channel partitioning.

This section introduces optimum basis functions for modulation based on a finite time interval of T of channel inputs and outputs. Subsections 4.4.1 and 4.4.2 show that the optimum basis functions become channel-dependent and are the eigenfunctions of a certain channel covariance function. There are two ways to accommodate intersymbol interference caused by a channel impulse response. Theoretically, Subsection 4.4.2 investigates a conceptual method for computing best possibilities. Subsection 4.4.4 instead investigates the elimination of the interference between subchannels with the often-encountered use of "guard periods".

4.4.1 Eigenfunction Transmission

A one-shot multi-channel modulated signal $x(t)$ satisfies

$$x(t) = \sum_{n=1}^{N} x_n \cdot \varphi_n(t) \quad , \tag{4.147}$$

while a succession of such transmissions satisfies

$$x(t) = \sum_{k} \sum_{n=1}^{N} x_{n,k} \cdot \varphi_n(t - kT) \quad . \tag{4.148}$$

The convolution of a channel response $h(t)$ with $x(t)$ produces

$$h(t) * x(t) = \sum_{k} \sum_{n=1}^{N} x_{n,k} \cdot \varphi_n(t - kT) * h(t) = \sum_{k} \sum_{n=1}^{N} x_{n,k} \cdot p_n(t - kT) \quad , \tag{4.149}$$

which is characterized by N pulse responses $p_n(t)$, $n = 1, ..., N$. Following a development in Section 3.1, the set of sampled outputs of the N matched filters $p^*(-t)$ at all symbol instants kT forms a sufficient statistic or complete representation of the transmitted signal as shown in Figure 4.13. There are N^2 channel-ISI-describing functions

$$q_{n,m}(t) \triangleq \frac{p_m(t) * p_n^*(-t)}{\|p_m\| \cdot \|p_n\|} \tag{4.150}$$

that characterize the $N \times N$ square matrix $\boldsymbol{Q}(t)$. The quantity $q_{n,m}(t)$ is the matched-filtered channel from subchannel m input (column index) to subchannel n output (row index). The system is now a multiple-input/multiple-output transmission system, sometimes called a MIMO, with sampled-time representation

$$\boldsymbol{Q}_k = \boldsymbol{Q}(kT) \quad . \tag{4.151}$$

Generalization of the Nyquist Criterion

Theorem 4.4.1 (Generalized Nyquist Criterion (GNC)) *A generalization of the Nyquist Criterion of Chapter 3 for no ISI nor intra-symbol interference (now both are called ISI together) is*

$$\boldsymbol{Q}_k = I\delta_k \quad , \tag{4.152}$$

that is the sampled MIMO channel matrix is the identity matrix.

proof: Since $\boldsymbol{Q}_k = I\delta_k = 0$ if $k \neq 0$, there is no interference from other symbols, and since $q_{n,m}(t) = \delta_{n,m} = 0$ if $n \neq m$ there is also no interference between subchannels. The unit value follows from the normalization of each basis function, $q_{n,n}(0) = 1$. **QED**

The earlier Nyquist Theorem of Chapter 3 is a special case that occurs when $\boldsymbol{Q}_k = q_k = \delta_k$. Thus, to avoid ISI, the designer could choose basis functions so that (4.152) holds. In general, there are many choices for such functions. For a sub-optimum set of basis functions that do not satisfy the Generalized Nyquist Criterion, vector equalization may be used as discussed in Section 5.7. This section and Chapter deal only with those that do satisfy the GNC or the GNC's discrete-time equivalents.

4.4.2 Choice of transmit basis

The channel impulse response forms a **channel autocorrelation function** according to

$$r(t) = h(t) * h^*(-t) \quad , \tag{4.153}$$

which is presumed to be nonzero only over a finite interval $(-T_H/2, T_H/2)$. This channel autocorrelation function defines a possibly countably infinite-size set of orthonormal **eigenfunctions**, $\{\varphi_n(t)\}$, nonzero only over the time interval $[-(T-T_H)/2, (T-T_H)/2)$, that satisfy the relation[18]

$$\rho_n \cdot \varphi_n(t) = \int_{-T/2}^{T/2} r(t-\tau) \cdot \varphi_n(\tau) d\tau \;\; n = 1, ..., \infty \;\; , \;\; \forall t \in [-(T-T_H)/2, (T-T_H)/2) \; , \tag{4.154}$$

that is a function $\varphi_n(t)$ that when convolved with the matched-filtered channel over the interval $[-T/2, T/2)$ reproduces itself, scaled by a constant ρ_n, called the **eigenvalue**. The set of eigenvalues is unique to the channel autocorrelation $r(t)$, while the eigenfunctions are not necessarily unique. [19] Each eigenfunction represents a "mode" of the channel through which data may be transmitted independently of the other modes with gain ρ_n. The restriction of the time interval to be equal to the symbol period allows intersymbol interference to be zero without further consideration. The restriction is not absolutely necessary, but then requires the additional constraint that successive translations of the basis functions at the symbol rate lead to a Q_k that satisfies the generalized Nyquist criterion. The eigenvalue problem is very difficult to solve when $r(kT) \neq \delta_k$. A number of mathematicians have addressed the problem when $r(kT) = \delta_k$, and the theory is known as "wavelets," "filter banks," "polyphase filters," or "reconstruction filters." However, this abstract theory is difficult to apply when there is ISI. Subsection 4.4.5 briefly discusses this concept, but a simpler concept known as a cyclic extension will be shown to circumvent the difficulty with neglible loss. Section 5.7 solves the problem with inter-block-symbol overlap directly with multidimensional block equalizers.

There may be a countably infinite number of eigenvalues and corresponding eigenfunctions, so the N largest are selected. There may be other sets of functions that could be used for transmit basis functions, but none could pass more information. To see this, one could suppose the existence of another distinct set of functions $\phi_n(t) for n = 1, 2, ...N$. These functions would have a linear orthonormal representation (think Gram Schmidt) on the N largest eigenfunctions and a component on the remaining eigenfunctions

[18] Most developments of eigenfunctions define the functions over the causal time interval $[0, T - T_H)$ - we instead noncausaly center this interval about the origin at $[-(T-T_H)/2, (T-T_H)/2)$ in this theoretically abstract case so that taking the limit as $T \to \infty$ produces a true two-sided convolution integral, which will then have eigenvalues as the Fourier transform.

[19] The eigenfunctions are generally difficult to compute (unless the symbol period is infinite) and used here for theoretical purposes. An approximation method for their computation is listed at the end of this subsection.

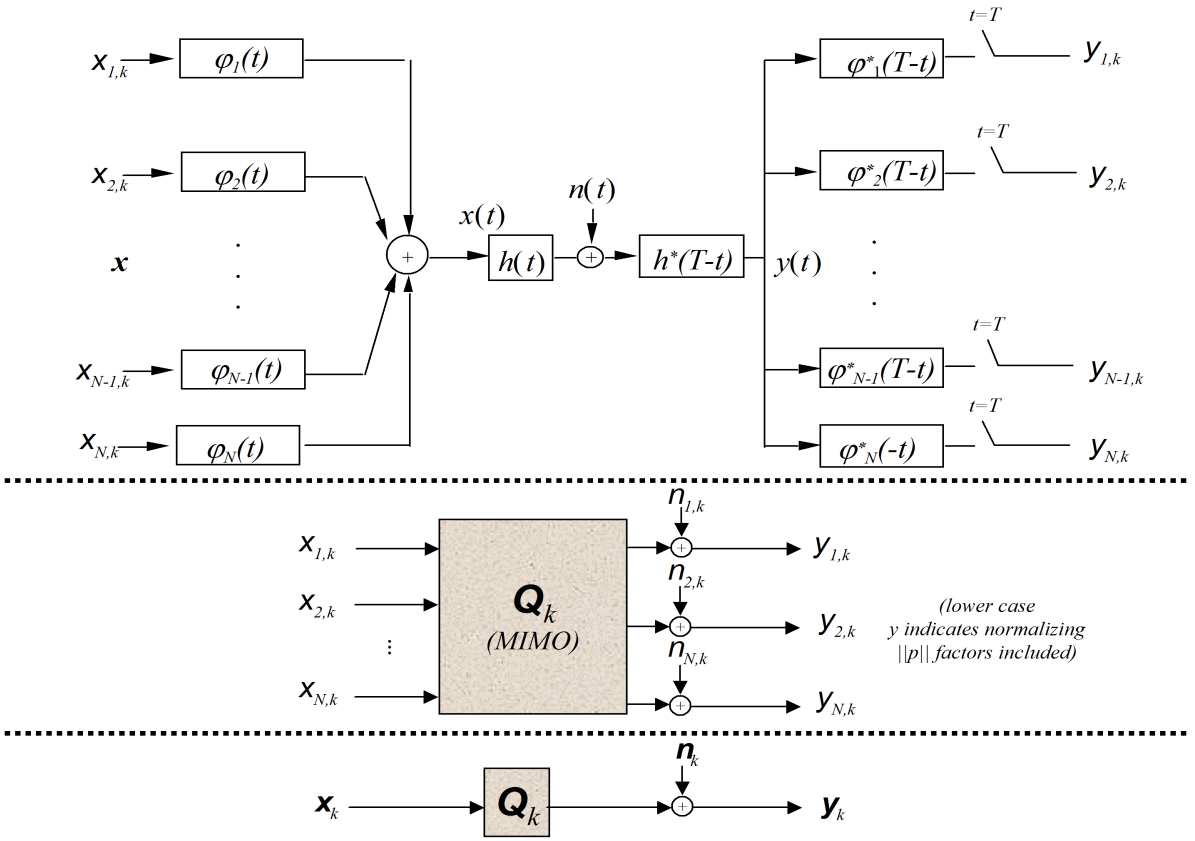


Figure 4.13: Modal Modulation: optimum channel partitioning.

(since the eigenfunctions are complete). Any energy on the remaining eigenfunctions necessarily would carry less energy at the channel output. Thus, when processed with a receiver matched filter $h^*(-t)$ that is known to be sufficient, the alternative distinct set would lose information, unless they have zero component on the smaller eigenfunctions. Thus, the largest set of N, or any set in 1-to-1 correspondence with them, are the best to use for transmission.

Modal Modulation (MM) in Figure 4.13 uses the channel-dependent orthonormal set of eigenfunctions (the N with largest eigenvalues) as a basis for modulation,

$$x(t) = \sum_{n=1}^{N} x_n \cdot \varphi_n(t) \quad . \tag{4.155}$$

The receiver signal $y(t)$ in Figure 4.13 is then

$$y(t) = \sum_{n=1}^{N} x_n \cdot [r(t) * \varphi_n(t)] + \tilde{n}(t) \tag{4.156}$$

where $\tilde{n}(t)$ is the matched-filtered noise with autocorrelation function $\frac{\mathcal{N}_0}{2} \cdot r(t)$. There is no information loss in this signal because it is the output of the matched filters for each of the transmit bases. Through Equation (4.154), this signal $y(t)$ is rewritten

$$y(t) = \sum_{n=1}^{N} (\rho_n x_n) \cdot \varphi_n(t) \quad , \tag{4.157}$$

which is a sum of weighted channel input samples, $\rho_n \cdot x_n$, times the orthonormal basis functions. This summation has the same form as the modulated waveforms studied in Chapter 1. Each symbol element x_n is replaced by $\rho_n x_n$.

The MAP/ML receiver is thus also the same as in Chapter 1 and recovers $\rho_n x_n$ by processing each filter output by a matched-filter/sampler for each of the basis functions, as shown in Figure 4.152. The noise samples at the output of these matched filters are independent for different n because the basis functions are orthogonal, or equivalently,

$$E[\tilde{n}_i \tilde{n}_j^*] = \frac{\mathcal{N}_0}{2} \int_{-T/2}^{T/2} \int_{-T/2}^{T/2} r(t-s)\varphi_i(t)\varphi_j^*(s)dtds \tag{4.158}$$

$$= \frac{\mathcal{N}_0}{2} \int_{-T/2}^{T/2} \rho_i \varphi_i(t)\varphi_j^*(t)dt \tag{4.159}$$

$$= \rho_i \frac{\mathcal{N}_0}{2} \delta_{ij} \quad . \tag{4.160}$$

Thus, the joint-probability density function $\mathrm{p}_{\boldsymbol{y}/\boldsymbol{x}}$ factors into N independent component distributions, and an ML/MAP detector is equivalent to an independent detector on each sample output. Thus, the MM transmission system generates a set of parallel channels to which the results of Sections 4.2 and 4.3 can then be applied. The SNR's are $\mathrm{SNR}_n = (\rho_n^2 \cdot \mathcal{E}_n)/(\rho_n \cdot \frac{\mathcal{N}_0}{2}) = \rho_n \cdot \mathcal{E}_n / \frac{\mathcal{N}_0}{2}$.

The functions $p_n(t)$ defined by modal modulation clearly satisfy the generalized Nyquist criterion because of the independence of the subchannels created and because they are zero outside the interval $[-T/2, T/2)$, thus averting intersymbol interference.

Theorem 4.4.2 (Optimality of Modal Modulation) *Given a linear additive Gaussian noise channel with ISI, modal modulation produces the largest $SNR_{m,u}$ of any N-dimensional set of orthogonal basis functions exist only on the interval $t \in [-(T-T_H)/2, (T-T_H)/2)$.*

Proof: The MM transmission system of Figure 4.152 can be optimally detected by observing each of the independent subchannel outputs individually. Its performance thus can not be improved. Any set of functions nonzero over the interval $[-(T-T_H)/2, (T-T_H)/2)$ that satisfies the generalized Nyquist criterion are necessarily the eigenfunctions. The N largest eigenvalues are unique and MM selects teh eigenfunctions corresponding to the N largest. A well-known property of the eigenvalues is that no other set of N has a larger product. Then,

$$\prod_{n=1}^{N} g_n \tag{4.161}$$

is a maximum. For any given set of energies $\{\mathcal{E}_n\}$, the product

$$\prod_{n=1}^{N} \left(1 + \frac{\mathcal{E}_n \cdot g_n}{\Gamma}\right) \tag{4.162}$$

is also a maximum because the function $1 + \frac{\mathcal{E}_n \cdot g_n}{\Gamma}$ is linear (convex) in g_n. Then since this function is maximized for any set of energies, it is certainly the maximum for the water-filling set of energies that are known to maximize for any set of $\{g_n\}$. Thus, the equivalent-channel SNR is the highest for any gap, and modal modulation achieves the highest possible SNR of any partitioning scheme on the interval $[-(T-T_H)/2, (T-T_H)/2)$. As $T \to \infty$ for fixed T_H, then the length of the autocorrelation function for any practical and thus finite-length channel becomes negligible, then modal modulation is also optimum over the entire time axis without restriction on the time interval to be shorter than the symbol period. **QED.**

The above theorem essentially states that MM is optimum given observation of a finite time interval $[-(T-T_H)/2, (T-T_H)/2)$. It is then natural to expect the following lemma that states that MM converges to MT modulation in performance as both allow the number of dimensions N to go to infinity while also allowing the time interval to span $(-\infty, \infty)$:

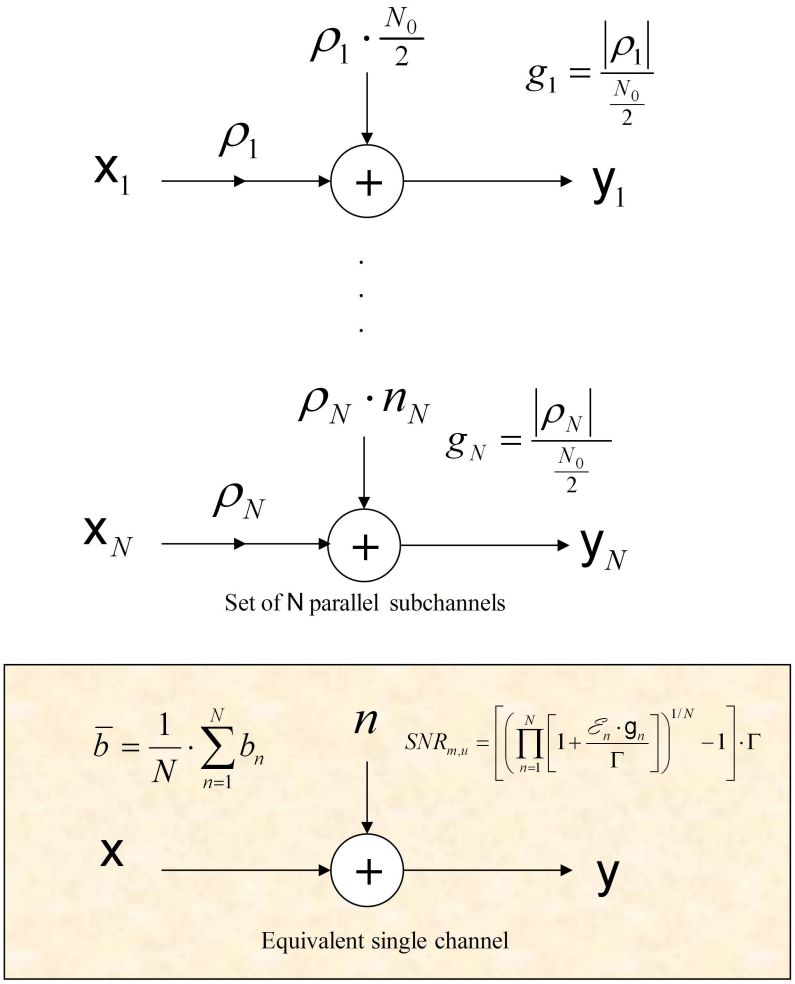


Figure 4.14: Parallel Subchannels of Modal Modulation.

Theorem 4.4.3 (Convergence of MT to Optimum) *Multitone Modulation converges to Modal Modulation as both* $N \to \infty$ *and* $[-T/2, T/2) \to (-\infty, \infty)$.

Proof: The set of eigenvalues of any autocorrelation function is unique. The set of eigenvalues essentially determine the performance of MM through the $\text{SNR}_{m,u}$ for that channel. By simple substitution of $\varphi_n(t) = e^{-\jmath\frac{2\pi n}{T}t}$ into the defining equation of eigenvalues shows that these exponential functions are a valid choice of eigenfunctions as the symbol period becomes infinite, $[-T/2, T/2) \to (-\infty, \infty)$, and the corresponding eigenvalues are the values of $R(\omega)$ at frequencies of the exponentials. Thus, the eigenvalues are the values of the $R(\omega)$ at each frequency. As $N \to \infty$, these frequencies become infinitely dense, and equal to those of the MT system, which then has no ISI on any tone. The corresponding MT receiver is optimum (ML/MAP). Thus, both systems have the same $\text{SNR}_{m,u}$ and both have optimum receivers, and by Theorem 4.4.2. The multitone system is thus optimum for infinite-length symbol period and an infinite number of tones. **QED.**

The implication of the above lemma is that for sufficiently large N and T, the designer can use either the MM method or the MT method and rest assured the design is as good as can be achieved on the linear ISI channel with Gaussian noise.

4.4.3 Limiting Results

The spectrum used by the basis functions of modal modulation thus converges also to the set of used tones in the infinite-dimensional MT system. The interval may be a set of continuous frequencies or several such sets and is denoted by Ω. The measure of Ω is the total bandwidth used

$$|\Omega| = \int_{\Omega} d\omega \quad . \tag{4.163}$$

An optimum bandwidth Ω^* will then be that corresponding to the subchannels used in water-filling as $N \to \infty$. If N^* such subchannels are used in MT, then

$$\lim_{N\to\infty} \frac{N^*}{N} \cdot 2\pi = |\Omega^*| \quad . \tag{4.164}$$

The data rate is

$$\lim_{T\to\infty} \frac{b}{T} \quad . \tag{4.165}$$

Continuous frequency can then replace the frequency index n according to

$$\omega = 2\pi \cdot \lim_{N\to\infty} \frac{n}{NT'} \; , \quad n = -N/2, ..., N/2 \quad , \tag{4.166}$$

and the width of a tone becomes

$$d\omega = \lim_{N\to\infty} \frac{2\pi}{NT'} \tag{4.167}$$

If $1/T'$ is sufficiently large to be at least twice the highest frequency that could be conceived of for use on any given band-limited channel, then Ω^* becomes the true optimum band for use on the continuous channel. The two-sided power spectral density at frequency ω corresponding to $\frac{n}{NT}$ then is

$$S_x(\omega) = \lim_{N\to\infty} \bar{\mathcal{E}}_n \quad . \tag{4.168}$$

For infinite time and frequency as used here, there is no need for complex baseband equivalents and thus all dimensions are considered real. The power then becomes

$$P_x = \lim_{N\to\infty} \frac{1}{N \cdot T'} \cdot \sum_{n=-N/2}^{N/2} \bar{\mathcal{E}}_n = \frac{1}{2\pi} \cdot \int_{\Omega} S_x(\omega) d\omega \quad . \tag{4.169}$$

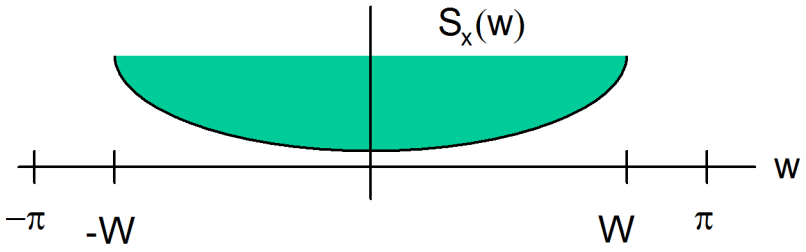


Figure 4.15: Capacity frequency interval for $1 + .9D^{-1}$ channel.

The water-filling equation then has continuous equivalent

$$\bar{\mathcal{E}}_n + \frac{\Gamma}{g_n} \rightarrow S_x(\omega) + \frac{\Gamma}{g(\omega)} = \lambda \text{ a constant} \tag{4.170}$$

subject to the total power constraint in (4.169) , which has corresponding solution for $S_x(\omega > 0$ for all $\omega \in \Omega^*$ and $S_x(\omega) = 0$ at all other frequencies. The data rate then becomes

$$R = \frac{1}{2\pi} \int_{\Omega^*} \frac{1}{2} \log_2 \left(1 + \frac{S_x(\omega) \cdot g(\omega)}{\Gamma} \right) d\omega \tag{4.171}$$

$$= \frac{1}{2\pi} \int_{\Omega^*} \frac{1}{2} \log_2 \left(\frac{\lambda \cdot g(\omega)}{\Gamma} \right) d\omega \tag{4.172}$$

EXAMPLE 4.4.1 ($1 + .9D$ channel) The channel with impulse response $h(t) = \text{sinc}(t) + .9\text{sinc}(t-1)$ has the same performance as the $1 + .9D^{-1}$ channel studied throughout this book, if the transmit filter is $\frac{1}{\sqrt{T}}\text{sinc}(t/T)$. An system with optimized MT basis functions of infinite length would have an optimum bandwidth $\Omega^* = [-W, W]$ for some $W \leq \pi/T$ as in Figure 4.15 Then, continuous water-filling with $\Gamma = 1$ produces

$$P_x = \int_{-W}^{W} \left(\lambda' - \frac{.181}{1.81 + 1.8\cos(\omega)} \right) \frac{d\omega}{2\pi} \tag{4.173}$$

where W is implicitly in radians/second for this example. If $P_x = 1$ with $\frac{\mathcal{N}_0}{2} = .181$, the integral in (4.173) simplifies to

$$\pi = \int_{0}^{W} \left(\lambda' - \frac{.181}{1.81 + 1.8\cos(\omega)} \right) d\omega \tag{4.174}$$

$$= \lambda' W - .181 \left\{ \frac{2}{\sqrt{1.81^2 - 1.8^2}} \arctan \left[\frac{\sqrt{1.81^2 - 1.8^2}}{1.81 + 1.8} \tan\left(\frac{W}{2} \right) \right] \right\} \tag{4.175}$$

At the bandedge W,

$$\lambda' = \frac{.181}{1.81 + 1.8\cos(W)} \quad . \tag{4.176}$$

leaving the following transcendental equation to solve by trial and error:

$$\pi = \frac{.181W}{1.81 + 1.8\cos(W)} - 1.9053 \arctan\left(.0526 \tan(W/2) \right) \tag{4.177}$$

$W = .88\pi$ approximately solves (4.177), and the corresponding value of λ is $\lambda = 1.33$.

The highest data rate with $1/T = 1$ is then

$$\begin{aligned}
\mathcal{C} &= \frac{2}{2\pi}\int_0^{.88\pi}\frac{1}{2}\log_2\left(\frac{1.33}{.181}(1.81+1.8\cos\omega)\right)d\omega && (4.178)\\
&= \frac{1}{2\pi}\int_0^{.88\pi}\log_2 7.35 d\omega + \frac{1}{2\pi}\int_0^{.88\pi}\log_2\left(1.81+1.8\cos\omega\right)d\omega && (4.179)\\
&= 1.266 + .284 && (4.180)\\
&\approx 1.55\text{bits/second} \quad . && (4.181)
\end{aligned}$$

This exceeds the 1 bit/T transmitted on this channel in the examples of Chapter 3. Later sections of this chapter show that the $\text{SNR}_{m,u}$ for this channel becomes 8.8 dB for large N, .4 dB better than the best MMSE-DFE, and actually 1.7 dB better than the precoded MMSE-DFE. The MT system has no error propagation, and is also an ML detector.

In computing data rates as in water-filling with $\Gamma > 0$ dB, the designer needs to remember that the gap concept is only accurate when $\bar{b} \geq 1$. Often water-filling may include regions of bandwidth for which $\bar{b}$ in at least those bands is less than 1. The gap approximation becomes increasingly accurate as the gap gets smaller, and indeed is accurate at all $\bar{b}$ when the gap is 0 dB, meaning capacity results are all exact.

Periodic Channels

A channel with periodic $r(t) = r(t+T)$ will always have eigenfunctions $\varphi_n(t) = e^{\jmath\frac{2\pi}{T}nt}$ for the interval $[-T/2, T/2)$. Periodic channels do not exist in practice, but designers often use extra bandwidth in the design of a transmission problem to make the finite-duration channel appear as if it were periodic. In this case, MT and MM would again be the same.

Computing the Eigenfunctions

The eigenfunctions can be difficult to compute in closed form for most channels. At a sampling rate $1/T'$ sufficiently high to capture any significant channel frequencies, the eigenfunction equation can be sampled over the time interval $-(T-T_H)/2 = -LT'$ to $(T-T_H)/2 = LT'$ to yield

$$\rho_n \cdot \begin{bmatrix} \varphi_n(-LT') \\ \varphi_n(-LT'+T') \\ \vdots \\ \varphi_n(LT') \end{bmatrix} = T' \cdot \begin{bmatrix} r(0) & \ldots & r(-2LT') \\ r(T') & \ldots & r(-2LT'+T') \\ \vdots & \ddots & \vdots \\ r(2LT') & \ldots & r(0) \end{bmatrix} \begin{bmatrix} \varphi_n(-LT') \\ \varphi_n(-LT'+T') \\ \vdots \\ \varphi_n(LT') \end{bmatrix} \quad (4.182)$$

$$\rho_n \cdot \boldsymbol{\varphi}_n = R\boldsymbol{\varphi}_n \quad . \quad (4.183)$$

This equation is equivalent to finding the eigenvectors and eigenvalues of the matrix R. The largest N eigenvalue magnitudes $|\rho_n|$ select the corresponding eigenvectors as the sampled basis functions. These basis functions may be interpolated to create eigenfunction approximations. Clearly, the smaller the T', the more accurate the approximation.

In implementation, Section 4.5 illustrates a yet better way to approximate eigenfunctions in discrete time, which is known as vector coding.

4.4.4 Time-Domain Packets and Partitioning

Overlap, Excess Bandwidth, and the Guard Period

A realistic transmission channel has most or all its nonzero impulse response confined to a finite time interval T_H. Thus, convolution of the (necessarily causal) channel impulse response with nonzero input basis functions existing over $[0, T)$ produces an output with duration longer than the symbol period

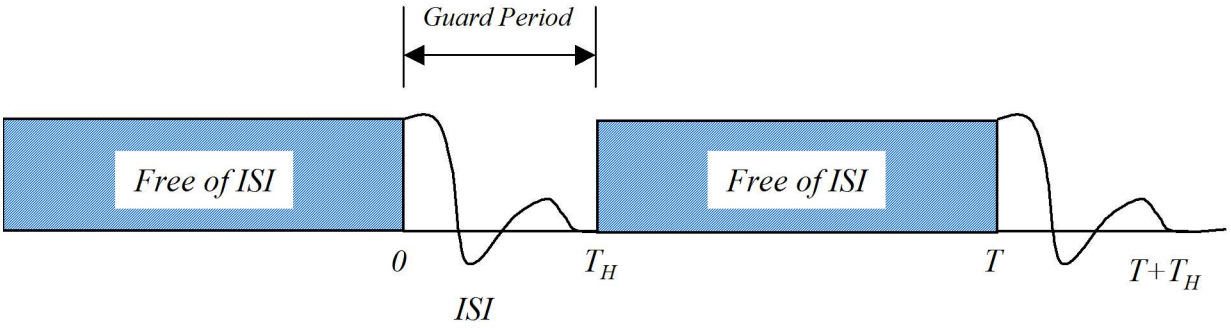


Figure 4.16: Illustration of guard period.

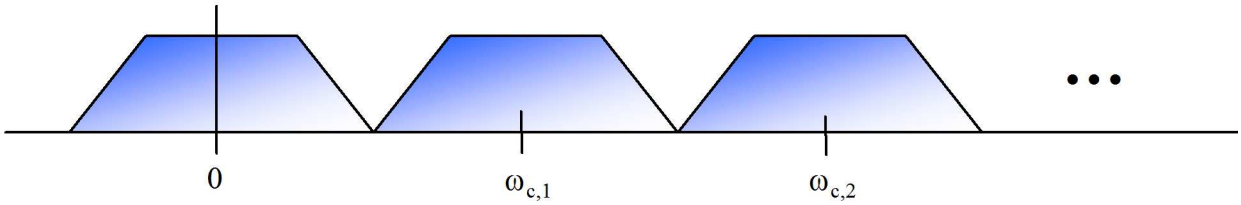


Figure 4.17: Filters with multitone.

T[20] as in Figure 4.16. The first T_H seconds of any symbol are thus possibly corrupted by intersymbol interference from the last symbol if that previous symbol had nonzero transmit energy over $(T-T_H, T)$. Thus, the receiver then usually ignores the first T_H seconds of any symbol. If $T_H << T$, then this overhead penalty, or "excess bandwidth" penalty, is small and no equalization need be used. For a fixed channel and thus fixed T_H, as $T \to \infty$, this excess bandwidth becomes negligible. This time period where the receiver ignores channel outputs is often called a **guard period** in multichannel modulation. For finite symbol period, the multichannel system then has an excess bandwidth given by the factor $\alpha = T_H/(T-T_H)$.

4.4.5 Filters with Multitone

Multitone partitioning almost satisfies the Generalized Nyquist criterion with any impulse response $h(t)$, but requires an infinite-length basis function $\varphi(t) = \frac{1}{\sqrt{T}}\text{sinc}\left(\frac{t}{T}\right)$ and is not physically realizable even when N is finite. The orthogonality of subchannels is ensured because the Fourier transforms of the basis function translates by $f = \frac{n}{T}$ do not overlap and thus are trivially orthogonal at the linear time-invariant channel output. ISI is nearly averted by the narrow bands that are nearly flat even at channel output. Thus, many designers have investigated the possibility of approximating the multitone functions with physically realizable basis functions, as generally depicted in Figure 4.17. There is no guard band in the time domain, but instead an excess bandwidth in the frequency domain.

The basic idea is to avoid the use of a T_H-second guard period in the time domain by instead having very sharp basis functions that approximate $\varphi(t) = \frac{1}{\sqrt{T}}\text{sinc}\left(\frac{t}{T}\right)$. One method might be to simply truncate this basis function to say $\pm$ 100 symbol periods, thus introducing a delay of approximately $101T$ in realization (and introducing a high complexity), and of course necessarily another $101T$ in the receiver. Contrast this with a delay of just T with the use of the guard period and one sees the method

[20] This section will now begin to pursue realizable transmission systems and not theoretical abstractions, thus the basis functions will necessarily have to be causal in the remainder of this section.

is not attractive. However, shorter length approximations may suffice. One method is to introduce again the raised cosine functions of Chapter 3 with excess bandwidth of 10 to 20%. Sometimes this level of excess bandwidth can be less for certain types of channels (or noise-equivalent channels) that have sharp band-edges, notches, or narrow-band noises, all of which result in long impulse responses and thus large T_h in the guard band of Section 4.4.4. Other functions may also achieve the same time of "low side bands" for the Fourier transform of the basis function. Of course, any such realization requires in reality some type of over-sampled digital signal processing. Section 4.6, Subsection 4.6.8 studies discrete realizations of a few methods that have been proposed to date.

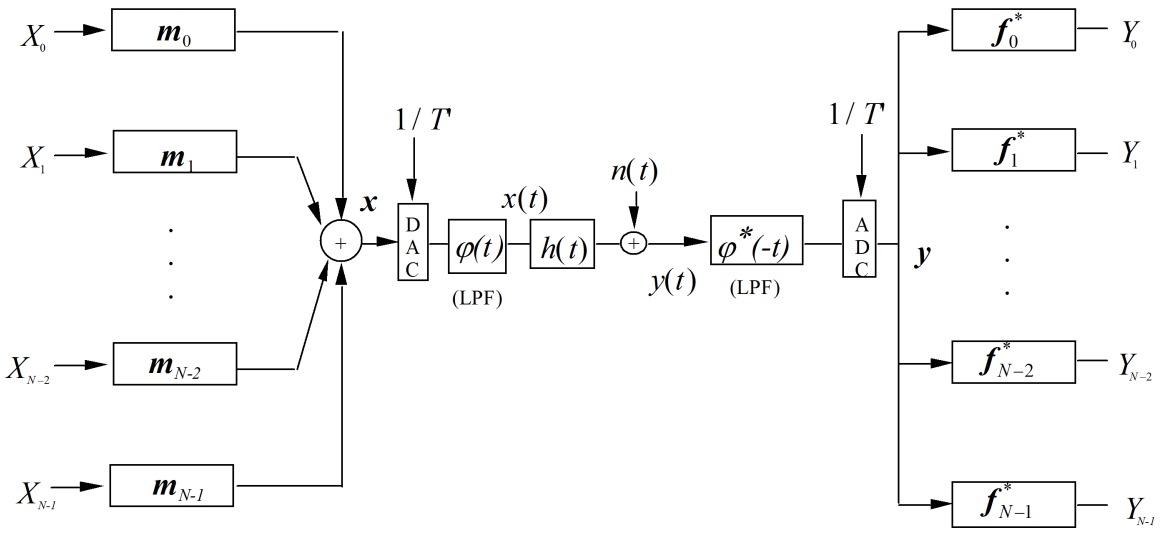


Figure 4.18: Digital realization of channel partitioning.

4.5 Discrete-time channel partitioning

While MM restricts attention to a finite time interval for construction of the transmit basis functions, these functions remain continuous in time and a function of the channel. Such functions would be difficult to implement exactly in practice. **Discrete-time channel partitioning** partitions a discrete-time description of the channel. This description is a linear matrix relationship between a finite number of output samples (presumably sampled at a rate $1/T'$ exceeding twice the highest frequency to be transmitted) and a corresponding finite set of channel input samples at the same rate that constitute the transmitted symbol. Such a description can never be exact in practice, but with a sufficient number of input and output samples included, this description can be very close to the exact action of the ISI channel. Figure 4.18 illustrates general N-dimensional discrete-time channel partitioning. Discrete-time **basis vectors**, $\boldsymbol{m}_n$ replace the continuous basis functions of MM or MT. These basis vectors have a finite length of $N + \nu$ samples over a duration of time T. The sampled pulse response, $p(kT')$, of the channel is assumed to span a time interval of less than $\nu + 1$ sample periods, where $(\nu + 1)T' = T_H$. Each basis vector, $\boldsymbol{m}_n$, multiplies a sub-symbol element X_n before being summed to form the transmit symbol vector $\boldsymbol{x}$.[21] Then, $\boldsymbol{x}$ is output through a digital-to-analog converter. Some transmit filtering of the output DAC occurs (typically analog lowpass anti-aliasing) before the signals pass through the ISI channel. The combined post-DAC filtering, channel impulse response, and pre-ADC filtering in the receiver form a pulse response $p(t)$. The sampling rate $1/T' = (N+\nu)/T$ is higher than twice the highest frequency that the designer hopes to use for the transmitted signals. The modulator attempts to use basis vectors, $\boldsymbol{m}_n$, that will remain orthogonal after undergoing the dispersive effect of the ISI channel, just as orthogonality was maintained with MM.

The receiver low-pass filters with gain $1/\sqrt{T'}$ (that is also included in the pulse response) to maintain noise per dimension at $\frac{N_0}{2}$), and then samples at rate $1/T'$ the received modulated signal. The matched filters are discrete-time and also finite length. The matched filters are only complex if the entire set of modulated signals had been passband modulated by an external passband modulator that is not shown in Figure 4.18, and would in this case correspond to a complex baseband-equivalent channel. There are $N + \nu$ input samples that lead to N matched-filter output samples, with ν samples lost to the guard period to avoid intersymbol interference. The samples can be real or complex, which in the latter case

[21] X_n (upper case) is used to avoid confusion with x_k, a transmitted sample that is the direct input to the discrete-time channel.

tacitly will imply a doubling of the number of real dimensions.

4.5.1 Optimal Partitioning Vectors - Vector Coding

The last N ADC channel-output symbol samples in Figure 4.18 have vector representation, with time indexed within a symbol from $k = -\nu$ to $N-1$,

$$\begin{bmatrix} y_{N-1} \\ y_{N-2} \\ \vdots \\ y_0 \end{bmatrix} = \begin{bmatrix} p_0 & p_1 & ... & p_\nu & 0 & ... & 0 \\ 0 & p_0 & \ddots & p_{\nu-1} & p_\nu & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & ... & 0 & p_0 & p_1 & ... & p_\nu \end{bmatrix} \begin{bmatrix} x_{N-1} \\ \vdots \\ x_0 \\ x_{-1} \\ \vdots \\ x_{-\nu} \end{bmatrix} + \begin{bmatrix} n_{N-1} \\ n_{N-1} \\ \vdots \\ n_0 \end{bmatrix} \quad , \tag{4.184}$$

or more compactly,

$$\boldsymbol{y} = P\boldsymbol{x} + \boldsymbol{n} \quad . \tag{4.185}$$

The $N \times (N+\nu)$ matrix P has a **singular value decomposition**

$$P = F \left[\Lambda \vdots \mathbf{0}_{N,\nu} \right] M^* \quad , \tag{4.186}$$

where F is a $N \times N$ unitary ($FF^* = F^*F = I$) matrix, M is a $(N+\nu) \times (N+\nu)$ unitary ($MM^* = M^*M = I$)matrix, and $\mathbf{0}_{N,\nu}$ is an $N \times \nu$ matrix of zeros. Λ is an $N \times N$ diagonal matrix with **singular values** λ_n, $n = 1, ..., N$ along the diagonal. The vector $\boldsymbol{x}$ is a data symbol vector. The notational use of P for the channel matrix suggests that any anti-alias analog filters at transmitter and receiver have been convolved with the channel impulse response $h(t)$ and included in the discrete-time response of the matrix channel.

Vector Coding creates a set of N parallel independent channels by using the transmit basis vectors, $\boldsymbol{m}$, that are the first N columns of M - in other words, the transmit vector $\boldsymbol{x}$ is obtained from the N vector components X_n, $n = 1, ..., N$ according to

$$\boldsymbol{x} = M \begin{bmatrix} \boldsymbol{X} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = [\boldsymbol{m}_{N-1}\, \boldsymbol{m}_{N-2}\, ...\, \boldsymbol{m}_1 \boldsymbol{m}_0\, ...\, \boldsymbol{m}_{-\nu}] \begin{bmatrix} X_{N-1} \\ X_{N-2} \\ \vdots \\ X_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \sum_{n=0}^{N-1} X_n \boldsymbol{m}_n \quad . \tag{4.187}$$

4.5.2 Creating the set of Parallel Channels for Vector Coding

The last ν columns of M that do not affect P because they are multiplied by zero in (4.187), these ν columns cannot contribute to the channel output and can be ignored in implementation. The corresponding vector-coding receiver uses discrete "matched filters" as the rows of F^*, forming

$$\boldsymbol{Y} = F^*\boldsymbol{y} = \begin{bmatrix} f_{N-1}^*\boldsymbol{y} \\ ... \\ f_0^*\boldsymbol{y} \end{bmatrix} \quad . \tag{4.188}$$

The mathematical description of the resultant N parallel channels is

$$\boldsymbol{Y} = \Lambda \boldsymbol{X} + \boldsymbol{N} \quad , \tag{4.189}$$

or, for each independent channel or entry in $\boldsymbol{Y}$,

$$Y_n = \lambda_n \cdot X_n + N_n \quad . \tag{4.190}$$

Since F is unitary, the noise vector $\boldsymbol{N}$ is also additive white Gaussian with variance per real dimension identical to that of $\boldsymbol{n}$, or $\frac{\mathcal{N}_0}{2}$.

Theorem 4.5.1 (Optimality of Vector Coding) *Vector Coding has the maximum $SNR_{m,u}$ for any discrete channel partitioning.*

Proof: The proof of this theorem is deferred to Chapter 5 (Lemma 5.4.1) where some not-yet-introduced concepts will make it trivial.

The optimality of VC is only strictly true when capacity-achieving codes with "Gaussian" distributions are used on each of the subchannels. The restriction to independent symbols, requiring the receiver to ignore the first ν samples of any symbol causes suboptimality. If this restriction were relaxed or omitted, then it is possible that a better design exists. For $N >> \nu$, such a potentially better method would offer only small improvement, so only the case of $\nu > .1N$ would be of interest in further investigation of this restriction (see Section 5.7).

Comment on correlated noise:

In the discrete-time transmission system of Figure 4.18, the noise is unlikely to be white. Section 1.7 previously indicated how to convert any channel into an equivalent white-noise channel, which involved receiver preprocessing by a canonical noise-whitening filter and the equivalent channel frequency response becoming the ratio of the channel transfer function to the square-root of the power-spectral density of the noise. Since this filter may be difficult to realize in practice, then the discrete-time processing also described in Section 1.7 of factoring the noise covariance to

$$E\left[\boldsymbol{n}\boldsymbol{n}^*\right] = R_{nn}\sigma^2 = R_{nn}^{1/2}R_{nn}^{-1/2}\sigma^2 \quad , \tag{4.191}$$

can be used. Then a discrete white-noise equivalent channel becomes

$$\boldsymbol{y} \leftarrow R_{nn}^{-1/2}\boldsymbol{y} = \left(R_{nn}^{-1/2}P\right)\boldsymbol{x} + \tilde{\boldsymbol{n}} \quad . \tag{4.192}$$

The extra matrix multiply is absorbed into the receiver processing as $F^*\boldsymbol{y} \rightarrow F^*R_{nn}^{-1/2}\boldsymbol{y}$. The preceding development the continues with P replaced by the matrix $R_{nn}^{-1/2}P$, the later of which then undergoes SVD. The matrix P will not be Toeplitz any longer, but will be very close to Toeplitz if the duration of the noise whitening filter is much less than the symbol period. In any case, vector coding does not require P to be Toeplitz.

4.5.3 An SNR for vector coding.

The number of bits/symbol for vector coding is achieved on the channel using a coding method with constant gap Γ (regardless of b_n) on all subchannels. This b is

$$b = (1/k)\sum_{n=1}^{N}\log_2\left(1 + \frac{\mathrm{SNR}_n}{\Gamma}\right) \quad , \tag{4.193}$$

where $k = 2$ for real subchannels and $k = 1$ for complex subchannels. When $\Gamma = 1$, the data rate on each subchannel is as high as possible with the given energy $\mathcal{E}_n$ and is sometimes called the "mutual information." (See Chapter 5 for more on mutual information). If the water-fill energy distribution is also used, then b is the capacity $b = c$ or highest possible number of bits per channel that can be reliably transmitted on the discrete-time channel. The best energy distribution for vector coding and $\Gamma > 1$ is a water-fill with $\mathrm{SNR}_n \rightarrow \mathrm{SNR}_n/\Gamma$, as in Section 4.3:

$$\mathcal{E}_n + \frac{\frac{\mathcal{N}_0}{2}\cdot\Gamma}{\lambda_n^2} = K \tag{4.194}$$

$$\sum_{n=1}^{N^*}\mathcal{E}_n = (N+\nu)\bar{\mathcal{E}}_{\boldsymbol{x}} \tag{4.195}$$

The SNR for vector coding is then

$$\mathrm{SNR}_{vc} = \Gamma \cdot \left[\prod_{n=1}^{N^*} \left(1 + \frac{\mathrm{SNR}_n}{\Gamma} \right) \right]^{1/(N+\nu)} - \Gamma \quad . \tag{4.196}$$

The quantity $(N+\nu)\bar{\mathcal{E}}_{\boldsymbol{x}}$ is now the total energy. The exponent depending on $N+\nu$ is used because vector coding uses $N+\nu$ channel input samples[22]. When $\Gamma = 1$, for complex passband systems,

$$\bar{b} = \frac{b}{2(N+\nu)} = .5 \log_2 \left(1 + \frac{\mathrm{SNR}_{vc}}{\Gamma} \right) \quad , \tag{4.197}$$

and

$$\bar{b} = \frac{b}{(N+\nu)} = .5 \log_2 \left(1 + \frac{\mathrm{SNR}_{vc}}{\Gamma} \right) \tag{4.198}$$

for real (PAM) baseband systems. Thus, the geometric SNR of vector coding can be compared against the detection SNR of an AWGN, a MMSE-LE, or a MMSE-DFE because SNR_{vc} has the same relation to achievable data rate with the same class of modulation codes of constant gap Γ. Further, we have the interesting result

$$\mathrm{SNR}_{vc} = 2^{2\bar{I}} - 1 \quad , \tag{4.199}$$

and

$$\mathrm{SNR}_{vc} = 2^{2\bar{C}} - 1 \tag{4.200}$$

when water-filling is used with $\Gamma = 1$. Thus the SNR is maximum when the data rate is maximized at capacity using water-filling. This SNR_{vc} could then replace $\mathrm{SNR}_{\mathrm{MFB}}$ as the highest possibleon an ISI channel. However it needs to be viewed with respect to the used bandwidth – equivalently the $\mathrm{SNR}_{\mathrm{MFB}}$ is usally a function of a different transmit-filter choice than VC. Also, SNR_{vc} is highest when capacity-achieving codes are used and is a function of the gap, unlike $\mathrm{SNR}_{\mathrm{MFB}}$.

EXAMPLE 4.5.1 (Vector Coding for $1 + .9D^{-1}$ Channel) For this channel, a guard period of one sample $\nu = 1$ is sufficient to prevent overlap between symbol transmissions at the channel output. A choice of $N = 8$ is consistent with previous invocations of this channel. The total energy is then $\mathcal{E} = (8+1) \cdot 1 = 9$. This energy may be distributed over a maximum of 8 subchannels. The channel-description matrix is

$$p = \begin{bmatrix} .9 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & .9 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & .9 & 1 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & .9 & 1 \end{bmatrix} \quad . \tag{4.201}$$

Singular value decomposition (implemented by the "svd" command in Matlab) produces the following 8 nonzero singular values

$$[1.87\ 1.78\ 1.64\ 1.45\ 1.22\ .95\ .66\ .34] \quad , \tag{4.202}$$

and corresponding subchannel SNR's

$$g_n = \frac{\lambda_n^2}{\sigma^2} = [19.3\ 17.6\ 15.0\ 11.7\ 8.3\ 5.0\ 2.4\ .66] \quad . \tag{4.203}$$

Execution of water-filling with $\Gamma = 0$ dB on the subchannels finds that only 7 can be used and that

$$K = \frac{1}{7} \left(9 + \sum_{n=0}^{6} \frac{1}{g_n} \right) = 1.43 \quad . \tag{4.204}$$

[22]The number of samples is $N+\nu$ dimensions for real channels, but $2N+2\nu$ dimensions for complex channels. In the complex case, each SNR factor would be squared and then (4.196) holds in either the complex or real case without further modification

The corresponding subchannel energies are:

$$[1.38\ 1.37\ 1.36\ 1.34\ 1.30\ 1.23\ 1.01\ 0] \quad , \tag{4.205}$$

and the SNR's are

$$[26.6\ 24.2\ 20.4\ 15.8\ 10.8\ 6.2\ 2.4\ 0] \quad , \tag{4.206}$$

resulting in an SNR of

$$\mathrm{SNR}_{VC} = \left[\prod_{n=0}^{6} \mathrm{SNR}_n + 1\right]^{1/9} - 1 = 6.46 = 8.1 \text{ dB} \quad . \tag{4.207}$$

The capacity for $N = 8$ and $\nu = 1$ is then

$$\bar{c} = \frac{1}{9}\sum_{n=0}^{6} \frac{1}{2}\log_2(1 + \mathrm{SNR}_n) = 1.45 \text{bits/dim.} \tag{4.208}$$

The capacity is close to the true capacity of 1.55 bits/dimension, which would be determined by executing this example as $N \to \infty$. These subchannels are exactly independent, and the SNR and capacity are exact - no approximations of no-ISI on a subchannel were made. The SNR is then 8.8 dB and exceeds the previous best equalized SNR on this channel, achieved by the MMSE-DFE. And, there is no error propagation. The overhead of $\nu = 1$ becomes zero as N increases.

4.6 Discrete Multitone (DMT) and OFDM Partitioning

Discrete MultiTone (DMT) and **Orthogonal Frequency Division Multiplexing (OFDM)** are two very common forms of Vector Coding that add a restriction to reduce complexity of implementation. Both DMT and OFDM have the same channel partitioning – they differ in the loading algorithm: OFDM puts equal bits on all subchannels, rather than optimizing b_n and $\mathcal{E}_n$, as in DMT. OFDM is used on broadcast (i.e., one-way channels) for which the receiver cannot tell the transmitter the bits and energies that are best. DMT is used on slowly time-varying two-way channels, like telephone lines. OFDM is used in wireless time-varying channels with codes that allow recovery of lost subchannels caused by time-varying notches in multipath intersymbol interference.

The DMT/OFDM channel partitioning forces the transmit symbol to have $x_{-k} = x_{N-k}$ for $k = 1, ...\nu$. Such repeating of the last ν samples at the beginning of the symbol is called a **cyclic prefix.** Tremendous processing simplification occurs with the cyclic prefix.

With cyclic prefix, the matrix description of the channel has a square $N \times N$ circulant P matrix:

$$\begin{bmatrix} y_{N-1} \\ y_{N-1} \\ \vdots \\ y_0 \end{bmatrix} = \begin{bmatrix} p_0 & p_1 & ... & p_\nu & 0 & ... & 0 \\ 0 & p_0 & \ddots & p_{\nu-1} & p_\nu & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & ... & 0 & p_0 & p_1 & ... & p_\nu \\ p_\nu & 0 & ... & 0 & p_0 & ... & p_{\nu-1} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ p_1 & ... & p_\nu & 0 & ... & 0 & p_0 \end{bmatrix} \begin{bmatrix} x_{N-1} \\ \vdots \\ x_0 \end{bmatrix} + \begin{bmatrix} n_{N-1} \\ n_{N-1} \\ \vdots \\ n_0 \end{bmatrix} \tag{4.209}$$

$$\boldsymbol{y} = \tilde{P}\boldsymbol{x} + \boldsymbol{n} \quad . \tag{4.210}$$

Singular value decomposition produces unique singular values if those values are restricted to be nonnegative real values (even when P is complex). For transmission, this restriction is superfluous and a variant of SVD will be simpler to compute. For the case of the cyclic prefix, the SVD may be replaced by the eigendecomposition or "spectral factorization" of the circulant matrix P,

$$P = M\Lambda M^* \tag{4.211}$$

where $MM^* = M^*M = I$ and Λ is a square diagonal matrix with eigenvalues on the diagonal λ_n. The magnitude of eigenvalues are equal to the singular values. Further, effectively, M and F become the same matrix for this special case. The product SNR for this alternative factorization in (4.211) is the same as for SVD-factorization because the magnitude of each eigenvalue equals a corresponding singular value.[23] This type of transmission could be called **Eigenvector Coding**, analogous to modal modulation's use of eigenfunctions. While the subchannel SNRs are the same, there are many choices for the "transform" matrices M and F in partitioning. Since all have the same performance, this subsection investigates a very heavily used choice that has two advantages – (1) the matrices M and F are not a function of the channel, and (2) they are efficiently implemented.

4.6.1 The Discrete Fourier Transform

A review of the DFT is helpful for DMT channel partitioning.

Definition 4.6.1 (Discrete Fourier Transform) *The* **Discrete Fourier Transform (DFT)** *of a N-dimensional input symbol $\boldsymbol{x}$ is*

$$\boldsymbol{X} \triangleq \begin{bmatrix} X_{N-1} \\ X_{N-2} \\ \vdots \\ X_0 \end{bmatrix} \tag{4.212}$$

[23] A caution to the reader using Matlab for eigenvalue/vector computation – the eigenvalues are not ususally ordered as they are in singular-value decomposition.

where

$$X_n = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} x_k e^{-\jmath \frac{2\pi}{N} kn} \ \ \forall \ \ n \in [0, N-1] \quad , \tag{4.213}$$

and

$$\boldsymbol{x} = \begin{bmatrix} x_{N-1} \\ x_{N-2} \\ \vdots \\ x_0 \end{bmatrix} \quad . \tag{4.214}$$

The corresponding **Inverse Discrete Transform (IDFT)** *is computed by*

$$x_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} X_n e^{\jmath \frac{2\pi}{N} kn} \ \ \forall \ \ k \in [0, N-1] \quad . \tag{4.215}$$

In matrix form, the DFT and IDFT are

$$\boldsymbol{X} = \mathcal{Q}\boldsymbol{x} \tag{4.216}$$

$$\boldsymbol{x} = \mathcal{Q}^* \boldsymbol{X} \quad , \tag{4.217}$$

where $\mathcal{Q}$ is the orthonormal matrix given by

$$\mathcal{Q} = \frac{1}{\sqrt{N}} \begin{bmatrix} e^{-\jmath \frac{2\pi}{N}(N-1)(N-1)} & ... & e^{-\jmath \frac{2\pi}{N} 2(N-1)} & e^{-\jmath \frac{2\pi}{N}(N-1)} & 1 \\ e^{-\jmath \frac{2\pi}{N}(N-1)\cdot(N-2)} & ... & e^{-\jmath \frac{2\pi}{N} 2(N-2)} & e^{-\jmath \frac{2\pi}{N}(N-2)} & 1 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ e^{-\jmath \frac{2\pi}{N}(N-1)} & ... & e^{-\jmath \frac{2\pi}{N} 2} & e^{-\jmath \frac{2\pi}{N}} & 1 \\ 1 & ... & 1 & 1 & 1 \end{bmatrix} \quad . \tag{4.218}$$

Both $\boldsymbol{x}$ and $\boldsymbol{X}$ can be presumed to be one period of a periodic sequence to generate DFT/IDFT values for indices k and/or n outside the interval $(0, N-1)$. The matrix $\mathcal{Q}^*$ appears the same as $\mathcal{Q}$ in (4.218) except that the exponents all have a plus sign where the minus now occurs. A further definition is of some vectors $\boldsymbol{q}_n$ according to

$$\mathcal{Q}^* = [\boldsymbol{q}_{N-1} \ , ..., \ \boldsymbol{q}_0] \quad , \tag{4.219}$$

which are useful in the following theorem's proof.

The IDFT, X_n, and DFT, x_k, values can be complex. For real time-domain signals, the restriction $X_{N-n} = X_n^*$ for $n = 1, ..., N-1$ holds. This restriction implies that there are $\bar{N} = N/2$ complex dimensions when N is even[24]. The IDFT generally corresponds to N input samples whether complex or real, while this textbook has previously used N to be the number of real dimensions and $\bar{N}$ to be the number of tones. Thus, there is a slight inconsistency in notation. This inconsistency is resolved simply by considering N in the IDFT and DFT definitions to be the number of tones used if complex, and twice the number of tones with the congjugate symmetry on the IDFT input enforced (or real time-domain signals). The FFT size should then be equal to the number of real dimensions only in the latter case of real baseband signals. When complex baseband signals are used, the number of real dimensions is twice the FFT size. With this understanding, Matlab can be used to easily generate the Q matrix here exactly according to the commands

```
J=hankel([ zeros(1,N-1) 1]')
Q=(1/sqrt(N))*J*fft(J)
```

[24] Actually, there are $\bar{N} - 1$ complex dimensions and two real dimensions corresponding to $n = 0$ and $n = \bar{N}$. The extra two real dimensions are often grouped together and considered to be one "complex" dimension, with possibly unequal variances in the 2 dimensions. Usually, practical systems use neither of the real dimensions, leaving $\bar{N} - 1$ subchannels

Lemma 4.6.1 (DFT is M for circularly prefixed VC) *The circulant matrix $P = M\Lambda M^*$ has a eigendecomposition with $M = \mathcal{Q}^*$ - that is the eigenvectors are the columns of the IDFT matrix and not dependent on the channel. Furthermore, the diagonal values of Λ are $\lambda_n = P_n$ - that is, the eigenvalues of this circulant matrix are found through the DFT.*

Proof: The eigenvalues and corresponding eigenvectors of a matrix are found according to

$$\lambda_n \boldsymbol{q}_n = P\boldsymbol{q}_n \tag{4.220}$$

$$= \begin{bmatrix} p_0 & p_1 & \dots & p_\nu & 0 & \dots & 0 \\ 0 & p_0 & \ddots & p_{\nu-1} & p_\nu & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ p_1 & \dots & p_\nu & 0 & \dots & 0 & p_0 \end{bmatrix} \frac{1}{\sqrt{N}} \cdot \begin{bmatrix} 1 \\ e^{-\jmath\frac{2\pi}{N}n} \\ \vdots \\ e^{-\jmath\frac{2\pi}{N}n} \end{bmatrix} \quad , \tag{4.221}$$

which has matrix equivalent $\mathcal{Q}^*\Lambda = P\mathcal{Q}^*$. By investigating the top row on right,

$$\frac{1}{\sqrt{N}} \cdot \left(p_0 + p_1 \cdot e^{-\jmath\frac{2\pi}{N}n} + \dots + p_{N-1} \cdot e^{-\jmath\frac{2\pi(N-1)}{N}n} \right) = P_n \quad , \tag{4.222}$$

so $\lambda_n = P_n$. The next row up is then

$$\frac{1}{\sqrt{N}} \cdot \left(p_0 \cdot e^{\jmath\frac{2\pi}{N}n} + p_1 + \dots + p_{N-1} \cdot e^{-\jmath\frac{2\pi(N-2)}{N}n} \right) = P_n \cdot \frac{e^{+\jmath\frac{2\pi}{N}n}}{\sqrt{N}} \quad . \tag{4.223}$$

Clearly, repeating this procedure leads to

$$P\boldsymbol{q}_n = P_n\boldsymbol{q}_n \quad , \tag{4.224}$$

making $\boldsymbol{q}_n$ an eigenvector of P. Thus, a choice of the eigenvectors is simply the columns of the IDFT matrix $\mathcal{Q}^*$. For this choice, the eigenvalues must exactly be the DFT values of the rows of P. **QED.**

4.6.2 DMT

Discrete Multi-tone (DMT) transmission uses the cyclic prefix and $M = FQ^*$ in vector coding for the constructed cyclic channel. These matrices are not a function of the channel because of the use of the cyclic prefix. **Orthogonal Frequency Division Multiplexing (OFDM)** also uses these same vectors (but does not use optimum loading, and instead sends equal energy and information on all subchannels).

Thus, the equivalent of the F and M matrices for VC with a cyclic prefix become the well-known DFT operation. The DFT can be implemented with $N \cdot log_2(N)$ operations, which is less than the full matrix multiplies of vector coding that would require N^2 equivalent operations. For large N, this is a very large computational reduction. Clearly, since an additional restriction of cyclic prefix is applied only to the case of DMT/OFDM channel partitioning, then DMT/OFDM can only perform as well as or worse than VC without this restriction. When $N >> \nu$, the difference will be small simply because the difference in P is small. The designer should take care to note that the energy constraint for DMT when executing loading algorithms is lower by the factor $N/(N+\nu)$ because of the "wasted energy" in the cyclic prefix, the price for the computational simplification.

Figure 4.19 illustrates the basic structure of the DMT transmission-system partitioning. Determination of the subsymbol values X_n, energy $\mathcal{E}_n$ and number of bits b_n occurs with a loading algorithm. The partitioning then uses the computationally efficient FFT algorithms for implementation of the DFT and IDFT.

For the case of $N \to \infty$, the cyclic prefix becomes negligible and vector coding performs the same as DMT, and all the individual subchannels of DMT also have the same SNR_n as VC as $N \to \infty$. This is consistent with the earlier finding that eigenvector coding in the limit of large symbol size and infinite number of subchannels becomes MT. VC is a discrete form of eigenvector coding while DMT is a discrete form of MT. In the limit, with sufficiently high sampling rate, DMT, MT, VC and eigenvector coding all perform the same. Only DMT is easily implemented. No other channel partitioning can perform better as long as N is sufficiently large.

The following list may help summarize two SNR's of interest, SNR_{MFB} and $\text{SNR}_{\text{DMT,U}}$:

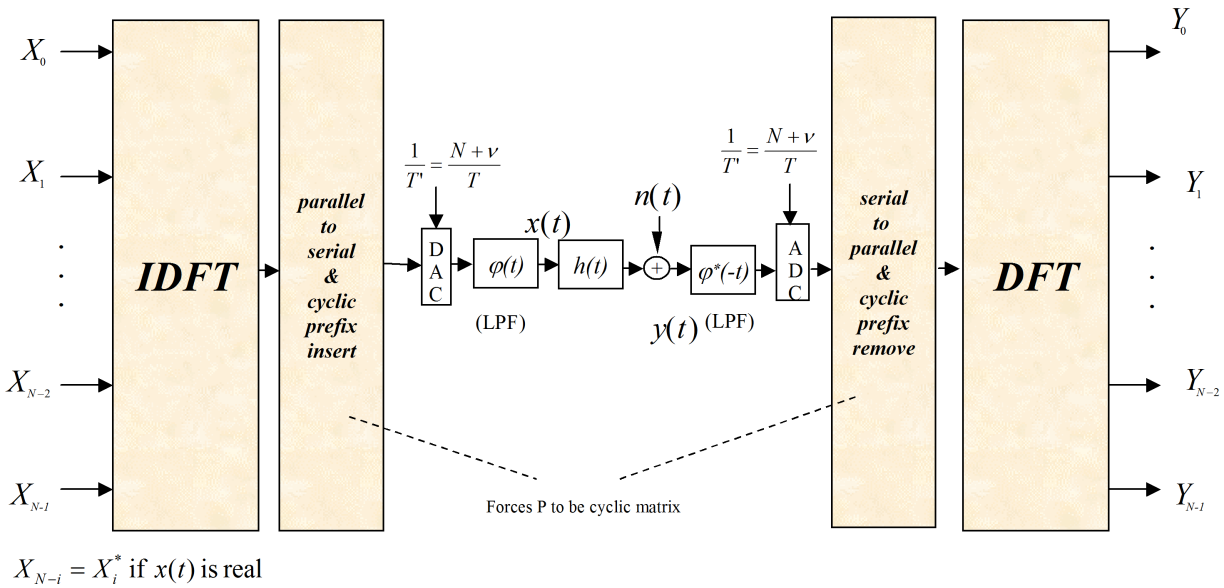


Figure 4.19: DMT and OFDM partitioning block diagram.

- $\text{SNR}_{\text{DMT,U}}$ is not a bound and is attainable, and $\text{SNR}_{\text{DMT,U}} \leq \text{SNR}_{\text{MFB}}$.
- outside of the cyclic-prefix loss, which tends to zero for infinite block length, $\text{SNR}_{\text{DMT,U}}$ is the highest possible SNR that can be achieved, so it is a better bound than SNR_{MFB} was in Chapter 3,
- If $\Gamma = 0$ dB, then there is no better system than DMT as the block length becomes large.
- An apparent quandry might be that in Chapter 9, MLSD attains the matched filter bound SNR for various finite-length impulse-response channels, but those situations have $Gamma > 0$ dB always and the sequence detector is in fact taking advantage of a code. If a code with zero gap had been used, MLSD for intersymbol interference will not exceed the performance of $\text{SNR}_{\text{DMT,U}}$ with zero dB gap.
- Thus, as will be discussed in Chapters 10 and beyond, the designer should not confuse a coding gain with a gain caused by partitioning of the channel ISI, for which there is no better method than DMT as $N \to \infty$.

EXAMPLE 4.6.1 (DMT for $1 + .9D^{-1}$ Channel) Continuing the example for the $1 + .9D^{-1}$ channel for DMT, the circulant channel matrix for $N = 8$ and $\nu = 1$ becomes

$$P = \begin{bmatrix} .9 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & .9 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & .9 & 1 & 0 & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & .9 \end{bmatrix} . \tag{4.225}$$

Table 4.6.1 summarizes the results of waterfilling with **8** units of energy (not 9, because 1 unit is wasted in the cyclic prefix on average). Only 7 of the 8 subchannels were used - note that 6 of the subchannels are effectively two-dimensional QAM subchannels.

n	$\lambda_n = \|P_n\|$	$g_n = \frac{\|P_n\|^2}{.181}$	$\mathcal{E}_n$	SNR_n	b_n
0	1.90	20	1.24	24.8	.969
1	1.76	17	1.23	20.9	2.34
2	1.76	17	1.23	20.9	2.34
3	1.35	9.8	1.19	11.7	2.23
4	1.35	9.8	1.19	11.7	2.23
5	.733	3	.96	2.9	1.85
6	.733	3	.96	2.9	1.85
7	.100	.05525	0	0	0

Table 4.3: Table of parameters for $1 + .9D^{-1}$ channel example with DMT with $\Gamma = 0$ dB.

The SNR is

$$\text{SNR}_{DMT} = \left[\prod_{n=0}^{6}(1+\text{SNR}_n)\right]^{1/9} - 1 = 7.6 \text{ dB} < \ 9.2 \text{ dB from Example 4.1.1.} \qquad (4.226)$$

The DMT SNR is less than that found earlier for MT, but is exact and not approximated. It is also slightly worse than the SNR of vector coding for the same number of dimensions. However, DMT can increase the number of dimensions significantly before the complexity of FFT's exceed that of the orthogonal matrix multiplies. One finds that for $N = 16$, this SNR increases to its maximum value of 8.8 dB.

This exampleprogresses to compare the complexity of this DMT system with its exact SNR now correctly computed with that of the DFE studied earlier and also against the VC system. To get 7.6 dB, the MMSE-DFE required 3 feedforward taps and 1 feedback tap for a complexity of 4 multiply-accummulates per sample. VC requires 7(8) multiply-accumulates in the receiver to get the slightly higher SNR of 8.1 dB, or a complexity 6.2/sample (6.2 = 7(8)/9). DMT requires a size-8 FFT, which nearly exactly requires $8\log_2(8)$ multiply accumulates when the real-output constraint is exploited. DMT then gets 7.6 dB also, but with 2.7 multiplies/sample. A factor of 2-3 improvement in complexity for a given level of performance with respect to the DFE and VC is common. For the size 16 DMT with SNR=8.8 dB, the complexity is still only 3.8/sample while for instance a DFE requires 10 feedforward taps and 1 feedback to get only 8.4 dB, then requiring 11/sample. Further, the DFE incurs a precoding loss of 1.2 dB.

Matlab programs

The following matlab functions have been written with credits to former students W. Rhee and A. Salvekar, and are useful for computing DMT performance and parameters quickly. See class web site for complete programs.

```
--------- rate-adaptive water-fill DMT -----------------------------------
function [gn,en_bar,bn_bar,Nstar,b_bar,SNRdmt]=DMTra(P,SNRrcvr,Ex_bar,N,gap)

% Updated J. Cioffi , Spring 2008 - credits to A. Salvekar and W. Rhee
%
% function [gn,en_bar,bn_bar,Nstar,b_bar]=dmtra(P,SNRrcvr,Ex_bar,N,gap)
%
% INPUTS
% P is the pulse response (and v is set equal to length(P)-1)
% Ex_bar is the normalized energy
```

```
% SNRrcvr is the receiver SNR (dB) if input were i.i.d. Ex_bar all
% dimensions, so what would be called an SNRmfb for that "white" input
% N is the (maximum) number of DMT tones, N>2, centered at 1/N
%   the sampling rate is presumed to be (N+v)/T where 1/T is DMT symbol
%   rate
% gap is the gap in dB
%
% OUTPUTS
% gn is vector of channel gains
% en_bar is the vector of energy/dim for all subchannel
% bn_bar is the vector of bit/dim for all subchannels
% Nstar is the number of used subchannel
% b_bar is the bit rate
% SNRdmt is the equivalet DMT SNR

% dB into normal scale
Noise_var=Ex_bar*(norm(P)^2)/(10^(SNRrcvr/10));
gap=10^(gap/10);
nu=size(P);
nu=max(nu(1),nu(2))-1;

% initialization
en=zeros(1,N);
bn=zeros(1,N);
gn=zeros(1,N);
Hn = zeros(1,N);

% subchannel center frequencies
f=-1/2+1/N:1/N:1/2;

% find Hn vector
for i=1:length(P)
Hn=Hn+P(i)*exp(j*2*pi*f*(i-1));
end

% find gn vector
gn=abs(Hn).^2/Noise_var;

%%%%%%%%%%%%%%%%%%%%%%
% Waterfilling for DMT %
%%%%%%%%%%%%%%%%%%%%%%

%sort
[gn_sorted, Index]=sort(gn);  % sort gain, and get Index

gn_sorted = fliplr(gn_sorted);% flip left/right to get the largest
                              % gain in leftside
Index = fliplr(Index);        % also flip index


num_zero_gn = length(find(gn_sorted == 0)); %number of zero-gain subchannels
Nstar=N - num_zero_gn;
  % Number of used channels,
  % start from N - (number of zero gain subchannels)
```

```
while(1)
  K=1/Nstar*(N*Ex_bar+gap*sum(1./gn_sorted(1:Nstar)));
  En_min=K-gap/gn_sorted(Nstar); % En_min occurs in the worst channel
  if (En_min<0)
     Nstar=Nstar-1;  % If negative En, continue with less channels
  else
     break;       % If all En positive, done.
  end
end

En=K-gap./gn_sorted(1:Nstar);  % Calculate En
Bn=.5*log2(K*gn_sorted(1:Nstar)/gap);  % Calculate bn

bn(Index(1:Nstar))=Bn; % return values in original index
en(Index(1:Nstar))=En; % return values in original index

middle = N/2; % Since channel is even, need to display
                                % only half of result
en_bar=en(middle:N);
bn_bar=bn(middle:N);

% calculate b_bar
b_bar=1/(N+nu)*(sum(bn));
SNRdmt=10*log10(2^(2*b_bar)-1);

--------------------------------------------------------------------------------

--------- margin-adaptive water-fill DMT -----------------------------------
function [gn,en_bar,bn_bar,Nstar,b_bar_check,margin]=DMTma(P,SNRrcvr,Ex_bar,b_bar,N,gap)

% EE379C 2008 Spring,
% J. Cioffi - credits to Seong Taek Chung
%
% P is the pulse response where v=length(P) -1
% SNRrcvr is the SNR (in dB) at te receiver input if the input is i.i.d
% b_bar is the target normalized bit rate
% Ex_bar is the normalized energy
% N is the total number of real/complex subchannels, N>2
% gap is the gap in dB
%
% gn is the vector of channel gains
% en_bar is the vector of energy/dim for all subchannels
% bn_bar is the vector of bit/dim for all subchannels
% Nstar is the number of used subchannel

% dB into normal scale
Noise_var=Ex_bar*(norm(P)^2)/(10^(SNRrcvr/10));
gap=10^(gap/10);
nu=length(P)-1;

% initialization
en=zeros(1,N);
bn=zeros(1,N);
```

```
gn=zeros(1,N);
Hn = zeros(1,N);

% subchannel center frequencies
f=-1/2+1/N:1/N:1/2;

% find Hn vector
for i=1:length(P)
Hn=Hn+P(i)*exp(j*2*pi*f*(i-1));
end

% find gn vector
gn=abs(Hn).^2/Noise_var;

%%%%%%%%%%%%%%%%%%%%%%%
% MA waterfilling %
%%%%%%%%%%%%%%%%%%%%%%%


%sort
[gn_sorted, Index]=sort(gn);  % sort gain, and get Index

gn_sorted = fliplr(gn_sorted);% flip left/right to get the largest
                              % gain in leftside
Index = fliplr(Index);        % also flip index

num_zero_gn = length(find(gn_sorted == 0)); %number of zero gain subchannels
Nstar=N - num_zero_gn;
  % Number of used channels,
  % start from N - (number of zero gain subchannels)
b=(N+nu)*b_bar;
Klog2tilde=2*b-(1/log(2))*sum(log(gn_sorted));

while(1)
  Klog2=(1/Nstar)*Klog2tilde + log(gap)/log(2);
  En_min=2^(Klog2)-gap/gn_sorted(Nstar); % En_min occurs in the worst channel
  if (En_min<0)
            Klog2tilde=Klog2tilde+(1/log(2))*log(gn_sorted(Nstar));
     Nstar=Nstar-1;  % If negative En, continue with less channels
  else
     break;       % If all En positive, done.
  end
end
K=2^(Klog2);
En=K-gap./gn_sorted(1:Nstar);  % Calculate En
Bn=.5*log2(K*gn_sorted(1:Nstar)/gap);  % Calculate bn

bn(Index(1:Nstar))=Bn; % return values in original index
en(Index(1:Nstar))=En; % return values in original index

en_bar=en;
bn_bar=bn;

% calculate b_bar
```

```
b_bar_check=1/(N+nu)*(sum(bn));

%calculate margin
margin=10*log10(N*Ex_bar/sum(en));

-----------------------------------------------------------------------------------------------

--------- rate-adaptive Levin-Campello DMT -----------------------------------
function [gn,En,bn,b_bar]=LC(P,SNRmfb,Ex_bar,Ntot,gap)

function [gn,En,bn,b_bar,SNRdmt]=DMTLCra(P,SNRrcvr,Ex_bar,N,gap)

%
% EE379C 2008 Spring
%
% Levin Campello's Method with DMT
%
% Inputs
% P is the pulse response(an nu is set to length(p) - 1
% SNRrcvr is the channel output SNR for an i.i.d. input
% Ex_bar is the normalized energy
% N is the total number of real/complex subchannels, N>2
% gap is the gap in dB
%
% Outputs
% gn is the vector of channel gains
% En is the vector energy distribution (PAM or QAM)
% bn is the vector bit distribution (PAM or QAM)
% b_bar is the number of bits per dimension in the DMT symbol
%
% The first and last bins are PAM; the rest are QAM.

% dB into normal scale
Noise_var=Ex_bar*(norm(P)^2)/(10^(SNRrcvr/10));
gap=10^(gap/10);
nu=length(P)-1;

% initialization
En=zeros(1,N/2+1);
bn=zeros(1,N/2+1);
gn=zeros(1,N/2+1);
Hn = zeros(1,N/2+1);
decision_table=zeros(1,N/2+1);

% subchannel center frequencies
f=0:1/N:1/2;

% find Hn vector
for i=1:length(P)
Hn=Hn+P(i)*exp(j*2*pi*f*(i-1));
end

% find gn vector
gn=abs(Hn).^2/Noise_var;
```

```
%debugging purpose
%plot(gn)

%%%%%%%%%%%%%%%%%%%%%%%%
% Levin Campello Loading %
%%%%%%%%%%%%%%%%%%%%%%%%

%initialization

%used energy so far
E_so_far=0;
%decision table - QAM and PAM
decision_table(2:N/2)=2*gap./gn(2:N/2);
% Gap formula incremental energies.
if gn(1) ~= 0
    decision_table(1)=3*gap/gn(1);
else
    decision_table(1)=inf;
end
if gn(N/2+1) ~=0
    decision_table(N/2+1)=3*gap/gn(N/2+1);
else
    decision_table(N/2+1)=inf;
end

%decision_table: debugging purpose

while(1)

    [y,index]=min(decision_table);
    E_so_far=E_so_far+y;

    if E_so_far > Ex_bar*N

        break;

    else

        En(index)=En(index)+y;
        bn(index)=bn(index)+1;

if (index ==1 | index == N/2+1)
  decision_table(index)=4*decision_table(index);
else
  decision_table(index)=2*decision_table(index);
end

    end

end

% calculate b_bar
b_bar=1/(N+nu)*(sum(bn));
```

```
SNRdmt=10*log10(gap*(2^(2*b_bar)-1));

--------------------------------------------------------------------------------

--------- margin-adaptive Levin-Campello DMT ----------------------------------
function [gn,en_bar,bn_bar,Nstar,b_bar_check,margin]=DMTma(P,SNRrcvr,Ex_bar,b_bar,N,gap)

% EE379C 2008 Spring,
% J. Cioffi - thanks to Seong Taek Chung
%
% P is the pulse response where v=length(P) -1
% SNRrcvr is the SNR (in dB) at te receiver input if the input is i.i.d
% b_bar is the target normalized bit rate
% Ex_bar is the normalized energy
% N is the total number of real/complex subchannels, N>2
% gap is the gap in dB
%
% gn is the vector of channel gains
% en_bar is the vector of energy/dim for all subchannels
% bn_bar is the vector of bit/dim for all subchannels
% Nstar is the number of used subchannel

% dB into normal scale
Noise_var=Ex_bar*(norm(P)^2)/(10^(SNRrcvr/10));
gap=10^(gap/10);
nu=length(P)-1;

% initialization
en=zeros(1,N);
bn=zeros(1,N);
gn=zeros(1,N);
Hn = zeros(1,N);

% subchannel center frequencies
f=-1/2+1/N:1/N:1/2;

% find Hn vector
for i=1:length(P)
Hn=Hn+P(i)*exp(j*2*pi*f*(i-1));
end

% find gn vector
gn=abs(Hn).^2/Noise_var;

%%%%%%%%%%%%%%%%%%%%%%%%
% MA waterfilling %
%%%%%%%%%%%%%%%%%%%%%%%%


%sort
[gn_sorted, Index]=sort(gn);  % sort gain, and get Index

gn_sorted = fliplr(gn_sorted);% flip left/right to get the largest
                              % gain in leftside
```

```
Index = fliplr(Index);        % also flip index

num_zero_gn = length(find(gn_sorted == 0)); %number of zero gain subchannels
Nstar=N - num_zero_gn;
  % Number of used channels,
  % start from N - (number of zero gain subchannels)
b=(N+nu)*b_bar;
Klog2tilde=2*b-(1/log(2))*sum(log(gn_sorted));

while(1)
  Klog2=(1/Nstar)*Klog2tilde + log(gap)/log(2);
  En_min=2^(Klog2)-gap/gn_sorted(Nstar); % En_min occurs in the worst channel
  if (En_min<0)
            Klog2tilde=Klog2tilde+(1/log(2))*log(gn_sorted(Nstar));
     Nstar=Nstar-1;  % If negative En, continue with less channels
  else
     break;       % If all En positive, done.
  end
end
K=2^(Klog2);
En=K-gap./gn_sorted(1:Nstar);  % Calculate En
Bn=.5*log2(K*gn_sorted(1:Nstar)/gap);  % Calculate bn

bn(Index(1:Nstar))=Bn; % return values in original index
en(Index(1:Nstar))=En; % return values in original index

en_bar=en;
bn_bar=bn;

% calculate b_bar
b_bar_check=1/(N+nu)*(sum(bn));

%calculate margin
margin=10*log10(N*Ex_bar/sum(en));
```

EXAMPLE 4.6.2 (ADSL) By far the most heavily used broadband internet delivery method in the world is asymmetric digital subscriber lines (ADSL) in Figure 4.20, allowing nominally 1.5 Mbps (range is 500 kbps to 8 Mbps) to customers on telephone lines **downstream** and roughly about 1/3 to 1/10 that rate from the customer to the telephone company **upstream** (where it is sent to the internet service provider). ADSL2+ extends to 512 tones while VDSL2 extends to a maximum of 4096 tones (with a possible double-spacing option of a symbol rate of 8 kHz).

ADSL is described ITU Standard G.992.1 and uses DMT. The symbol rate is $T = 250\mu$s. The downstream modulation uses 256 4.3125 kHz wide subchannels and a sampling rate of $1/T'$=2.208 MHz. The cyclic prefix is then necessarily 40 samples. Each time-domain symbol contains then 512+40 or 552 samples. Hermetian symmetry is used to create a real signal for transmission over the band from 0 to 1.104 MHz. Typically, the first 2-3 tones near DC and DC are not used to prevent interference into voiceband telephony (POTS = plain old telephone service), which shares the same telephone line as ADSL. Upstream transmission uses instead 32 tones to frequency 138 kHz. Tone 256 is also not used. Tone 64 (276 kHz) is reserved for pilot signal (known point in 4 QAM sent continuously) that is used to recover the symbol and sampling rates. The sampling rate upstream is 276 kHz and the cyclic prefix is 5 samples for a block size of 69 samples. Upstream is then exactly 1/8 downstream.

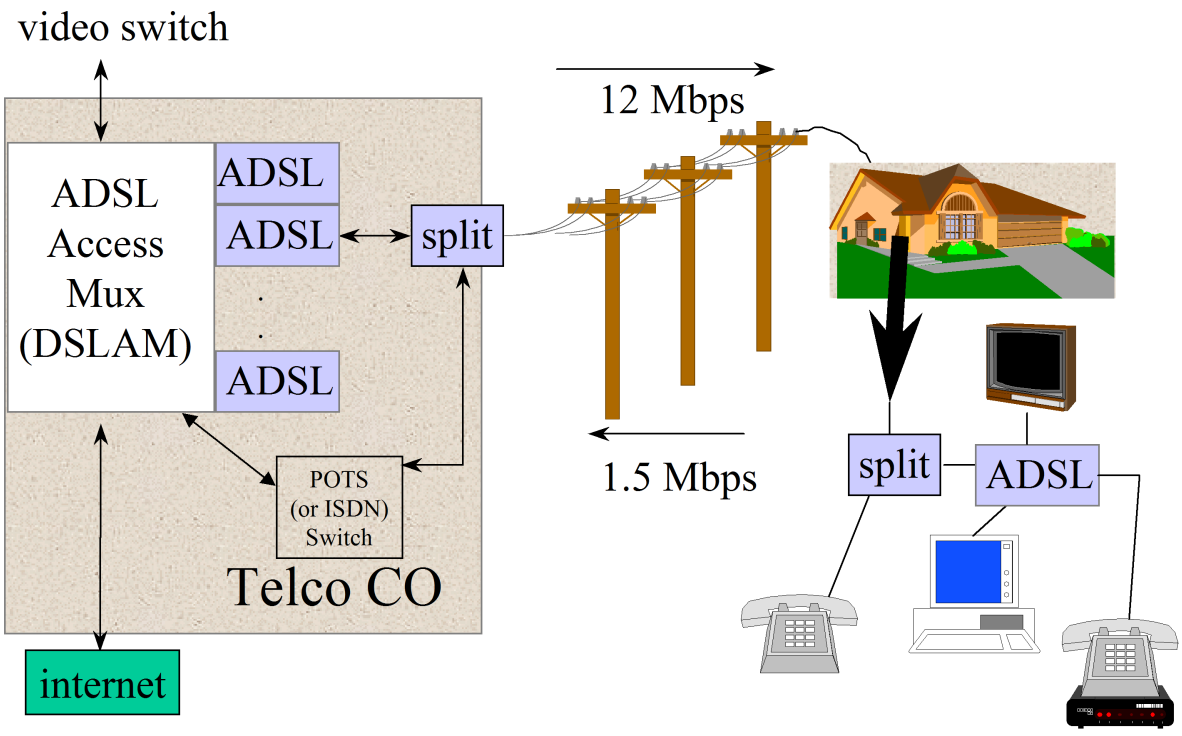


Figure 4.20: ADSL use example.

Downstream tones may or may not share the upstream band (when they do share, a device called an echo canceller is used to separate the two directions of transmission). Usually, the upstream and downstream bands are placed for further adaptive refinement by loading as shown in Figure 4.21. Some typical ADSL channels (taken from the American National Standards Institute) appear in Figure 4.22. The lines have significant variation with length, and also with impairments like bridged-taps (open circuited extension phones) that cause the ripple in the characteristics in the second graph in Figure 4.22. Also noises are highly frequency selective and could be nearby radio stations captured by the phone-line or other DSLs with varying bandwidths that are sufficiently close to crosstalk into the ADSL line. All this channel variation among possible lines for deployment of ADSL creates a huge need for adaptive loading, which will optimize the DMT performance for each situation.

A maximum power of 20.5 dBm is permitted in ADSL downstream and 14.5 dBm upstream. The maximum number of bits permitted to be loaded on to any single tone is $b_n \leq 15$. While antognists once said ADSL would fail as being too complex to implement, components that implement a complete modem (analog and digital) now sell for a few dollars. While the design may be conceptually complex, most customers simply call the phone company, procure a modem, plug and play in less than 15 minutes – a testimonial to the engineer – a conceptually complex design is acceptable if it keeps the customer use very simple (the opposite is almost never true).

Wireless transmission often uses OFDM transmission because no optimization at the transmitter of b_n is possible if the channel varies too rapidly with time. Instead, over a wide bandwidth in wireless transmission, multipath ISI typically results in notches in the transmission band. Part of the signal (some tones) is lost. Thus, with coding methods not discussed until later chapters of this book, it is possible to recover the information of several lost tones as long as the code is sufficiently powerful, no matter where those lost tones are located (that is, no matter which tone indices $\{n\}$ are affected). Also, in broadcast wireless transmission, there may be several different "channels" to each customer of a service, and no

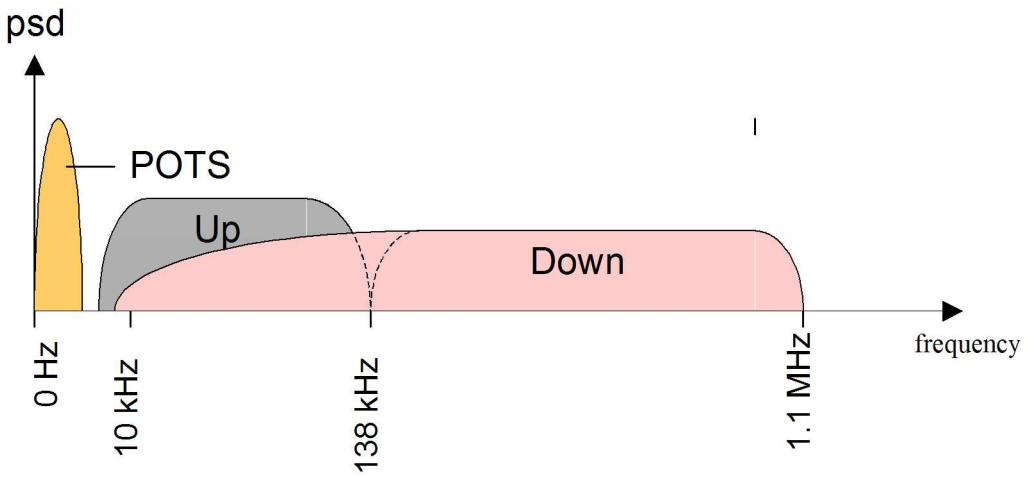
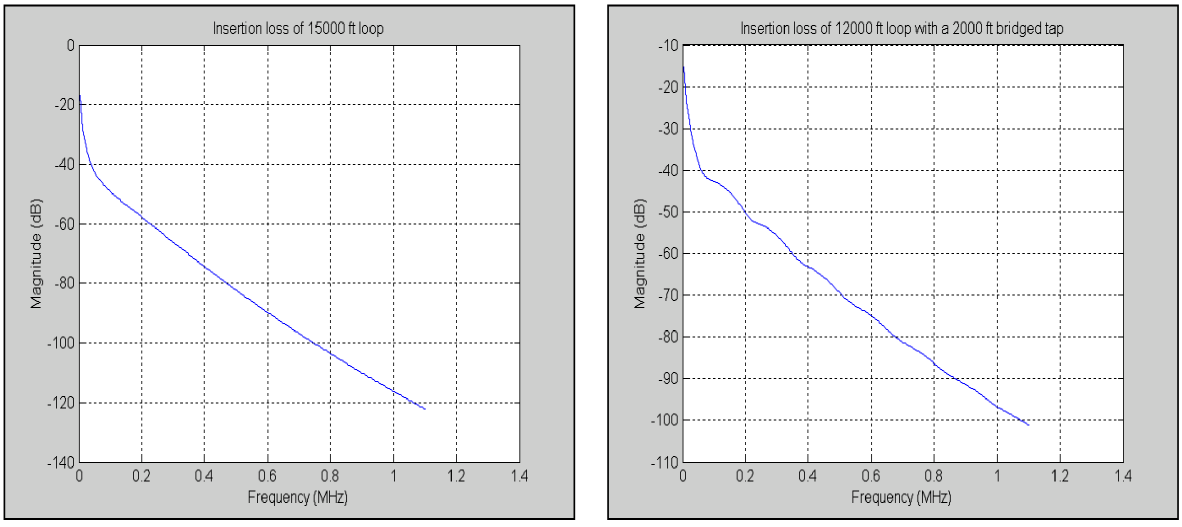


Figure 4.21: ADSL spectrum use - further adaption by loading within each band.



5 km loop

4 km loop with bridged tap

Figure 4.22: ADSL example lines.

R (Mbps)	constellation	code rate	b_n	b_n	b
6	BPSK	1/2	1/2	1/4	24
9	BPSK	3/4	3/4	3/8	36
12	4QAM	1/2	1	1/2	48
18	4QAM	3/4	3/2	3/4	72
24	16QAM	1/2	2	1	96
36	16QAM	3/4	3/2	3/4	144
48	64QAM	1/2	3	3/2	192
54	64QAM	3/4	9/2	9/4	216

Table 4.4: Table of IEEE802.11(a) data rates and associated parameters.

"feedback channel," so then the b_n can not be optimized as a function of the identified channel (which necessary must occur at the receiver, which is the only place the channel can be known practically and estimated). The use of OFDM in wireless transmission with codes is often called COFDM. COFDM is the most powerful method for handling the time-varying channel, even though OFDM might have been greatly inferior to DMT if the channel could have been known at the transmitter. COFDM's greatest asset is simply the practical advantage that the transmitter is basically fixed (an IFFT operation with fixed bit loading and energy on all tones) – actually the gain and phase of each subchannel is estimated and used, but only at the receiver. Rapid determination of a full DFE-like equalizer is difficult and loses performance greatly in wireless transmission, leaving multicarrier a superior method coincidentally for wireless transmission as well as wireline. However, the key is the additional code, as nominally an "un-loaded" or uncoded OFDM signal will not be very effective. This heuristic explanation then leads to the next couple of examples of OFDM partitioning use.

EXAMPLE 4.6.3 (802.11a Wireless LAN - WiFi) The IEEE has standardized a wireless local-area-network known as IEEE 802.11(a). This method transmits up to 54 Mbps symmetrically between wireless users in relative close proximity (nominally, within a building or complex), nominally of 100 meters or less with one of 3 transmit power levels (assuming 6 dB antenna gain) of 16, 23, or 29 dBm. The system is complex baseband (unlike ADSL, which is real baseband) with $\bar{N} = 64$ (so up to 128 real dimensions). The carrier frequencies are in the 5 GHz range and are exactly (in MHz) $5180 + i(20)$ (16 dBm) , $5260 + i(20)$ (23 dBm), or $5745 + i(20)$ (29 dBm) where $i = 0, 1, 2, 3$.

At baseband, the IEEE 802.11(a) system numbers the tones from -31 to +31 (the 64th carrier at the band edge is not used). The sampling frequency is exactly 20 MHz (complex, so two 20 MHz real converters conceptually) and the carrier spacing is then 20MHz/64=312.5 kHz. The cyclic prefix has 16 complex samples, and thus the symbol consists of 64+16=80 samples. The symbol rate is then 250 kHz (or $T = 4\mu$s, while $T' = 50$ns). The guard period (or maximum length of ISI for zero inter-tone distortion) is 0.8μs.

Of the 63 possible carriers cited, the carrier in the middle of the band carries a fixed phase (not data) for carrier recovery. Also, tones -31 to -27 and 27 to 31 at the band edges are not used to allow analog-filter separation of adjacent carrier bands of different 802.11(a) signals. Furthermore, carriers, -21, -7, 7, and 21 are pilots for symbol-timing recovery. This leaves 48 tones for carrying data. The total bandwidth used is 15.5625 MHz. Data rates with integer numbers of bits per tone will thus be multiples of

$$R = k(1 \text{ bit } /2 \text{ dimensions}) \cdot (48 \text{ tones}) \cdot 250kHz \text{ or } 6 \cdot kMbps. \tag{4.227}$$

Table 4.4 summarizes the options for transmission. The coding rate is a fraction of the actual bits that carry information while the remaining bits are redundant and used for overhead as described in Chapters 10 and 11 (the extra bits help reduce the gap).

Table 4.4 also shows that while all (used) tones carry the same $b_n = b/48$, no tone carries the full number of messages possible in the QAM constellation because of the coding used

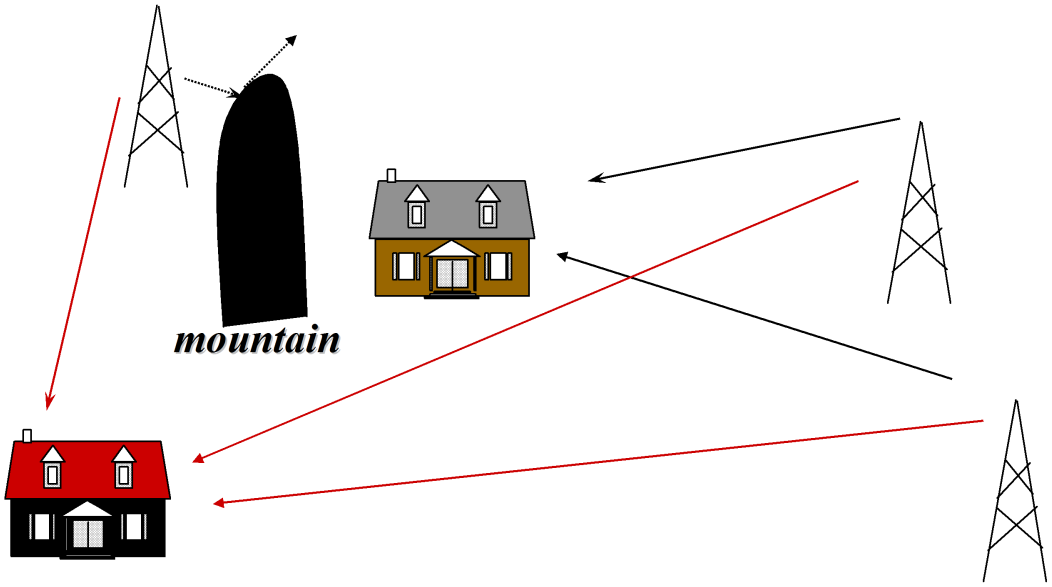


Figure 4.23: Single Frequency Network example for television-broadcast channel partitioning.

– without this coding, OFDM would not be a wise transmission method for wireless. The spectrum mask of the composite analog signal uses flat power spectral density (power divided by 16 MHz) over the used band, and must be 20 dB attenuated at frequencies further than ±11 MHz from carrier, 28dB attenuated at ±20 MHz from carrier, and 40dB attenuated at ± 30 MHz from carrier.

Two new IEEE standards IEEE 802.11(g) and IEEE802.11(n) have extended 802.11(a) to allow carrier frequencies near 2.4 GHz and to allow use of multiple input and output antennas respectively. 802.11(n) speeds can range to a few 100 Mbps. The potential multiple antenna use being suggested by this instructor/author in a 1994 version of this class (then called EE479) and then successfully pursued by several former students. Chapter 5 illustrates a general theory of how to generate the spatial or extra channels in various multiple-input/multiple-output channels.

There is very little adaptive spectra in any of the IEEE 802.11 standards, other than a selection of data rate. The future may see uplink power control used to at least set the power levels of signals to the base station so that sidebands of a nearby signal do not unduly disrupt the main spectrum of a distant signal.

EXAMPLE 4.6.4 (Digital Video Broadcast) Digital terrestrial television broadcast has been very successful in most of the world. This successful system is known generally as Digital Video Broadcast (DVB). A reason for the success elsewhere is the use of a COFDM system with something called SFN (single frequency network), as illustrated in Figure 4.23. SFN basically uses multiple broadcast sites to deliver exactly the same signal to customers so that buildings, structures, and hills blocking one path to customer likely do not block another path. However, when more than one signal reaches the customer, likely with different delays, the effect is equivalent to intersymbol interference or multipath distortion. This delay can be quite large (10's of μs) if transmitters are say 10's of km apart. Over the 6-7.6 MHz wide bands of Television, such delay would overwhelm even the best of Decision Feedback Equalizers as there could be 100's of deep notches in the transmission band, leading to huge length filters and numerical/training problems. However, for COFDM, codes are used to

mitigate any loss of tones at frequencies disabled by such notches. The SFN basically solves one problem (blocking of signal), but forces the use of multicarrier or OFDM.

The DVB system is also complex-baseband and uses conventional TV carrier frequencies (basically above 50 MHz and 7.61 MHz spacing outside US). The DVB system uses either $\bar{N}$=2048 or 8192 tones, of which either 1705 or 6817 are used. The number of pilots used for recovery of symbol rate is 45 or 177, leaving 1650 or 6650 tones available for carrying information. The exact parameters depend on the cyclic prefix length, which itself is programmed by the broadcaster in DVB to be one of 1/4, 1/8, 1/16, or 1/32 of the symbol length. The tone width is fixed at approximately 1.116 kHz (896μs) for 8192 tones and 4464 kHz (224μs) for 2048 tones. The symbol period T thus varies with cyclic prefix and can be any of 1120μs, 1008μs, 952μs, or 924μs for 8192 tones and any of 280μs, 252μs, 238μs, or 231μs for 2048 tones. The sampling rate is fixed at $1/T' \approx 9.142$ MHz ($T' = 10.9375$ ns) for each of two real sampling devices.

Basically, used tones have 4QAM, 16QAM, or 64QAM transmission with various code rates of first code rate (172/204)$\times$ second code rate 1/2, 2/3, 3/4, 5/6, and 7/8, and produces data rates between approximately 4.98 Mbps and 31.67 Mbps (selected by broadcaster, along with number of TV channels carried from 2 to 8. This system at least doubles the number of channels of analog TV, and often increases by a factor of 8, enabling greater choice by consumer and higher revenue by broadcaster. DVB is not used in the USA, which standardized earlier than other countries unfortunately on an inferior VSB-based single-carrier system with lower range.

4.6.3 An alternative view of DMT

If the input sequence is periodic with period N samples, then the corresponding output sequence from an FIR channel is also periodic. The output of the channel may be computed by taking the IDFT of the sequence $Y_n = P_n \cdot X_n$ (where P_n are the samples of the DFT of the channel pulse response samples). When the cyclic prefix is used, however, the input appears periodic as far as the last N samples of any symbol, which are all the receiver processes. DMT thus can also be viewed a circular convolution "trick," although this obscures the optimality and connections to VC.

4.6.4 Noise-Equivalent Channel Interpretation for DMT

The equivalence of an additive ("colored") Gaussian noise channel to an AWGN was established in Section 1.7 and used in Chapters 3 and 4. The colored Gaussian noise generally has power spectral density $\frac{\mathcal{N}_0}{2} R_n(f)$. The channel pulse response of the equivalent AWGN is the pulse response of the original channel adjusted by the inverse square-root autocorrelation of the Gaussian noise:

$$P(f) \rightarrow \frac{P(f)}{\sqrt{S_n(f)}} \quad . \tag{4.228}$$

The receiver uses a "noise-whitening" filtering $1/\sqrt{S_n(f)}$ as a preprocessor to the matched-filter and sampling device. The noise-equivalent viewpoint construes this whitening filter as part of the channel, thus altering the channel pulse response according to (4.228).

For DMT/OFDM, the noise equivalent view discussed earlier leads to a set of subchannels with signal gains $\bar{\mathcal{E}}_{\boldsymbol{x}} \frac{|P(n/T)|^2}{S_n(n/T)}$ and AWGN variance $\frac{\mathcal{N}_0}{2}$, as long as R_{nn} is also circulant. In practice, $S_n(n/T)$ does not correspond to a circulant matrix. As $N \rightarrow \infty$, a stationary noise deviates from a circulant noise by a vanishingly small amount if the autocorrelation coefficients of the noise have finite length, which will occur for reasonable noises. Indeed, the autocorrelation matrix can be close to circulant, even for moderate values of N. A more common view, which avoids the use of the whitening filter, is that each of the subchannels has gain $\bar{\mathcal{E}}_{\boldsymbol{x}} |P(n/T)|^2$ and noise $\frac{\mathcal{N}_0}{2} S_n(n/T)$. The set of SNR's remains the same in either case, but the need for the whitening filter is eliminated in the second viewpoint. The second viewpoint has the minor flaw that the noise variances per subchannel are not completely correct unless the noise $n(t)$ is cyclostationary with period T or NT' is large with respect to the duration of the noise

autocorrelation function. In practice, the values used for N are almost always more than sufficiently large that implementations corresponding to the second viewpoint are ubiquitously found. If a noise truly is very narrow band (and thus has a long autocorrelation function, the small set of frequencies over which it is substantial in energy is typically not used in loading, making its deviation from a more flat noise in effect irrelevent.) Noise power is measured as in Section 4.7.

4.6.5 Toeplitz Distribution

The rigorous mathematical justification for convergence of vector coding to DMT derives from **Toeplitz Distribution** arguments:

> **Theorem 4.6.1 (Toeplitz Distribution)** Given a square Toeplitz matrix R with first row $[r_0\ r_1\ ...\ r_{N-1}]$, which extends to $N = \infty$ by allowing both ends of the first row to grow as $[...\ r_{-2}\ r_{-1}\ r_0\ r_1\ r_2\ ...]$ while maintaining Toeplitz structure, the Fourier Transform is defined by
>
> $$R(f) = \sum_{k=-\infty}^{\infty} r_k e^{-\jmath 2\pi f k} \quad . \tag{4.229}$$
>
> Any N-dimensional Toeplitz submatrix of R along its diagonal has eigenvalues $\lambda_k^{(N)}$. Then for any continuous function $g[\cdot]$
>
> $$\lim_{N\to\infty} \frac{1}{N} \sum_{k=1}^{N} g[\lambda_k^{(n)}] = \int_{-.5}^{.5} g[R(f)]df \quad . \tag{4.230}$$
>
> **No proof given:**

Thus, for the square toeplitz matrix $P\boldsymbol{R_{xx}}P^*$ that has eigenvalues $\mathcal{E}_n|\lambda_n|^2$, and as $N \to \infty$, then

$$\bar{b} = \frac{1}{N} \sum_{n=1}^{N} .5 \log_2 \left(1 + \frac{\mathcal{E}_n|\lambda_n|^2}{\frac{\mathcal{N}_0}{2}} \right) \to \int_{-.5}^{.5} .5 \log_2 \left(\frac{R_x(f)|P(f)|^2}{\frac{\mathcal{N}_0}{2}} \right) \quad .df \tag{4.231}$$

This expression is the mutual information for the discrete-time channel with sampling rate $1/T'$, as derived in Chapter 8. This integral for data rate in bits per dimension maximizes to capacity, $\bar{b} = \bar{c}$, when a water-fill distribution is used for $R_x(f)$,

$$R_x(f) + |P(f)|^2 / \frac{\mathcal{N}_0}{2} = \text{ constant} \quad . \tag{4.232}$$

So, Toeplitz distribution also relates that vector-coding converges (in performance) to MT and to the optimum transmission system when water-fill is used. A caution to note is that the sampling rate $1/T'$ needs to be selected sufficiently high for this discrete-time system to converge to the true optimum of capacity over all sampling rates. When equivalent noise channels are used,

$$|P(f)|^2 \to \frac{|P(f)|^2}{\frac{\mathcal{N}_0}{2} R_n(f)} \tag{4.233}$$

in the above integral.

4.6.6 Gain Normalizer Implementation - The FEQ

For DMT and/or VC, the multichannel normalizer is often called an FEQ (Frequency Domain Equalizer), which is shown in Figure 4.24.

A ZFE is applied to each complex DMT/OFDM subchannel with target X_n for each subchannel. From Section 3.7, with the matched filter absorbed into the equalizer setting W_n, the filter is a single

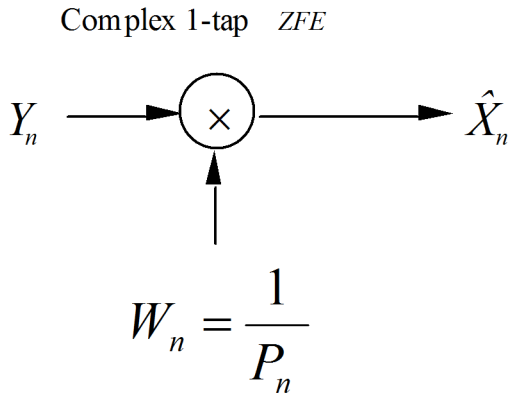


Figure 4.24: The FEQ - frequency-domain equalizer.

complex multiply that adjusts the gain and phase of each subchannel independently to zero the noise-free error between $W_n \cdot Y_n$ and X_n. The FEQ setting for the n^{th} subchannel of DMT is

$$W_n = \frac{1}{P_n} \quad . \tag{4.234}$$

The algorithm that converges to this setting as a function of time was provided earlier in Section 4.3 in Equation (4.134).

Because it is a single tap and cannot change (unbiased) SNR, the FEQ simply scales the DFT outputs so that the DMT/OFDM receiver outputs estimate X_n, and on each subchannel

$$\text{SNR}_{ZFE,n} = \text{SNR}_n \quad . \tag{4.235}$$

With a common code class, $\mathcal{C}$, and constant gap Γ, a common decoder can be reused to decode all subchannels.

4.6.7 Isaksson's Zipper

Often DMT partitioning is used on a bi-directional system. That means both directions of transmission share a common channel. The signals may be separated by echo cancellation (a method of synthesizing from the known transmitted signal a replica of any reflected transmit energy at the opposite-direction signal's co-located receiver and subtracting. The signals may also be separated by assignment of independent tone sets for each direction of transmission, known as frequency-division multiplexing. Both echo cancellation and frequency-division multiplexing have enormous simplification of their signal processing if the DMT signals in the two directions have common sampling rates and symbol rates. Alignment of the symbol boundary at one end is simple because the choice of the transmitted symbol boundary can be made to align exactly with the received DMT signals symbol boundary. However, it would appear that alignment at the other end is then impossible with any realistic channel having delay.

This basic problem is illustrated in Figure 4.25 for DMT systems with $N = 10$, $\nu = 2$, and $\Delta = 3$ (the delay of the channel). LT and NT represent different ends of a transmission line at which each such end has both a transmitter and a receiver. The ds and us appearing in Figure 4.25 refer to downstream (LT to NT) and upstream (NT to LT) transmission. Alignment at the LT leads to misalignment at the NT. Figure 4.26 shows a solution that adds a 6 sample cyclic suffix after the DMT signal, repeating the beginning of the symbol (that is the beginning after the cycli prefix). There are now 6 valid positions for which a receiver FFT to process the recieved signal without interference among tones. One of those 6 positions at the NT downstream is the same position as one of the 6 positions upstream, allowing alignment at the NT and LT simultaneously for the price of an additional 6 wasted samples. The value 6

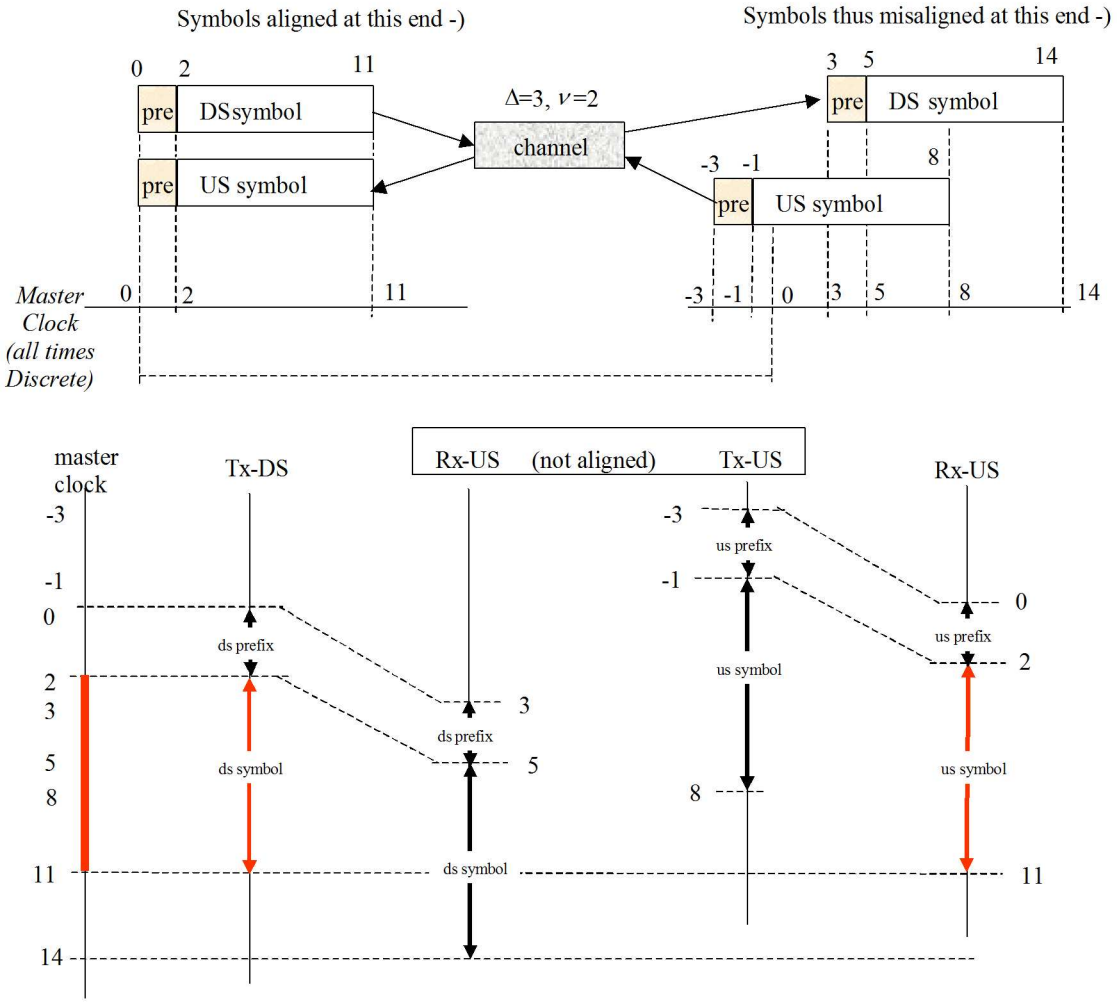


Figure 4.25: Illustration of alignment problem in DMT with only cyclic prefix.

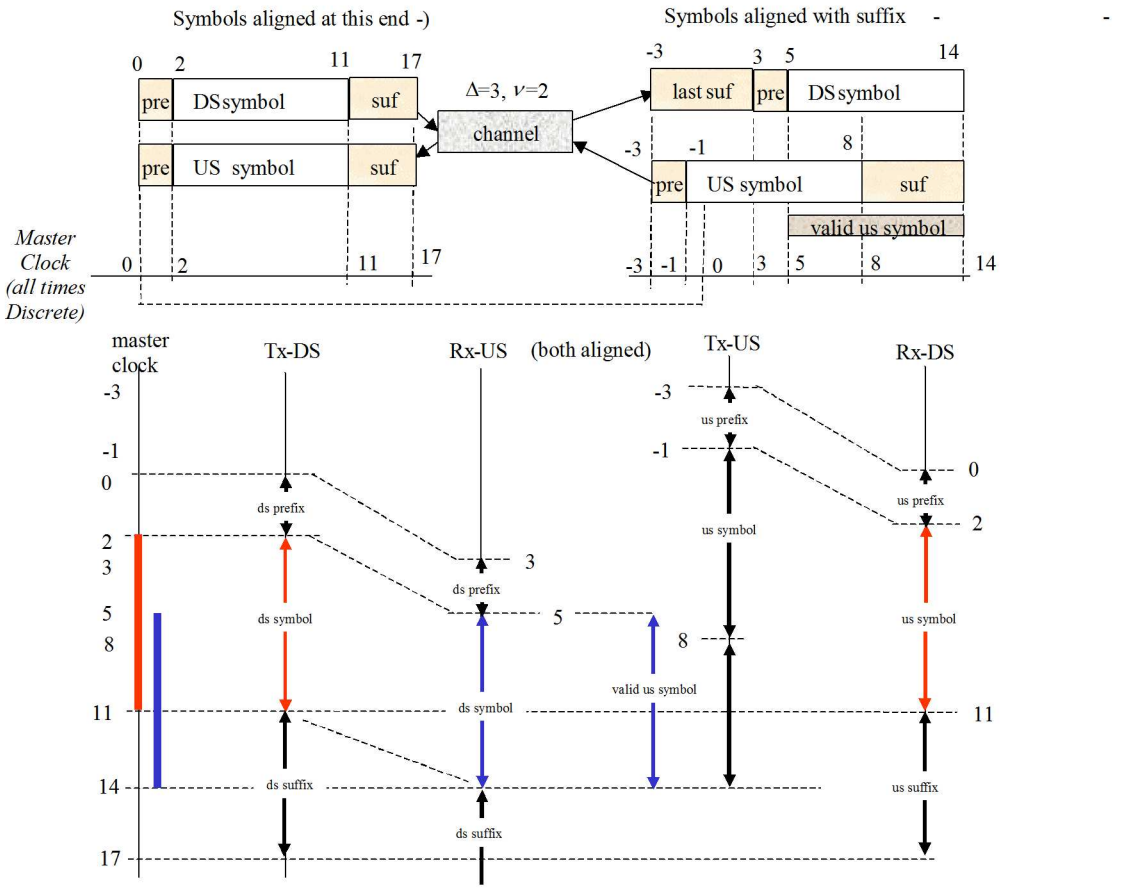


Figure 4.26: Use of cyclic suffix to align DMT symbols at both ends of transmission.

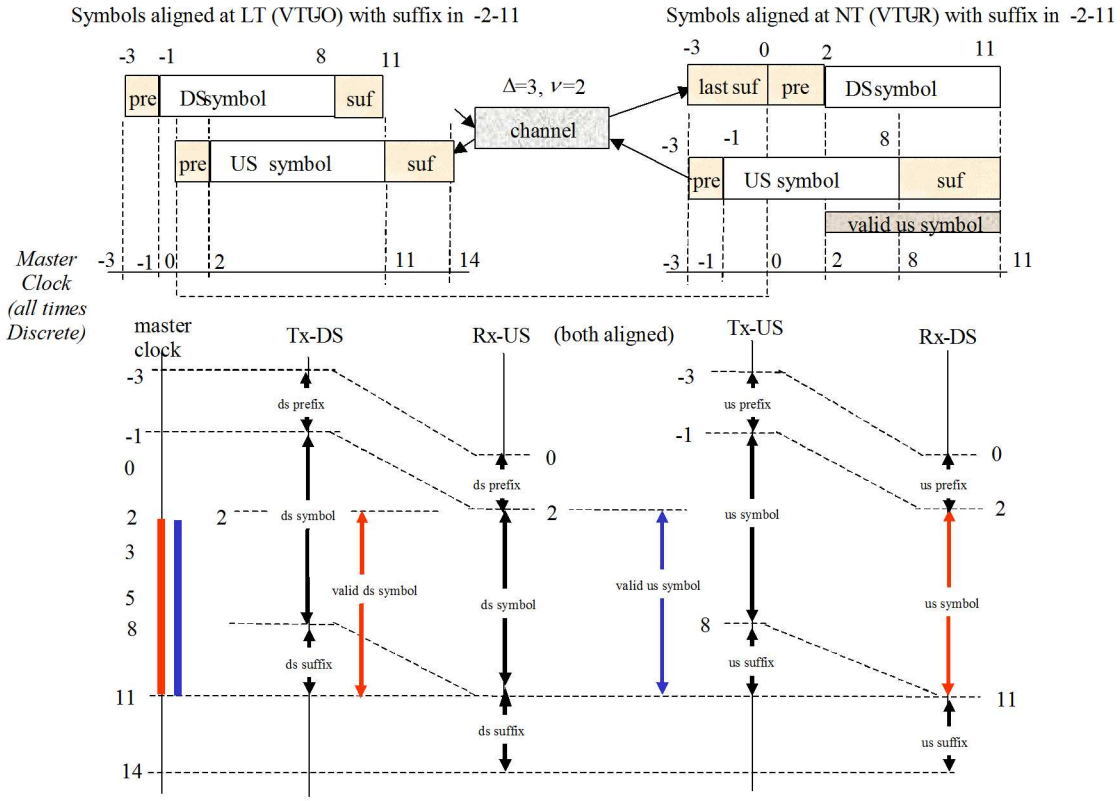


Figure 4.27: Use of both cyclic suffix and timing advance to align DMT signals at both ends of transmission.

is clearly in general $2 \cdot \Delta$. As $N \to \infty$, the cyclic suffix overhead penalty can be made negligible. Figure 4.27 shows a method to reduce the smallest cyclic suffix length to Δ by advancing the LT downstream signal by Δ samples, allowing one of 3 valid downstream positions at the LT to corresponding to one of 3 valid upstream positions, and correspondingly one overlap also at the NT. This is the minimum cyclic suffix length necessary for alignment at both ends simultaneously. This method was introduced in 1997 by Michael Isaksson, then of Telia Research in Sweden and is standardized for use in what is known as VDSL2.

EXAMPLE 4.6.5 (VDSL) VDSL is a recent offspring of ADSL that uses the cyclic suffix and timing advance of Zipper. VDSL, like ADSL, is baseband real, except up to 16 times wider bandwidth than ADSL and intended for use on shorter phone lines where speeds as high as 150 Mbps (more typically 10 Mbps) symmetric can be achieved with VDSL. In VDSL, both upstream and downstream transmission may use up to 4092 tones, so $N = 8192$ with congugate symmetry. The symbol period remains 4000 Hz and the tone spacing is 4.3125 kHz (or optionally 8.625 kHz). The cyclic prefix is now $16 \times 40 = 640$ samples long. The sampling rate is 35.328 MHz. The cyclic prefix period is actually shared with cyclic suffix of minimum-length, determined on a per-line basis adaptively, and DMT symbols are aligned at both ends. The upstream and downstream tones are allocated currently via frequency-division mulitiplexing (above 138 kHz, and can be both directions below that) with frequencies determined as best needed as a function of line, channel, and service-provider desires. The Zipper duplexing leaves no need for concern with complicated analog filters to separate up and down transmissions – this is entirely achieved through digital signal processing and is sometimes also known as "digital duplexing."

4.6.8 Filtered Multitone (FMT)

Filtered Multi-Tone (FMT) approaches use excess bandwidth in the frequency domain (rather than or in addition to excess dimensions in a time-domain guard period) to reduce or eliminate intersymbol interference. There have been many approaches to this "FMT" area under the names of "wavelets (DWMT , W=wavelet)", "filter banks" , "orthogonal filters", "polyphase" by various authors. This subsection pursues a recent approach, specifically given the name FMT, pioneered by Cherubini, Eleftheriou, and Olcer of IBM. This latter FMT approach squarely addresses the issue of the channel impulse response's ($h(t)$'s) usual destruction of orthogonality that was basically ignored in all the other approaches. The basic idea in all approaches though is to find a discrete-time realizable implementation of a filter that basically approximates the GNC-satisfying "brick-wall" frequency-tone characteristic of the sinc(t) function of ideal multi-tone. Essentially, realizable basis functions with very low frequency content outside a main "lobe" of the fourier transform (i.e., the main lobe is centered at the carrier frequency of the tone and there are second, third, fourth, etc. lobes that decay in any realizable approximation to the ideal band-pass filter characteristic of a modualted sinc function). Very low frequency content means the functions after passing through any channel should be almost orthogonal at the output. This "almost" orthogonal unfortunately is the catch in most approaches, which ultimately fail on one channel or another. The FMT approach essentially provides a way of quantifying the amount of miss and working to keep it small on any given channel. There will be a trade-off similar to the trade-off of ν versus N in terms of dimensionality lost in exchange for ensuring realizable and orthogonal basis functions, which will be characterized by essentially the excess bandwidth need in FMT.

The original modulated continuous-time waveform is again

$$x(t) = \sum_{j=-\infty}^{\infty} \sum_{n=0}^{N-1} X_{n,j} \cdot \varphi_n(t - jT) \quad , \tag{4.236}$$

where j is a symbol index. This waveform in discrete construction is sampled at some sampling period T' where $T = (N+\nu)T'$, but there is no cyclic prefix or time-domain guard period. In the FMT approach, ν is sometimes set to zero (unlike DMT). The discrete-time FMT modulator implementation is illustrated in Figure 4.28. The up-arrow device in Figure 4.28 is an interpolator that simply inserts $(N + \nu - 1)$ zeros between successive possibly nonzero values. The down-arrow devices selects every $(N+\nu)^{th}$ sample from its input and ignores all others. The down-arrow devices are presumably each offset successively by T' sample periods from one another. In FMT approaches, each of the discrete-time basis functions is a modulated version of a single low-pass function

$$\varphi_n(t) = \varphi(t) \cdot e^{\jmath \frac{2\pi}{T} \cdot \frac{N+\nu}{N} \cdot nt} \quad . \tag{4.237}$$

Each FMT tone when $\nu > 0$ has excess bandwidth in that the tone spacing is $(1 + \frac{\nu}{N})\frac{1}{T}$ like DMT but without cyclic prefix. Said excess bandwidth can be used to obtain a realizable approximation to an ideal bandpass filter for all tones (subchannels).

A sampling time index is defined according to

$$m = k \cdot (N + \nu) + i \;\; i = 0, ..., N + \nu - 1 \quad , \tag{4.238}$$

where i is an index of the sample offset within a symbol period and k measures as usual symbol instants. The samples $x_m = x(mT')$ of the continuous time waveform are then (with $\varphi_n(mT') = \varphi_{n,m}$)

$$x_m = \sum_{j=-\infty}^{\infty} \sum_{n=0}^{N-1} X_{n,j} \cdot \varphi_{n,m-j\cdot(N+\nu)} \tag{4.239}$$

$$= \sum_{j=-\infty}^{\infty} \sum_{n=0}^{N-1} X_{n,j} \cdot \varphi_{n,k\cdot(N+\nu)+i-j\cdot(N+\nu)} \; \forall i = 0, ..., N + \nu - 1 \tag{4.240}$$

Each of the $N + \nu$ basis filters is also indexed by the sample offset i and essentially decomposes into $\varphi_{n,k\cdot(N+\nu)+i-j\cdot(N+\nu)} = \varphi^{(i)}_{n,k-j}$, each of which can be then implemented at the symbol rate (rather than

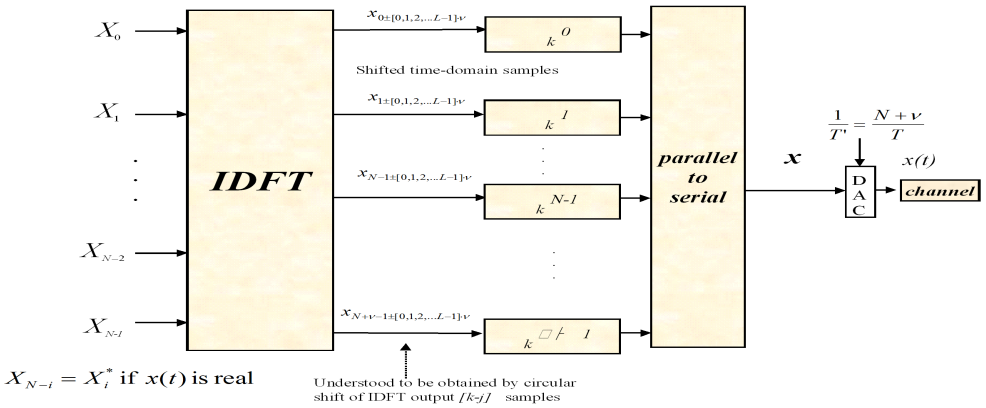


Figure 4.28: FMT partitioning block diagram.

the original single filter at the sampling rate). Further substitution of the expression in (4.237) finds

$$\varphi^{(i)}_{n,k-j} = \varphi^{(i)}_{k-j} \cdot e^{\jmath \frac{2\pi}{N} n \cdot [(k-j)\cdot(N+\nu)+i]} \tag{4.241}$$

$$= \varphi^{(i)}_{k-j} \cdot e^{\jmath \frac{2\pi}{N} n \cdot [(k-j)\cdot(N+\nu)]} \cdot e^{\jmath \frac{2\pi}{N} n \cdot [i]} \quad . \tag{4.242}$$

Substitution of (4.242) into 4.240 provides for all $i = 0, ..., N+\nu-1$

$$x_{k\cdot(N+\nu)+i} = \sum_{j=-\infty}^{\infty} \sum_{n=0}^{N-1} X_{n,j} \cdot \varphi^{(i)}_{k-j} \cdot e^{\jmath \frac{2\pi}{N} n \cdot [(k-j)\cdot(N+\nu)]} \cdot e^{\jmath \frac{2\pi}{N} n \cdot [i]} \tag{4.243}$$

$$= \sum_{j=-\infty}^{\infty} \varphi^{(i)}_{k-j} \cdot \sum_{n=0}^{N-1} X_{n,j} \cdot e^{\jmath \frac{2\pi}{N} n \cdot [(k-j)\cdot(N+\nu)]} \cdot e^{\jmath \frac{2\pi}{N} n \cdot [i]} \quad . \tag{4.244}$$

$$= \sum_{j=-\infty}^{\infty} \varphi^{(i)}_{k-j} \cdot x_{i,j}\left([k-j]\nu\right) \tag{4.245}$$

where $x_{i,j}\left([k-j]\nu\right)$ is the N-point inverse discrete Fourier transform of $X_{n,j} \cdot e^{\jmath \frac{2\pi}{N} n \cdot (k-j)\cdot \nu}$ in the i^{th} sampling offset.[25] Nominally, this IDFT would need to be recomputed for every $(k-j)$ - however one notes that the term $e^{\jmath \frac{2\pi}{N} n \cdot (k-j)\cdot \nu}$ that depends on $(k-j)$ is simply a circular shift of the time-domain output by $(k-j)\nu$ samples. Figure 4.29 illustrates the use of the IDFT (or IFFT for fast computation) and shows that the filter inputs may be circularly shifted versions of previous symbols IDFT outputs when $\nu \neq 0$. If the FIR filter $\varphi^{(i)}_k$ has length $2L+1$, say indexed from time 0 in the middle to $\pm L$, then previous (and future - implement with delay) would use IDFT output time-domain samples that correspond to (circular) shifts of increasing multiples of ν samples as the deviate from center. When $\nu = 0$, no such shifting of the inputs from preceeding and succeeding symbols is necessary. This shifting is simply a mechanization of interpolation to the sampling rate associated with the excess bandwidth that is $(1+\nu/N)$ times faster than with no excess bandwidth.

Clearly if $\nu = 0$ and $\varphi_k = \delta_k$, then FMT and DMT are the same (assuming same $1/T = 1/(NT')$). However, the two systems differ in how excess bandwidth is realized (DMT in time-domain cyclic prefix,

[25] Since $i = 0, ..., N+\nu-1$, the samples of $x_{i,j}\left([k-j]\nu\right)$ will be interpretted as periodic in i with period N.

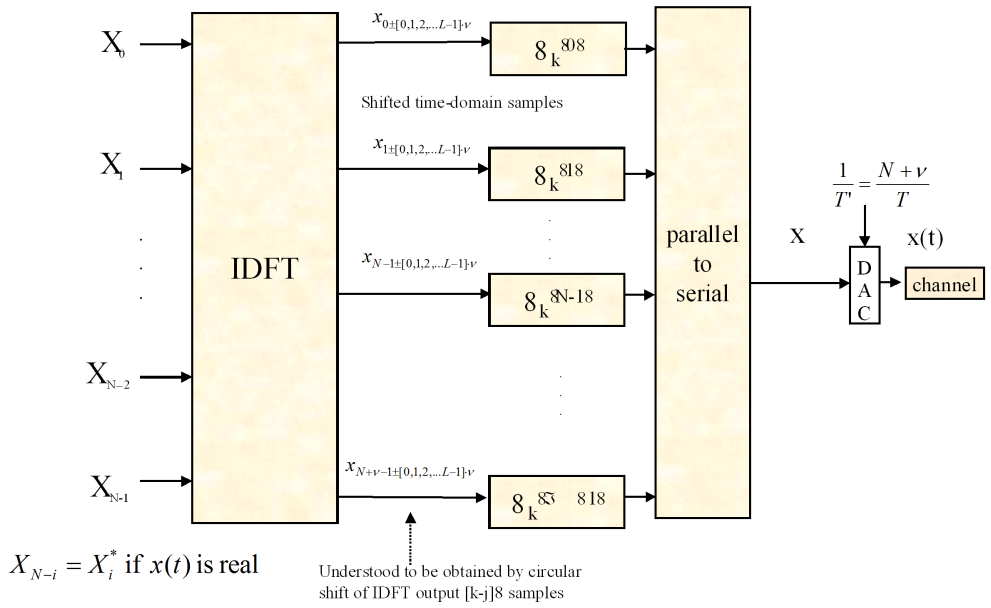


Figure 4.29: FMT partitioning using DFT's and symbol-rate filters.

parameter	DMT	FMT
Δf	$\frac{1+\frac{\nu}{N}}{T}$	$\frac{1+\frac{\nu}{N}}{T}$
symbol length	$N+\nu$ samples	$N+\nu$ samples
cyc. pre. length	ν	0
lost freq dimensions	0	ν
excess bandwidth α	$\frac{\nu}{N}$	$\frac{\nu}{N}$
φ_k	δ_k	lowpass filter
delay	$N+\nu$	integer $\cdot(N+\nu)$
implementation	N-point FFT	N-point FFT plus symbol-rate filter
performance	likes small-ν channels	likes sharp bandpass channels

Table 4.5: Comparison of DMT and FMT

FMT in essentially wider tones than would otherwise be necessary without cyclic prefixes by same ratio ν/N). FMT also allows a shaping filter to be used to try to approximately synthesize ideal-multitone bandpass subchannels. DMT and FMT are compared in Table 4.5.

Any FMT system will have at least some residual intersymbol interference and interchannel interference that is hopefully minimized by good choice of φ_k. However, equalizers may be necessary on each subchannel and possibly between a few adjacent subchannels as shown in Figure 4.30. Usually, the functions have such low out-of-band energy that only intersymbol interference on each subchannel are of concern, so the receiver reduces to the FFT followed by a bank of symbol-rate DFE's as shown in Figure 4.30.

Filter Selection for FMT

There are an enormous number of possible choices for the function $\varphi(t)$ in FMT. Some authors like to choose wavelet functions that have very low sidelobes but these often introduce intersymbol interference while eliminating crosstalk. While wavelets may apply nicely in other areas of signal processing, they are largely ineffective in data transmission because they ignore the channel. Thus, pursuit of a large number of wavelet functions is probably futile and a digression from this text.

Instead, IBM has suggested a couple of functions for DSL applications that appear promising and

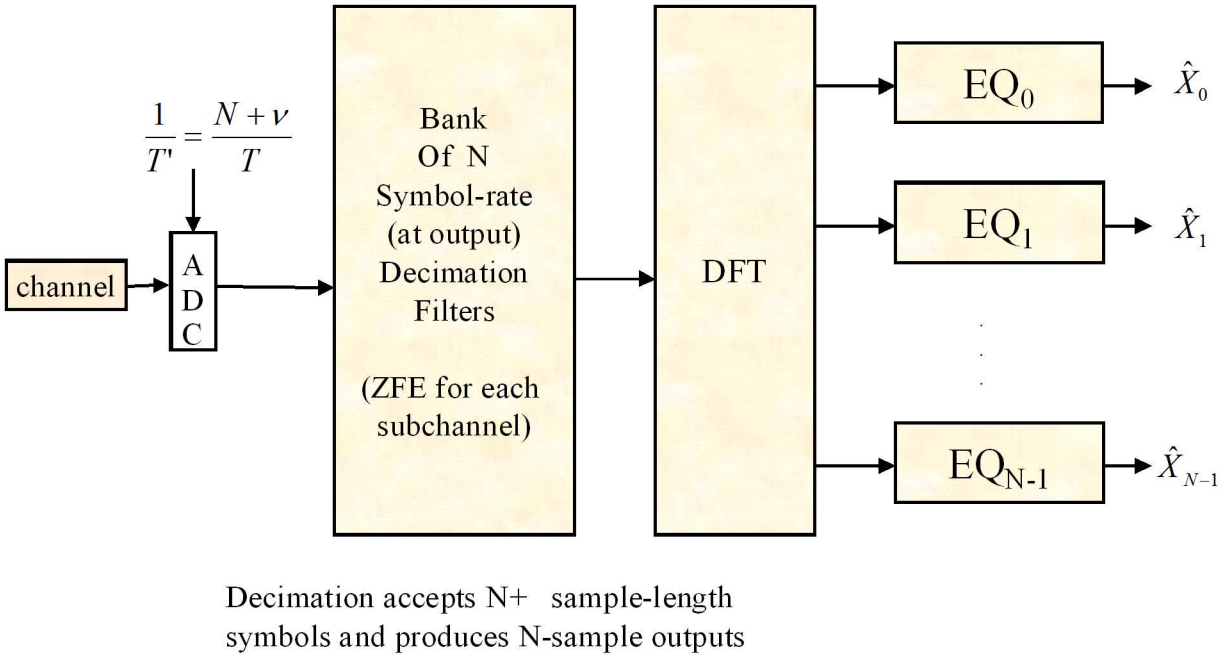


Figure 4.30: FMT receiver block diagram assuming negligible or no inter-channel interference.

were designed with cognizance of the class of channels that are seen in those applications:

EXAMPLE 4.6.6 (Cherubini's Coaster) Cherubini's coaster functions use no excess bandwidth and have $\nu = 0$. The coaster is actually a set of functions parameterized by $0 < \rho < 1$ and the first-order IIR lowpass filter

$$\Phi(D) = \sqrt{\frac{1+\rho}{2}} \cdot \frac{1+D}{1+\rho \cdot D} \quad . \tag{4.246}$$

As $\rho \to 1$, the functions become very sharp and the impulse response φ_k is very long. Figure 4.31 illustrates these functions for a $\rho = .1$. The securves exhibit rapid decay with frequency, basically ensuring in practice that the basis functions remain orthogonal at the channel output. Also, there will be intersymbol interference, but as $\rho \to 1$, this ISI becomes smaller and smaller and the functions approach the ideal multitone functions (except this IIR filter is stable and realizable). A DFE is typically used in the receiver with Cherubini's coaster. The hope is that there is very little intersymbol or inter-channel interference and that N can be reduced (because the issue of ν/N excess bandwidth being small is inpertinent when $\nu = 0$). The discrete-time response is given by

$$\varphi_k = \sqrt{\frac{1+\rho}{2}} \cdot \left[\delta_k + (1-\rho)(-\rho)^{k-1} \cdot u_{k-1}\right] \quad . \tag{4.247}$$

In this figure, zero-order-hold (flat interpolation) is used between T'-spaced samples to provide a full Fourier transform of a continuous signal (such zero-order hold being characteristic of the behavior of a DAC). This effect causes the frequency rippling in the adjacent bands that is shown, but is quite low with respect to what would occur if $\varphi_k = \delta_k$ where this rippling would follow a $T \cdot \text{sinc}(fT)$ shaping with first "side lobe" down only 13.5 dB from the peak as in Figure 4.39 of Section 4.8 instead of the 55 dB shown with Cherubini's Coaster.

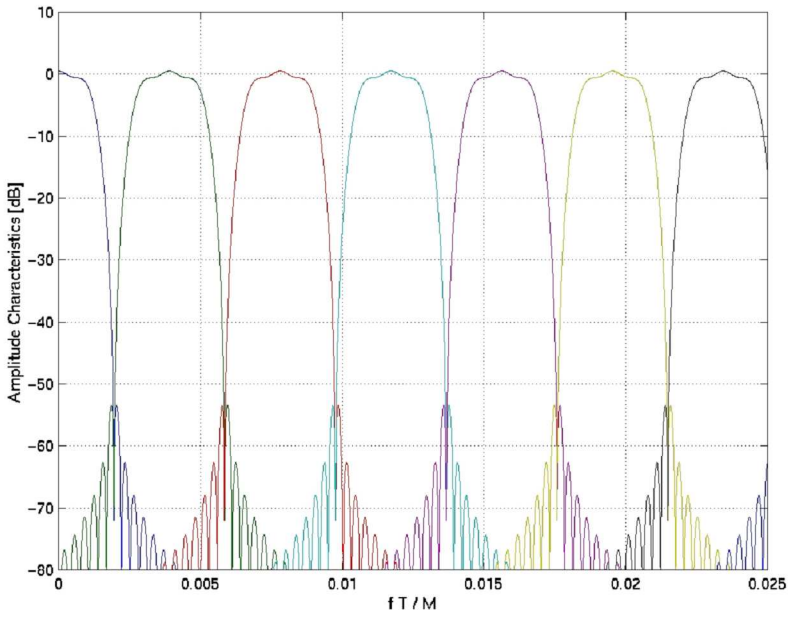


Figure 4.31: Cherubini's Coaster basis functions in frequency domain with $\rho = .1$.

The second example uses excess bandwidth and raised-cosine shaping.

EXAMPLE 4.6.7 (Raised-Cosine with $\nu > 0$) Raised cosine functions with excess bandwisth (ν/N) can also be used. In this case the function $\varphi_k^{(i+\nu)}$ can be simplified from the square-root raised cosine in Section 3.2 to

$$\varphi_k^{(i)} = \frac{4\sqrt{1+\frac{N}{\nu}}}{\pi\sqrt{\nu T'}} \cdot \frac{\cos\left(\pi\left[k+\frac{i}{N}\right]\right) + \frac{1+\frac{N}{\nu}}{4\cdot\left[k+\frac{i}{N}\right]} \cdot \sin\left(\pi\left[k+\frac{i}{N}\right] \cdot \frac{N-\nu}{N+\nu}\right)}{1-16\cdot\left[k+\frac{i}{N}\right]^2} \quad . \tag{4.248}$$

This function with sufficiently large number of taps in FIR approximation (for given excess bandwidth, smaller ν/N means longer FIR filter in approximation) can be made to have sufficiently low out-of-band energy and clearly avoids intersymbol interference if the tones are sufficiently narrow.

4.7 Multichannel Channel Identification

Generally, channel identification methods measure the channel pulse response and noise power spectral density, or their equivalents (that is direct measurement of SNR's without measuring pulse response and noise separately). Multi-channel channel identification directly estimates signal and noise parameters for each of the subchannels. Many channel identification methods exist, but only a DMT-specific method for channel identification appears in this section. Other methods are discussed in Chapter 7. The method of this section separately identifies first the pulse-response DFT values at the subchannel center frequencies used by a DMT system and then second the noise-sample variances at these same frequencies. Channel identification then finishes by computing the SNR's from the results of gain and noise estimation. The measurement of the pulse-response DFT values is called **gain estimation**[26], while the measurement of the noise variances is called **noise estimation**. This section's particular method is in common use because it reuses the same hardware/software as steady-state transmission with DMT; however, it is neither optimum nor necessarily a good choice in terms of performance for limited data observation. It is easy to implement. Fortunately, a lengthy training interval is often permitted for DMT-transmission applications, and this method will become very accurate with sufficient observation. The presented method is usually not appropriate for wireless OFDM systems.

The channel noise is an undesired disturbance in gain estimation, and this noise needs to be averaged in determining subchannel gains as in Subsection 4.3.5. When gain estimation is complete, the estimated (noiseless) channel output can be subtracted from the actual channel output to form an estimate of the channel noise. The spectrum of this channel noise can then be estimated as in Section 4.7.2. In gain estimation, care is taken to ensure that any residual gain-estimation error is sufficiently small that it does not significantly corrupt the noise-spectral-estimation process. Such small error can require large dynamic range in estimating the subchannel gains.

Equation (4.140) shows that the channel SNR g_n can be computed from the known subchannel input distance, d_n, the known multichannel normalized output distance d, and the noise estimate $\tilde{\sigma}_n^2$. During training, it is possible to send the same constellation on all subchannels, thus essentially allowing $d_n = d$, and thus making

$$g_n = \frac{1}{\tilde{\sigma}_n^2} \quad . \tag{4.249}$$

Two questions were not addressed, however, by Subsection **??**, which are invidually the gain constants μ and μ' for the updating loops for the multichannel normalizer W_n in (4.134) and for the noise-variance estimation in (4.138), respectively. These values will be provided shortly, but a more direct consideration of computation of g_n will be considered for the special case of training first. The choice of μ is an example of the general problem of gain estimation in Subsection 4.7.1, which is the estimation of the subchannel signal gain. The choice of the second gain constant μ' is an example of the general problem of noise estimation in Subsection 4.7.2, which is the estimation of the unscaled subchannel noise.

4.7.1 Gain Estimation

Figure 4.32 illustrates gain estimation. The known training sequence, x_k, is periodic with period N equal to or greater than the number of coefficients in the unknown channel pulse response p_k ("periodic" can mean use of the cyclic prefix of Section 4.4). The channel output sequence is

$$y_k = x_k * p_k + u_k \tag{4.250}$$

where u_k is an additive noise signal assumed to be uncorrelated with x_k. The channel distortion sequence is denoted by u_k, instead of n_k, to avoid confusion of the frequency index n with the noise sequence and because practical multichannel implementations may have some residual signal elements in the distortion sequence u_k (even though independence from x_k is assumed for simplification of mathematics).

Gain estimation constructs an estimate of the channel pulse response, $\hat{p}_k$, by minimizing the mean square of the error

$$e_k = y_k - \hat{p}_k * x_k \quad . \tag{4.251}$$

[26] This gain is usually a complex gain meaning that both subchannel gain and phase information are estimated.

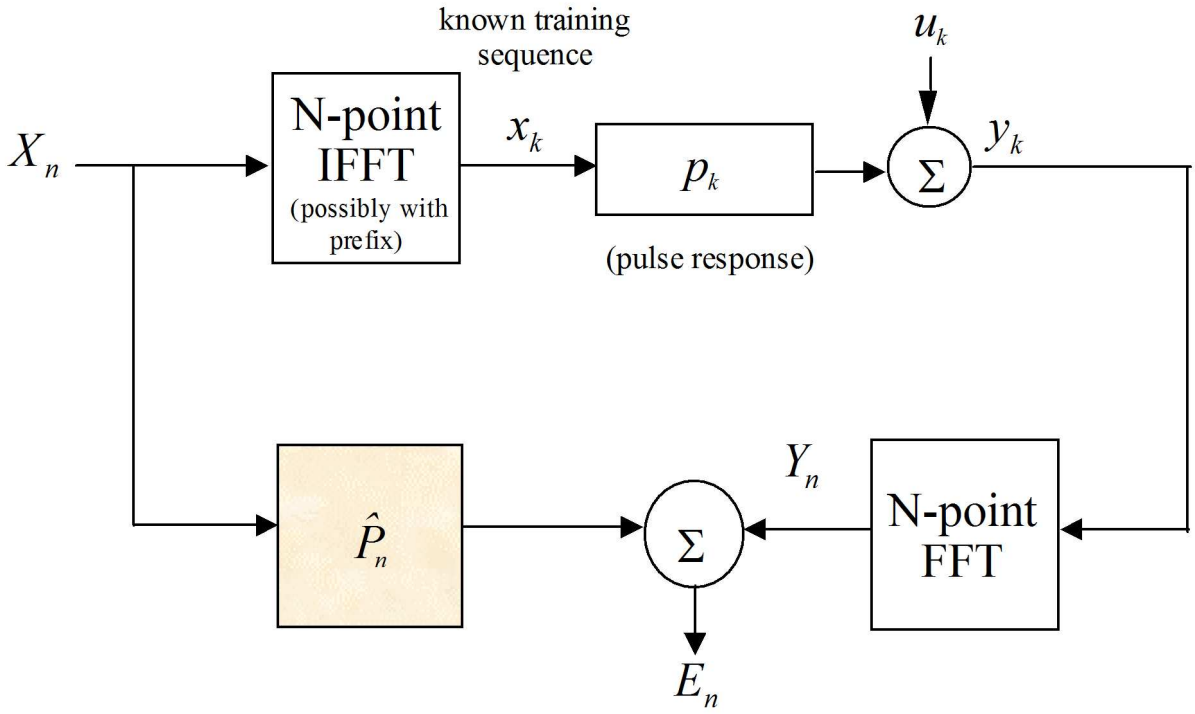


Figure 4.32: Channel-identification basics and terminology.

Ideally, $\hat{p}_k = p_k$.

The Discrete Fourier Transform, DFT, of x_k is

$$X_n = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} x_k \cdot e^{-\jmath \frac{2\pi}{N} kn} \quad . \tag{4.252}$$

When x_k is periodic[27] with period N samples, the DFT samples X_n are also periodic with period N and constitute a complete representation for the periodic time sequence x_k over all time. The N-point DFT's of y_k, u_k, $\hat{p}_k$, p_k, and e_k are Y_n, U_n, $\hat{P}_n$, P_n, and E_n, respectively.

Since x_k is periodic, then

$$Y_n = P_n \cdot X_n + U_n \quad . \tag{4.253}$$

The corresponding frequency-domain error is then

$$E_n = Y_n - \hat{P}_n \cdot X_n \quad n = 0, ..., N \quad . \tag{4.254}$$

(Recall that subchannels 0 and N are one-dimensional in this development.) A vector of samples will contain one period of samples of any of the periodic sequences in question, for instance

$$\boldsymbol{x} = \begin{bmatrix} x_{N-1} \\ \vdots \\ x_1 \\ x_0 \end{bmatrix} \quad . \tag{4.255}$$

Then the relations

$$\boldsymbol{X} = Q^* \boldsymbol{x} \quad \boldsymbol{x} = Q\boldsymbol{X} \tag{4.256}$$

[27] The periodicity can also be emulated by a cyclic prefix of length ν samples at the beginning of each transmitted block.

follow easily, where Q^* is the matrix describing the DFT ($q_{N-1-i,N-1-k} = e^{-\jmath\frac{2\pi}{N}ki}$) and $QQ^* = Q^*Q = I$. Similarly, $\boldsymbol{y}$, $\boldsymbol{Y}$, $\boldsymbol{e}$, $\boldsymbol{E}$, $\boldsymbol{p}$, $\boldsymbol{P}$, $\hat{\boldsymbol{p}}$, $\hat{\boldsymbol{P}}$, $\boldsymbol{u}$ and $\boldsymbol{U}$ as corresponding time- and frequency-domain vectors. For the case of the non periodic sequences u_k and thus y_k and e_k, a block time index l differentiates between different channel-output blocks (symbols) of N samples, i.e. $\boldsymbol{y}_l$.

The MSE for any estimate $\boldsymbol{p}$ is then

$$\begin{aligned} \text{MSE} &= E\left\{|e_k|^2\right\} && (4.257)\\ &= \frac{1}{N}E\left\{\|\boldsymbol{e}\|^2\right\} = \frac{1}{N}\sum_{k=0}^{N-1} E\left\{|e_k|^2\right\} && (4.258)\\ &= \frac{1}{N}E\left\{\|\boldsymbol{E}\|^2\right\} = \frac{1}{N}\sum_{n=0}^{N} E\left\{|E_n|^2\right\} \quad . && (4.259) \end{aligned}$$

The tap error is $\delta_k \triangleq p_k - \hat{p}_k$ in the time domain and $\Delta_n \triangleq P_n - \hat{P}_n$ in the frequency domain. The autocorrelation sequence for any discrete-time sequence x_k is defined by

$$r_{xx,k} = E\left\{x_m x^*_{m-k}\right\} \tag{4.260}$$

with corresponding discrete power spectrum

$$R_{xx,n} = Q\ r_{xx,k} \quad . \tag{4.261}$$

The norm tap error (NTE) is defined as

$$\begin{aligned} \text{NTE} &\triangleq E\left\{\|\boldsymbol{p} - \hat{\boldsymbol{p}}\|^2\right\} = E\left\{\|\boldsymbol{\delta}\|^2\right\} && (4.262)\\ &= E\left\{\|\boldsymbol{P} - \hat{\boldsymbol{P}}\|^2\right\} = E\left\{\|\boldsymbol{\Delta}\|^2\right\} && (4.263)\\ &= \sum_{n=0}^{N} |P_n - \hat{P}_n|^2 \quad . && (4.264) \end{aligned}$$

When the estimate $\hat{\boldsymbol{p}}$ equals the channel pulse response $\boldsymbol{p}$, then

$$e_k = u_k \tag{4.265}$$

and

$$r_{ee,k} \triangleq E\left[e_m e^*_{m-k}\right] = r_{uu,k} \ \ \forall\, k = 0, ..., N-1 \quad , \tag{4.266}$$

where $r_{ee,k}$ is the autocorrelation function of e_k, and $r_{uu,k}$ is the autocorrelation function of u_k. The MSE is then, ideally,

$$\text{MSE} = r_{ee,0} = \sigma_u^2 \quad , \tag{4.267}$$

the mean-square value of the channel noise samples. $\sigma_u^2 = \frac{\mathcal{N}_0}{2}$ for the AWGN channel. In practice, $\hat{p}_k \neq p_k$ and the excess MSE is

$$\text{EMSE} \triangleq \text{MSE} - \sigma_u^2 \quad . \tag{4.268}$$

The overall channel SNR is

$$\text{SNR} = \frac{r_{xx,0}\|\boldsymbol{p}\|^2}{r_{uu,0}} \tag{4.269}$$

and the SNR of the n^{th} channel is

$$\text{SNR}_n = \frac{R_{xx,n}|P_n|^2}{R_{uu,n}} \quad . \tag{4.270}$$

The gain-estimation SNR is a measure of the true channel-output signal power on any subchannel to the excess MSE in that same channel (which is caused by $\hat{\boldsymbol{p}} \neq \boldsymbol{p}$), and is

$$\gamma_n \triangleq \frac{R_{xx,n}|P_n|^2}{R_{ee,n} - R_{uu,n}} \quad . \tag{4.271}$$

EXAMPLE 4.7.1 (Accuracy) γ_n should be 30-60 dB for good gain estimation. To understand this requirement on γ_n, consider the case on a subchannel in DMT with $\text{SNR}_n = \frac{R_{xx,n}|P_n|^2}{R_{uu,n}} = 44$ dB, permitting 2048 QAM to be transmitted on this subchannel. The noise is then 44 dB below the signal on this subchannel.

To estimate this noise power to within .1 dB of its true value, any residual gain estimation error on this subchannel, $R_{ee,n} - R_{uu,n}$, should be well below this mean-square noise level. That is

$$R_{ee,n} < 10^{.1/10} \cdot R_{uu,n} \approx (1 + 1/40) \cdot R_{uu,n} \tag{4.272}$$

Then,

$$\gamma_n = \frac{R_{xx,n}|P_n|^2}{R_{ee,n} - R_{uu,n}} = \frac{\text{SNR}_n}{(1/40)} = 40 \cdot \text{SNR} \tag{4.273}$$

is then 60 dB (= 44+16 dB). Similarly, on a comparatively weak subchannel for which we have SNR_i=14 dB for 4-point QAM, one would need γ_n of at least 30 dB (= 14+16).

In general, for x dB accuracy

$$\gamma_n \geq \left(10^{x/10} - 1\right)^{-1} \cdot \text{SNR}_n \quad . \tag{4.274}$$

An Accurate Gain Estimation method

A particularly accurate method for gain estimation uses a cyclically prefixed training sequence x_k with period, N, equal to or slightly longer than ν, the length of the equalized channel pulse response. Typically, N is the same as in the DMT transmission system. The receiver measures (and possibly averages over the last L of $L + 1$ cycles) the corresponding channel output, and then divides the DFT of the channel output by the DFT of the known training sequence.

The channel estimate in the frequency domain is

$$\hat{P}_n = \frac{1}{L} \sum_{l=1}^{L} \frac{Y_{l,n}}{X_{l,n}} \quad . \tag{4.275}$$

Performance Analysis The output of the n^{th} subchannel on the l^{th} symbol produced by the receiver DFT is $Y_{l,n} = P_n X_{l,n} + U_{l,n}$. The estimated value for $\hat{P}_n$ is

$$\hat{P}_n = P_n + \sum_{l=1}^{L} \frac{U_{l,n}}{L \cdot X_{l,n}} \quad , \tag{4.276}$$

so

$$\Delta_n = -\sum_{l=1}^{L} \frac{U_{l,n}}{L \cdot X_{l,n}} \quad . \tag{4.277}$$

By selecting the training sequence, $X_{l,n}$, with constant magnitude, the training sequence becomes $X_n = |X|e^{\jmath\theta_{l,n}}$.

The error signal on this subchannel after $\hat{P}_n$ has been computed is

$$E_n = Y_n - \hat{P}_n X_n = \Delta_n X_n + U_n \tag{4.278}$$

$$= U_n + \frac{1}{L} \sum_{l=1}^{L} U_{l,n} e^{\jmath(\theta_n - \theta_{l,n})} \quad . \tag{4.279}$$

The phase term on the noise will not contribute to its mean-square value. U_n corresponds to a block that is not used in training, so that U_n and each of $U_{l,n}$ are independent, and

$$E\left\{|E_n|^2\right\} = R_{ee,n} = R_{uu,n} + \frac{1}{L} R_{uu,n} = (1 + \frac{1}{L}) R_{uu,n} \quad . \tag{4.280}$$

The excess MSE on the i^{th} subchannel then has variance $(1/\cdot L)R_{uu,n}$ that is reduced by a factor equal to the the number of symbol lengths that fit into the entire training interval length, with respect to the mean-square noise in that subchannel. Thus the gain-estimation SNR is

$$\gamma_n = L \cdot \frac{R_{xx,n}|P_n|^2}{R_{uu,n}} = L \cdot \text{SNR}_n \quad , \tag{4.281}$$

which means that the performance of the recommended gain estimation method improves linearly with L on all subchannels. Thus, while noise may be relatively higher (low SNR_n) on some subchannels than on others, the extra noise caused by misestimation is a constant factor below the relative signal power on that same subchannel. In general then for x dB of accuracy

$$L \geq \left(10^{x/10} - 1\right)^{-1} \quad . \tag{4.282}$$

For an improvement in gamma 16 dB, as determined earlier in an example provides .1 dB error or less no matter what the subchannel SNR_n, gain estimation requires $L = 40$ symbols of training.

For instance, the previous ADSL and VDSL examples of Section 4.3 used symbol periods of 250μs. Thus, gain estimation to within .1 dB accuracy is possible (on average) within 10ms using this simple method.

Windowing When the FFT size is much larger than the number of significant channel taps, as is usually the case, it is possible to significantly shorten the training period. This is accomplished by transforming (DFT) the final channel estimate to the time domain, windowing to $\nu+1$ samples and then returning to the frequency domain. Essentially, all the noise on sample times $\nu + 2...N$ is thus removed without disturbing the channel estimate. The noise on all estimates, or equivalently the residual error in the SNR, improves by the factor N/ν with windowing. For the ADSL/VDSL example again, if the channel were limited to 32 nonzero successive samples in duration, then windowing would provide a factor of (32/512) reduction or 12 dB of improvement, possibly allowing just a few DMT symbols to have an (on average) accuracy of .1 dB, and accurate gain estimation in a few ms.

Suggested Training Sequence

One training sequence for the gain estimation phase is a random or white sequence[28] L blocks are used to create $\hat{P}_n$, which is computed recursively for each of the subchannels, one symbol at a time, by accumulation, and then divided by L. The L blocks can be periodic repeats or cyclic prefix can be used to make the training sequence appear periodic. For reasons of small unidentified channel aberrations, like small nonlinearities caused by drivers, converters, amplifiers, it is better to use a cyclic prefix because any such aberrations will appear in the noise estimation to be subsequently described.

In complex baseband channels a "chirp" training sequence sometimes appears convenient

$$x_k = e^{j\frac{2\pi}{N}k^2} \quad . \tag{4.283}$$

This chirp has a the theoretical minimum peak-to-average (one-dimensional) power ratio of 2 and is a "white sequence," with $r_{xx,k} = \delta_k$, with δ_k being the Kronecker delta (not the norm tap error). While "white" is good for identifying an unknown channel, a chirp's low PAR can leave certain small nonlinear "noise" distortion unidentified, which ultimately would cause the identified SNR_n's to be too optimistic.

The value for μ in the FEQ The EMSE for the first-order zero-forcing update algorithm in Equation (4.134) can be found with some algebra to be

$$\text{EMSE} = \frac{\mu \cdot \mathcal{E}_n}{2 - \mu \cdot \mathcal{E}_n} R_{uu,n} \quad . \tag{4.284}$$

[28] Say the PRBS of Chapter 10 with polynomial $1 + D + D^{16}$) with $b_n = 2$ on channels $n = 1, ..., N-1$ and $b_n = 1$ on the DC and Nyquist subchannels. One needs to ensure that the seed for the PRBS is such that no peaks occur that would exceed the dynamic range of channel elements during gain estimation. Such peaks are usually compensated by a variety of other mechanisms when they occur in normal transmission (rarely) and would cause unnecessarily conservative loading if their effects were allowed to dominate measured SNRs.

Thus, for the multichannel normalizer to maintain the same accuracy as the initial gain estimation in Equation (4.280)

$$\frac{1}{L} = \frac{\mu \cdot \mathcal{E}_n}{2 - \mu \cdot \mathcal{E}_n} \tag{4.285}$$

or

$$\mu = \frac{2}{(L+1) \cdot \mathcal{E}_n} \quad , \tag{4.286}$$

so for $L = 40$ and $\mathcal{E}_n = 1$, the value would be $\mu = 1/42$.

4.7.2 Noise Spectral Estimation

It is also necessary to estimate the discrete noise power spectrum[29], or equivalently the mean-square noise at each of the demodulator DFT outputs, computation of the SNR's necessary for the bit-distribution. Noise estimation computes a spectral estimate of the error sequence, E_n, that remains after the channel response has been removed. The training sequence used for gain estimation is usually continued for noise estimation.

Analysis of Variance

The variance of L samples of noise on the n^{th} subchannel is:

$$\hat{\sigma_n^2} = \frac{1}{L} \sum_{l=1}^{L} |E_{l,n}|^2 \quad . \tag{4.287}$$

For stationary noise on any tone, the mean of this expression is the variance of the (zero-mean) noise. The "variance of this variance estimate" is computed as

$$\text{var}\left(\hat{\sigma_n^2}\right) = \frac{1}{L^2}\left(3 \cdot L \cdot \sigma_n^4 - L \cdot (\sigma_n^2)^2\right) = \frac{2}{L}\sigma_n^4 \tag{4.288}$$

for Gaussian noise. Thus the excess noise in the estimate of the noise is therefore $\sqrt{2/L} \cdot \sigma_n^2$ and the estimate noise decreases with the square-root of the number of averaged noise terms. To make this extra noise .1 dB or less, we need $L = 3200$. Recalling that the actual computed value for the noise estimate will have a standard deviation on average that leads to less than .1 dB error, there is still a 10% chance that the statistics of the noise will lead to deviation greater than 1 standard deviation (assuming Gaussian distribution, or approximating the more exact Chi-square with a Gaussian). Thus, sometimes a yet larger value of L is used. 98% confidence occurs at $L = 12,800$ and 99.9% confidence at $L = 28,800$. If the equivalent noise response is short, windowing could again be used to reduce this training time.

> **EXAMPLE 4.7.2 (ADSL)** The ADSL example with symbol rate of 4000 Hz has been discussed previously. The training segment for SNR estimation in G.992.1 initially is called Medly and consists of 16000 symbols (with QPSK on all subchannels) for 4 seconds of training. This exceeds the amount given as a guideline above - this extra time is provided with the cognizance that the standard deviation of the measured noise and channel should lead to no more than .1 dB error with a smaller training interval, but that there are still statistical deviations with real channels. Essentially, with this increased value, about 99% of the time the computed value will deviate by this amount of .1 dB or less. The gain estimation error clearly can be driven to a negligible value by picking L to a few hundred without significantly affecting training time. (The long training interval also provides time for SNR computation and loading algorithms after gains and noises are estimated.) Windowing may not be helpful in DSL noise estimation because the equivalent noise response can be long and noise can vary significantly with frequency.

[29] Clearly the noise can not have been whitened if it is unknown and so this step precedes any noise whitening that might occur with later data transmission.

The value for μ' in the FEQ The standard deviation for the first-order noise update in the FEQ can be found to be

$$\text{std}(\tilde{\tilde{\sigma}}_n^2) = \sqrt{\frac{\mu'}{2-\mu'}}\tilde{\sigma}_n^2 \quad . \tag{4.289}$$

Thus, to maintain a constant noise accuracy, the step size should satisfy

$$\frac{2}{L} = \frac{\mu'}{2-\mu'} \tag{4.290}$$

or

$$\mu' = \frac{4}{L+2} \tag{4.291}$$

so when $L = 6400$, then $\mu' \approx 6.25 \times 10^{-4}$. In reality, the designer might want high confidence (not just an average confidence) that the accuracy is .1 dB, in which case a μ of perhaps 10^{-4} would suggest the chance of missing would be the same as noise occuring with 4 times the standard deviation (pretty small in Gaussian and Xi-square distributions).

From Section 4.3.5, gain normalizers in general and thus the FEQ update the noise using the proportionality

$$\frac{1}{g_n} \propto \tilde{\sigma}_n^2 \quad . \tag{4.292}$$

Effectively, the normalizer then tracks the ratio

$$\text{gain} = \frac{\tilde{\sigma}_{n,current}^2}{\tilde{\sigma}_{init,n}^2} \quad . \tag{4.293}$$

While it may take a long averaging time to introduce .1 dB accuracy, clearly an abrupt change in noise of several dB or more on any (or many) single tones can be detected rapidly.

A simple mechanism would be to introduce a second noise loop that follows Equation (4.134) with a μ'' replacing μ' where $\mu'' >> \mu'$. This faster update the noise variance estimate would be much less accurate. However, to discern whether the noise has changed rapidly, it provides an early indication. Basically, this faster swapping exploits the fact that the bit distribution is near converged in steady-state operation just before a large noise change on some tones might occur. Thus, the bit-swap controller is looking only for a gross change to initiate more rapid swapping. The following example illustrates.

> **EXAMPLE 4.7.3 (ADSL swap tracking speed)** As an example, suppose 1 dB error (so a factor of 1+1/10) of accuracy is sufficient to detect whether a noise change of several dB has occurred. In this case
>
> $$\frac{2}{\sqrt{L}} < \frac{1}{10} \tag{4.294}$$
>
> which means an interval of 200 symbols would be sufficient. With ADSL's symbol rate of 4000 Hz, this corresponds to 50ms. This would correspond to 20 bits swapped per second if a large noise event were detected, much faster than the usual situation of perhaps a bit per second or less. As long as FEQ gains and bit tables are maintained (scaled as discussed in Subsection 4.3.5), th e higher accuracy loop corresponding to .1 dB and thus using μ' then replaces the faster loop as long as no more large noise increases are observed.

Figure 4.33 illustrates the basic concept of faster swapping.

4.7.3 SNR Computation

The signal to noise ratio is then estimated for each subchannel independently as

$$\text{SNR}_n = \mathcal{E}_n \frac{|\hat{P}_n|^2}{\hat{\sigma}_n^2} \quad . \tag{4.295}$$

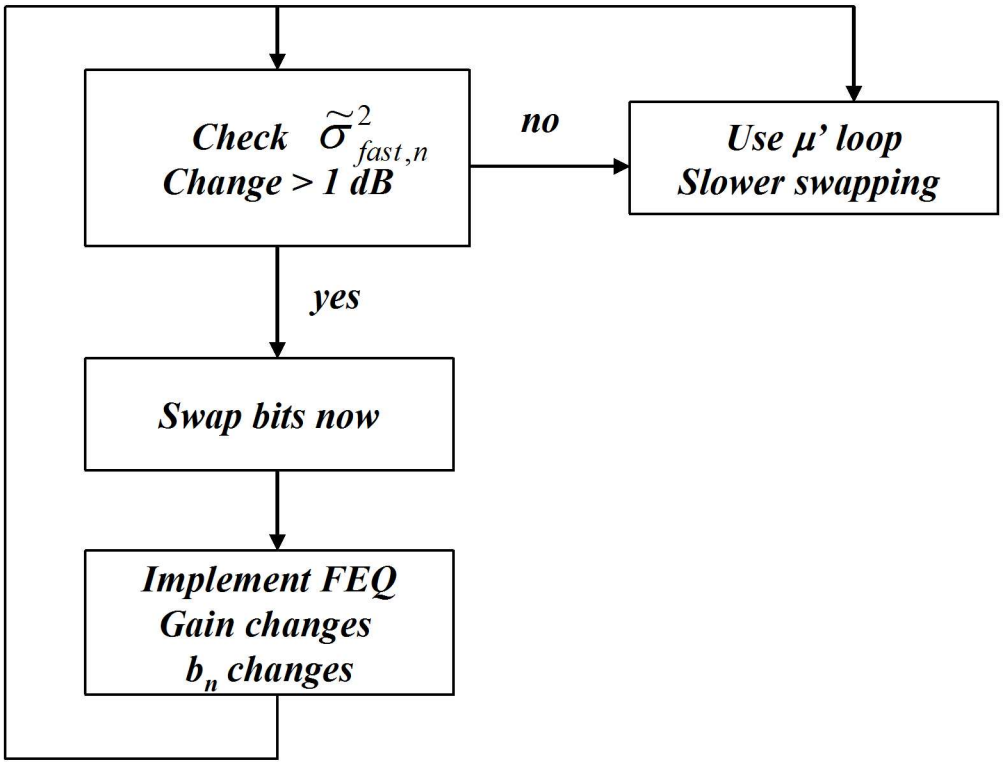


Figure 4.33: Flow chart for fast-swap augmentation of noise calculation.

Equivalently, using the FEQ, the SNR is given by (4.249). Loading can be executed in the receiver (or after passing SNR values, in the transmitter) and then the resulting b_n and $\mathcal{E}_n$ (or relatively increase/decrease in $\mathcal{E}_n$ with respect to what was used last) are sent to the transmitter via the reverse link always necessary in DMT. This step is often called "the exchange" in DMT modems and occurs as a last step prior to live transmission of data, often called "showtime!"

4.8 Stationary Equalization for Finite-length Partitioning

A potential problem in the implementation of DMT/OFDM and VC channel-partitioning methods is the use of the cyclic prefix or guard period of an extra ν samples, where $\nu+1$ is the length in sampling periods of the channel pulse response. The required excess-bandwidth is then ν/N. On many practical channels, ν can be large - values of several hundred can occur in both wireless and wired transmission problems, especially when channels with narrowband noise are considered. To minimize the excess bandwidth, N needs to be very large, potentially 10,000 samples or more. Complexity is still minimal with DMT/OFDM methods with large N, when measured in terms of operation per unit time interval. However, large N implies large memory requirement (to store the bit tables, energy tables, FEQ coefficients, and intermediate FFT/IFFT results), often dominating the implementation. Further, large N implies longer **latency** (delay from transmitter input to receiver output) in processing. Long latency complicates synchronization and can disrupt certain higher-level data protocols used and/or with certain types of carried applications signals (like voice signals).

One solution to the latency problem, invented and first studied by J. Chow (not to be confused with the P. Chow of Loading Algorithms), is to use a combination of short-length fixed (i.e. stationary) equalization and DMT/OFDM (or VC) to reduce ν, and thereby the required N, to reach the highest performance levels with less memory and less latency. The equalizer used is known as a **time equalizer (TEQ)**, and is studied in this section. Chow's basic MMSE approach is shown here to result in an eigenvector-solution problem that can also be cast in many other "generalized eigenvector" forms. Various authors have studied the TEQ since J. Chow, and to produce some improvements at least in perspective: Al-Dhahir and Evans have developed yet better theories for the TEQ, but the solution will not be easily obtained in closed form and requires advanced numerical optimization methods for non-convex optimization – Chow's TEQ often performs very close to those methods. Essentially these methods attempt to maximize the overall SNR_{dmt} by optimizing equalizer settings and energy/bit distributions jointly. A simpler approach using "cyclic reconstruction" appears later in this chapter.

Subsection 4.8.1 studies the general data-aided equalization problem with infinite-length equalizers. The TEQ is there seen to be a generalization of the feedforward section of a DFE. Subsection 4.8.2 then reviews performance calculation of the finite-length TEQ for a target ν that is potentially less than the channel pulse-response length. Subsection 4.8.3 illustrates the zero-forcing white-input specialization of Chow's TEQ. Most reasonable TEQ approaches perform about the same, usually allowing infinite-length block-symbol performance with reasonable finite block lengths. A TEQ example introduced in Subsection 4.8.2 and revisited in Subsection 4.8.3 illustrates such TEQ improvement for short block length. Subsection 4.8.5 investigates the joint loading/equalizer design problem and then suggests a solution that could be implemented.

The very stationarity of the TEQ renders it suboptimum because the DMT/OFDM or VC system is inherently cyclostationary with period $N+\nu$ samples. There are 3 "simplifications" in the development of the TEQ that are disturbing:

1. The equalizer is stationary and should instead be periodic with period $N+\nu$.

2. The input sequence x_k will be assumed to be "white" or i.i.d. in the analysis ignoring any loading that might have changed its autocorrelation.

3. A scalar mean-square error is minimized rather than the multichannel signal-to-noise quantity $\text{SNR}_{m,u}$.

Nonetheless, TEQs are heavily used and often very effective in bringing performance very close to the infinite-length multitone/modal-modulation optimum. An optimum equalizer is cyclostationary or periodic and found in Chapter 5, but necessarily more complex. When such an cyclic-equalizer is used, then problems 2 and 3 are eliminated. If the stationary fixed equalizer is used, Al-Dhahir and Evans have considered solutions to problems 2 and 3, and generally follow a solution posed in Subsection 4.8.5.

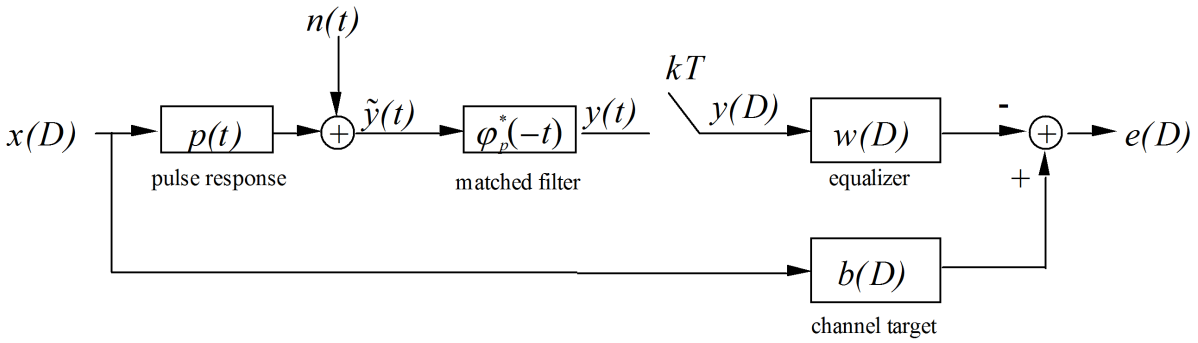


Figure 4.34: The Maximum Likelihood Infinite-Length Equalization problem.

4.8.1 The infinite-length TEQ

The general baseband complex decision-aided, or in our study **input-aided**, equalization problem appeared in Figure 4.34. The channel output $y_p(t)$ is

$$y_p(t) = \sum_k x_k \cdot p(t - kT) + n_p(t) \quad . \tag{4.296}$$

The channel output signal $y_p(t)$ is convolved with a matched (or anti-alias) filter $\varphi_p^*(-t)$ (where a superscript of $*$ denotes conjugate), sampled at rate $1/T$, and then filtered by a linear equalizer. The matched-filter output is information lossless. When the equalizer is invertible, its output is also information lossless with respect to the channel input. Thus, a maximum-likelihood detector applied to the signal at the output of the equalizer w_k in Figure 4.34 is equivalent in performance to the overall maximum-likelihood detector for the channel.

Heuristically, the function of the linear equalizer is to make the equalizer output appear like the output of the second filter, which is the convolution of the desired channel shaping function b_k and the known channel input x_k. The target b_k has a length $\nu + 1$ samples that is usually much less than the length of the channel pulse response. The minimum mean-square error between the equalizer output and the desired channel shaping is a good measure of equalizer performance. However, the implication that forcing the channel's response to fall mainly in $\nu + 1$ sample periods only loosely suggests that the geometric-SNR for the set of parallel subchannels will be maximized. Also, an imposition of a fixed intersymbol interference pattern, rather than one that varies throughout a symbol period is also questionable. Nonetheless, this method will be effective in a large number of practical situations. Further development requires some review and definition of mathematical concepts related to the channel.

Minimum Mean-Square Error Equalization

The **error sequence** in Figure 4.34 is

$$e_k = b_k * x_k - w_k * y_k \tag{4.297}$$

where $*$ denotes convolution. The **signal** is defined as the sequence $b_k * x_k$, which has signal energy $\mathcal{E}_s = \|b\|^2 \mathcal{E}_{\boldsymbol{x}}$. The **D-Transform** of a sequence x_k remains $X(D) \triangleq \sum_k x_k D^k$. The notation $x^*(D^{-*}) = \sum_k x_{-k}^* D^{-k}$ represents the time-reversed conjugate of the sequence. Thus,

$$E(D) = B(D)X(D) - W(D)Y(D) \quad . \tag{4.298}$$

The **minimum mean-square error** is

$$\sigma^2_{\text{MMSE}} = \min_{w_k\ ,\ b_k} E\left[|e_k|^2\right] \quad . \tag{4.299}$$

A trivial solution is $W(D) = B(D) = 0$ with $\sigma^2_{\text{MMSE}} = 0$. Aversion of the trivial solution requires positive signal power so that $\mathcal{E}_s > 0$ or, equivalently, $\|b\|^2 = \text{ constant } = \|p\|^2 > 0$. For flat "white" excitation (i.e., $S_x(D) = \bar{\mathcal{E}}_{\boldsymbol{x}}$) of the channel defined by $B(D)$,

$$\text{SNR}'_{\text{MFB}} = \frac{\mathcal{E}_s}{\sigma^2_{\text{MMSE}}} \quad . \tag{4.300}$$

Maximizing SNR'_{MFB} is then equivalent to minimizing σ^2_{MMSE} since $\|b\|^2$ is constant. While this problem formulation is the same as the formulation of decision feedback equalization, the TEQ does **not** restrict $B(D)$ to be causal nor monic.

The orthogonality principle of MMSE estimation states that the error sequence values e_k should be orthogonal to the inputs of estimation, some of which are the samples y_k. Thus, $E\left[e_k y_l^*\right] = 0 \ \forall \ k, l$, which is compactly written

$$\begin{aligned} R_{ey}(D) &= 0 & (4.301)\\ &= E\left[E(D) \cdot Y^*(D^{-*})\right] & (4.302)\\ &= B(D) \cdot \bar{R}_{xy}(D) - W(D) \cdot \bar{R}_{yy}(D) \quad . & (4.303) \end{aligned}$$

then,

$$W(D) = B(D) \cdot \frac{R_{xy}(D)}{R_{yy}(D)} \quad . \tag{4.304}$$

Then,

$$Y(D) = \|p\| Q(D) X(D) + n(D) \quad , \tag{4.305}$$

where $Q(D) \triangleq \frac{1}{2\pi\|p\|^2} \int_{-\pi}^{\pi} \sum_n \left|P\left(\frac{\omega - 2\pi n}{T}\right)\right|^2 d\omega$. Then,

$$\begin{aligned} \bar{R}_{xy}(D) &= \bar{\mathcal{E}}_{\boldsymbol{x}} \cdot \|p\| \cdot Q(D) & (4.306)\\ \bar{R}_{yy}(D) &= \bar{\mathcal{E}}_{\boldsymbol{x}} \cdot \|p\|^2 \cdot Q(D) \cdot \left(Q(D) + \frac{1}{\text{SNR}_{\text{MFB}}}\right) \quad . & (4.307) \end{aligned}$$

Recalling the canonical factorization from Chapter 3 DFE design,

$$Q(D) + \frac{1}{\text{SNR}_{\text{MFB}}} = \gamma_0 \cdot G(D) \cdot G^*(D^{-*}) \quad . \tag{4.308}$$

This factorization is sometimes called the "key equation."

The error autocorrelation sequence is

$$R_{ee}(D) = B(D) \left(R_{xx}(D) - \frac{R_{xy}(D)}{R_{yy}(D)} R_{yx}(D)\right) B^*(D^{-*}) \quad , \tag{4.309}$$

where the reader may recognize the autocorrelation function of the MMSE linear equalizer error in the center of the above expression. We simplify this expression further using the key equation to obtain

$$R_{ee}(D) = \left(\frac{\bar{\mathcal{E}}_{\boldsymbol{x}}}{\gamma_0 \cdot \text{SNR}_{\text{MFB}}}\right) \cdot \frac{B(D) \cdot B^*(D^{-*})}{G(D) \cdot G^*(D^{-*})} \quad . \tag{4.310}$$

Defining the (squared) norm of a sequence as

$$\|x\|^2 \triangleq \sum_{k=-\infty}^{\infty} |x_k|^2 \quad , \tag{4.311}$$

the mean square error from (4.310) is

$$\text{MSE} = \left(\frac{\bar{\mathcal{E}}_{\boldsymbol{x}}}{\gamma_0 \cdot \text{SNR}_{\text{MFB}}}\right) \|\frac{b}{g}\|^2 \quad . \tag{4.312}$$

From the Cauchy-Schwarz lemma, we know

$$\|b\|^2 = \|\frac{b}{g} \cdot g\|^2 < \|\frac{b}{g}\|^2 \|g\|^2 \tag{4.313}$$

or

$$\|\frac{b}{g}\|^2 > \frac{\|b\|^2}{\|g\|^2} \quad . \tag{4.314}$$

Thus, the MMSE is

$$\sigma^2_{\text{MMSE}} = \left(\frac{\bar{\mathcal{E}}_{\boldsymbol{x}}}{\gamma_0 \cdot \text{SNR}_{\text{MFB}}} \right) \quad , \tag{4.315}$$

which can be achieved by a multitude of $B(D)$ if the degree of $G(D)$ is less than or equal to ν[30], only one of which is causal and monic.

The MMSE solution has any $B(D)$ that satisfies

$$B(D) \cdot B^*(D^{-*}) = G(D) \cdot G^*(D^{-*}) \quad . \tag{4.316}$$

From (4.304) to (4.308), one finds $W(D)$ from $B(D)$ as

$$W(D) = \frac{1}{\gamma_0 \cdot \|p\|} \frac{B(D)}{G(D) \cdot G^*(D^{-*})} \quad , \tag{4.317}$$

which is always invertible when $\frac{\mathcal{N}_0}{2} > 0$. The various solutions differ in phase only. Such a solution is biased and the equalized channel is not exactly $B(D) = b\_i \cdot D^{-i} + ... + b_{-i+\nu} \cdot D^{-i+\nu}$ and in fact will be reduced by by the scale factor $w_i/b_{-i} \leq 1$. Such a bias can be removed by the inverse of this factor without changing performance and the channel will then be exactly $B(D)$ with an exactly compensating increase in error variance.

Solutions

As mentioned earlier, there are many choices for $B(D)$, each providing then a corresponding TEQ $W(D)$ given by Equation (4.317).

The MMSE-DFE The Minimum-Mean-Square Error Decision Feedback Equalizer (MMSE-DFE) chooses causal monic $B(D)$ as

$$B(D) = G(D) \;\text{ so }\; W(D) = \frac{1}{\gamma_0 \cdot \|p\| \cdot G^*(D^{-*})} \quad , \tag{4.318}$$

is anti-causal.

In ML detection – that is DMT or VC detection – with the MMSE-DFE choice of TEQ filter $W(D)$, the feedback filter $B(D)$ is never implemented avoiding any issue of error propagation or need for precoders and their correspodning loss. That ML detector is easily implemented in the case of DMT or VC (but requires exponentially growing complexity, see the Viterbi MLSD method of Chapter 9, if PAM/QAM is used).

The MMSE-AR Equalizer The MMSE Auto Regressive Equalizer (MMSE-ARE) corresponds to the anti-causal monic $B(D)$ choice

$$B(D) = G^*(D^{-*}) \;\text{ so }\; W(D) = \frac{1}{\gamma_0 \cdot \|p\| \cdot G(D)} \quad , \tag{4.319}$$

is causal. This is another possible solution and the MMSE level and the SNR_{MFB} are the same as all the other optimum solutions, including the MMSE-DFE. One would not make the choice in (4.319) if sample-by-sample decisions are of interest because neither future decisions are available in the receiver nor can a Tomlinson precoder be implemented (properly) with a noncausal $B(D)$. However, such causality is not an issue with the TEQ and ML detection via either DMT or VC.

[30] If the degree is greater than ν, then the methods of Section 4.8.2 must instead be used directly.

Comparison of MMSE-DFE and MMSE-ARE

Both of the MMSE-DFE and the MMSE-ARE equalizer have the same performance (when ML detection is used). Essentially the causality/noncausality conditions of the feedforward filters and feedback filters have been interchanged. Because $B(D)$ is never used in the receiver (only for our analysis), either structure could be used in a practical situation. A multicarrier modulation method used with $W(D)$ equalization would also perform the same for either choice of $W(D)$ (or any other optimum choice).

Thus, it would appear that since all structures are equivalent. The designer might as well continue with the use of the MMSE-DFE, since it is a well-known receiver structure. However, when finite-length filters are used, this can be a very poor choice in attempting to maximize performance for a given complexity of implementation. As a trivial example, consider a channel that is maximum phase, the feedforward filter for a MMSE-DFE, in combination with the matched filter, will try to convert the channel to minimum-phase (at the output of $W(D)$, leading to a long feedforward section when (as in practice) the matched-filter and feedforward filter are implemented in combination in a fractionally spaced equalizer. On the other hand, the MMSE-ARE equalizer feedforward section (combined with matched filter) is essentially (modulo an anti-alias filter before sampling) one nonzero tap that adjusts gain. The opposite would be true for a minimum-phase channel.

In data transmission, especially on cables or over wireless transmission paths, the channel characteristic is almost guaranteed to be of mixed-phase if any reflections (multipath, bridge-taps, gauge-changes, slight imbalance of terminating impedances) exist. It is then natural (and correct) to infer that the best input-aided equalization problem is one that chooses $B(D)$ and $W(D)$ to be both mixed-phase also. The problem then becomes for a finite-length feedforward section ($W(D)$) of $M+1$ taps and a finite-length channel model of $\nu+1$ taps ($B(D)$), to find the best $B(D)$ and $W(D)$ for an ML detector designed for $B(D)$, the signal constellation, and additive white Gaussian noise (even if the noise is not quite Gaussian or white). This is the problem addressed in Section 4.8.2 and the solution is often neither MMSE-DFE nor MMSE-ARE filtering.

4.8.2 The finite-length TEQ

Like the DFE of Section 3.6, the finite-length MMSE-TEQ problem is characterized by the error

$$\boldsymbol{e}_k(\Delta) = \boldsymbol{b}^* \boldsymbol{x}_{k-\Delta} - \boldsymbol{w}^* \boldsymbol{y}_k \tag{4.320}$$

where

$$\boldsymbol{x}_{k-\Delta} \triangleq \begin{bmatrix} x_{k-\Delta} \\ x_{k-\Delta-1} \\ \vdots \\ x_{k-\Delta-\nu} \end{bmatrix} \quad ; \tag{4.321}$$

$\boldsymbol{b}^* \triangleq [b_0^* \; b_1^* \; ... \; b_\nu^*]$ is not necessarily causual, monic nor minimum-phase; and $\boldsymbol{w}^* \triangleq [w_0^* \; w_1^* \; ... \; w_{L-1}^*]$, L is the number of equalizer coefficients and may be at another sampling rate as in Section 3.6 for the fractionally spaced case[31]; and

$$\boldsymbol{y}_k \triangleq \begin{bmatrix} y_k \\ y_{k-1} \\ \vdots \\ y_{k-L+1} \end{bmatrix} \quad . \tag{4.322}$$

The length ν is typically less than the length of the channel pulse response and equal to the value anticipated for use in a multichannel modulation system like DMT/OFDM or VC.

The input/output relationship of the channel is

$$\boldsymbol{y}_k = P\boldsymbol{x}_k + \boldsymbol{n}_k \quad . \tag{4.323}$$

[31]The equalizer output should be decimated to some integer multiple of the channel-input sampling rate. Usually that integer multiple is just unity in DMT and vector coding designs because that sampling rate is already greater than twice the highest frequency likely to be used. However, as Pal has noted, with very short equalizers, sometimes an oversampling factor greater than 1 can improve performance for fixed complexity, depending on channel details.

P need not be a convolution matrix and can reflect whatever guard bands or other constraints have been placed upon the transmit signal.

The MSE is minimized with a constraint of $\|\boldsymbol{b}\|^2 = c$ a constant, with c usually chosen as $c = \|p\|^2$ to force the flat-input SNR_{MFB} of the channel to be the flat-input SNR_{MFB} of the TEQ problem. However, this non-zero constant produces the same TEQ shape and does not change performance but simply avoids the trivial all-zeros solution for $\boldsymbol{w}$. Minimization first over $\boldsymbol{w}$ forces

$$E[\boldsymbol{e}(\Delta)\boldsymbol{y}_k^*] = 0 \quad . \tag{4.324}$$

The solution forces

$$\boldsymbol{w}^* = \boldsymbol{b}^* \boldsymbol{R_{xy}}(\Delta) \boldsymbol{R_{yy}}^{-1} \tag{4.325}$$

for any $\boldsymbol{b}$. Then, $\boldsymbol{e}(\Delta)$ becomes

$$\boldsymbol{e}(\Delta) = \boldsymbol{b}^*(\boldsymbol{x}_{k-\Delta} - \boldsymbol{R_{xy}}(\Delta)\boldsymbol{R_{yy}}^{-1}\boldsymbol{y}) \quad . \tag{4.326}$$

(4.326) has a vector of the minimized MSE finite-length LE errors within the parentheses. This MMSE-LE error vector has an autocorrelation matrix that can be computed as

$$R_{LE}(\Delta) = \mathcal{E}_{\boldsymbol{x}} I - \boldsymbol{R_{xy}}(\Delta)\boldsymbol{R_{yy}}^{-1}\boldsymbol{R_{yx}}(\Delta) \quad . \tag{4.327}$$

Tacit in (4.327) is the assumption that $\boldsymbol{R_{yy}}$ is a fixed $L \times L$ matrix, which is equivalent to assuming that the input to the channel is stationary as is the noise. In the case of transmission methods like DMT and VC, this stationarity is not true and the input is instead cyclostationary over a symbol period. Furthermore, the initial solution of the TEQ is often computed before transmit optimization has occurred so that $\boldsymbol{R_{xx}}$ is often assumed to be white or a diagonal matrix as will also be the case in this development. With these restrictions/assumptions, the overall MMSE is then the minimum of

$$MMSE = \min_{\boldsymbol{b},\Delta} \boldsymbol{b}^* R_{LE}(\Delta)\boldsymbol{b} \quad . \tag{4.328}$$

with the constraint that $\|\boldsymbol{b}\|^2 = c$. The solution of this problem is easily shown to be the eigenvector of $\boldsymbol{R_{ee}}$ that corresponds to the minimum eigenvalue. Thus

$$\boldsymbol{b} = \sqrt{c} \cdot \boldsymbol{q}_{min}(\Delta) \quad . \tag{4.329}$$

The issue of bias can return in MMSE design and is somewhat more involved than in Chapter 3. To calculate the correct unbiased distortion energy in general

$$\begin{aligned} E_\Delta(D) &= B(D) \cdot X(D) \cdot D^\Delta - W(D) \cdot Y(D) & (4.330)\\ &= B(D) \cdot X(D) \cdot D^\Delta - W(D) \cdot P(D) \cdot X(D) + W(D) \cdot N(D) & (4.331) \end{aligned}$$

By noting that the equalizer output component $W(D) \cdot P(D) \cdot X(D + W(D) \cdot N(D)$ will in general be scaled by α from the desired $B(D) \cdot D^\Delta \cdot X(D)$ so that $W(D) \cdot P(D) \cdot X(D + W(D) \cdot N(D) = \alpha \cdot B(D) \cdot X(D) \cdot D^\Delta + E_U(D)$, then

$$\begin{aligned} E_\Delta(D) &= B(D) \cdot X(D) \cdot D^\Delta - \alpha \cdot B(D) \cdot D^\Delta \cdot X(D) + E_U(D) & (4.332)\\ &= (1-\alpha) \cdot X(D) \cdot B(D) \cdot D^\Delta + E_U(D) \quad . & (4.333) \end{aligned}$$

Bias removal then would occur by multiplying the equalizer output by $1/\alpha$, which would more than likely be absorbed into the TEQ. The bias factor itself is obtained by first forming the convolution $C(D) = W(D) \cdot P(D)$, which is the noise-free equalized output. The bias will then be the ratio

$$\alpha = c_\Delta / b_0 \quad . \tag{4.334}$$

To compute the $r_{u,E}$ (4.333) leads to

$$\begin{aligned} \text{MMSE} &= (1-\alpha)^2 \cdot \bar{\mathcal{E}}_{\boldsymbol{x}} \cdot \|b\|^2 + r_{u,E} & (4.335)\\ &= (1-\alpha)^2 \cdot \bar{\mathcal{E}}_{\boldsymbol{x}} \cdot \|p\|^2 + r_{u,E} & (4.336)\\ &= \lambda_{min}\|p\|^2 \quad . & (4.337) \end{aligned}$$

Thus,

$$r_{u,E} = \|p\|^2 \left[\lambda_{min} - (1-\alpha)^2 \cdot \bar{\mathcal{E}}_{\boldsymbol{x}}\right] \tag{4.338}$$

provides the value for $r_{u,E}$. The unbiased $\text{SNR}_{TEQ,U}$ for the equalized channel is then

$$\text{SNR}_{TEQ,U} = \frac{\alpha^2 \cdot \bar{\mathcal{E}}_{\boldsymbol{x}} \cdot \|p\|^2}{r_{u,E}} = \frac{\alpha^2 \cdot \bar{\mathcal{E}}_{\boldsymbol{x}}}{\lambda_{min} - (1-\alpha)^2 \cdot \bar{\mathcal{E}}_{\boldsymbol{x}}} \quad . \tag{4.339}$$

However, such bias removal is not necessary in the DMT systems because of the ensuing FEQ that will remove it automatically in any case. The TEQ setting is determined by computing $\boldsymbol{R_{ee}}(\Delta)$ for all reasonable values of Δ and finding the MMSE (Δ) with the smallest value. The eigenvector corresponding to this Δ is the correct setting for $\boldsymbol{b}$ (with scaling by $\sqrt{c}$) and $\boldsymbol{w}$, the TEQ, is determined through Equation (4.325).

An alternative for Δ determination notes that equation (4.327 can also be written

$$R_{LE}(\Delta) = \mathcal{E}_{\boldsymbol{x}} I - \bar{\mathcal{E}}_{\boldsymbol{x}}^2 \left[010\right] P^* \boldsymbol{R_{yy}}^{-1} P \left[010\right] \quad , \tag{4.340}$$

where the position of the "1" determines Δ. Then, the best delta corresponds to the maximum eigenvalue of the matrix $P^* \boldsymbol{R_{yy}}^{-1} P$.

The $L \times L$ matrix $\boldsymbol{R_{yy}}$ need only be inverted once, but the eigenvectors must be computed several times corresponding to all reasonable values of Δ. This computation can be intensive. For FIR channels, the matrix $\boldsymbol{R_{yy}}$ is given by

$$\begin{aligned} \boldsymbol{R_{yy}} &= P \boldsymbol{R_{xx}} P^* + R_{\boldsymbol{nn}} \text{ general non-white input case} & (4.341) \\ &= \bar{\mathcal{E}}_{\boldsymbol{x}} P P^* + R_{\boldsymbol{nn}} \text{ usual white-input case.} & (4.342) \end{aligned}$$

Any stationary $\boldsymbol{R_{xx}}$ can be assumed. If a cyclostationary (perhaps an $\boldsymbol{R_{xx}}$ representing some known input optimization and cyclostationarity) $\boldsymbol{R_{xx}}$ is used, then $\boldsymbol{R_{yy}}$ becomes cyclostationary also and a function of both position within a symbol. The cross correlation matrix is

$$\boldsymbol{R_{xy}}(\Delta) = E\left[\boldsymbol{x}_{k-\Delta} \boldsymbol{y}_k\right] = E\left[\boldsymbol{x}_{k-\Delta} \boldsymbol{x}_k^*\right] P^* \quad . \tag{4.343}$$

The cross-correlation matrix will have zeros in many positions determined by Δ unless again an optimized input is being used and then again this matrix would also become a function of both Δ and the position within a symbol. Because it is hard to know which position within a symbol to optimize or to know in advance the optimized spectrum used on the channel (which usually is affected by the TEQ), the TEQ is set first prior to channel identification, partitioning, and loading. It could be periodically updated.

EXAMPLE 4.8.1 (IIR Channel $1/1-.9D$) A discrete-time channel might have an IIR response so that no finite ν is ever completely sufficient. An example is the single-pole IIR digital channel

$$P(D) = \frac{1}{1-.9D} \quad , \tag{4.344}$$

with time-domain response easily found to be

$$p_k = .9^k \ \ \forall \ \ k \geq 0 \quad . \tag{4.345}$$

The gain of the channel is $\|p\|^2 = \frac{1}{1-.9^2} = 5.2632$. Let the AWGN variance per real dimension be $\sigma^2 = .1$ and the input energy is $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$. If PAM were used on this channel, then $\text{SNR}_{\text{MFB}} = 17.2$ dB. It is interesting to investigate the amount of energy with guard period of ν samples as ν varies. The total energy outside of a guard period of length ν is then

$$\|p\|^2 - \sum_{k=0}^{\nu} \frac{1-(.81)^{\nu+1}}{1-.81} \quad . \tag{4.346}$$

This lost energy, which becomes distortion for the DMT or VC system is in Table 4.6

ν	0	1	2	3	4	...	∞
$\mathcal{E}_{lost}$	4.26	3.45	2.99	2.26	1.83	...	0
dB loss $= \frac{5.26-\mathcal{E}_{lost}}{5.26}$	-7.2	-4.6	-3.6	-2.4	-1.9	...	0

Table 4.6: Table of energy loss by truncation (rectangular windowing) of $p_k = .9^k \cdot u_k$.

To have only .1 dB loss, then $\nu > 16$, meaning $N > 500$ for small loss in energy and bandwidth. Alternatively, a TEQ can reshape the channel so that most of the energy is in the first ν coefficients. Because of the special nature of a single-pole IIR channel, it is pretty clear that a 3-tap equalizer with characteristic close to $1 - .9D$ will reduce ν substantially while altering the noise. Thus, it is fairly clear that an equalizer with $L = 3$ taps and delay $\Delta = 0$ should work fairly well. This example also uses $\nu = 1$. For these values, some algebra leads to

$$\boldsymbol{R_{xy}}(0) = \begin{bmatrix} 1 & 0 & 0 \\ .9 & 1 & 0 \end{bmatrix} \tag{4.347}$$

and

$$\boldsymbol{R_{yy}} = \begin{bmatrix} 5.3636 & .9(5.2636) & .81(5.2636) \\ .9(5.2636) & 5.3636 & .9(5.2636) \\ .81(5.2636) & .9(5.2636) & 5.3636 \end{bmatrix} \quad . \tag{4.348}$$

The error autocorrelation matrix is

$$\boldsymbol{R_{ee}} = \begin{bmatrix} .1483 & -.0650 \\ -.0650 & .1472 \end{bmatrix} \quad , \tag{4.349}$$

with eigenvalues .2128 and .0828 and corresponding eigenvectors [.7101 −.7041] and [.7041 .7101] respectively. Then $\boldsymbol{R_{ee}}$ has smallest eigenvalue .0828 with corresponding MMSE=.4358, much less than the $\sigma^2 + 3.45 = .1 + 3.45 = 3.55$ (3.45 is the entry for energy loss in Table 4.6 above when $\nu = 1$) for no equalization. Clearly the equalizer reduces distortion significantly with respect to no use of equalizer. The corresponding setting for $\boldsymbol{b}$ from Equation (4.329) is

$$B(D) = \|p\| \cdot [.7041\ .7101] = \sqrt{5.2632} \cdot [.7041\ .7101] = 1.6151 \cdot [1 + 1.0084D] \quad . \tag{4.350}$$

$B(D)$ is not minimum phase in this example, illustrated that the TEQ is not simply a "DFE solution." The corresponding value for $\boldsymbol{w}$ from Equation (4.325) leads to $W(D) = 1.4803 \cdot (1 + .1084D - .8907D^2)$ and a new equalized channel

$$\begin{aligned} z(D) &= 1.4803 \cdot \left(1 + .1084D - .8907D^2\right) \cdot \frac{X(D)}{1 - .9D} + W(D) \cdot N(D) & (4.351) \\ &= 1.4803 \left[1 + (.1084 + .9)D + (.81 + .9 \cdot .1084 - .8907) \cdot \frac{D^2}{1 - .9D}\right] \cdot X(D) + W \cdot N \\ &= \underbrace{1.4803\left[1 + (.1084 + .9)D\right]}_{\frac{w_0}{b_0}B(D)} + \underbrace{1.4803(.01686) \cdot \frac{X(D) \cdot D^2}{1 - .9D} + W(D) \cdot N(D)}_{\frac{w_0}{b_0}E(D)} \quad . \end{aligned}$$

The bias in estimating $B(D)$ is $w_0/b_0 = \frac{1.4808}{1.6151} = .9168$ and can be removed by multiplying the entire received signal by the constant b_0/w_0 without altering performance – this step is usually not necessary in DMT systems because the multichannel normalizer automatically includes any such adjustment. The PSD of the distortion can be determined as

$$\begin{aligned} R_{EE}(D) \cdot \left(\frac{w_0}{b_0}\right)^2 &= 1.4803^2 \cdot (.01686)^2 \frac{1}{1 - .9^2} + \cdot.1 \cdot W(D) \cdot W^*(D^{-*}) & (4.352) \\ &= 1.4803^2 \cdot \left[(.001496) + .1 \cdot \left(1 + .1084^2 + .8907^2\right)\right] & (4.353) \\ r_{EE}(0) \cdot \left(\frac{w_0}{b_0}\right)^2 &= .3988 = .4538 \cdot \left(\frac{w_0}{b_0}\right)^2 \quad . & (4.354) \end{aligned}$$

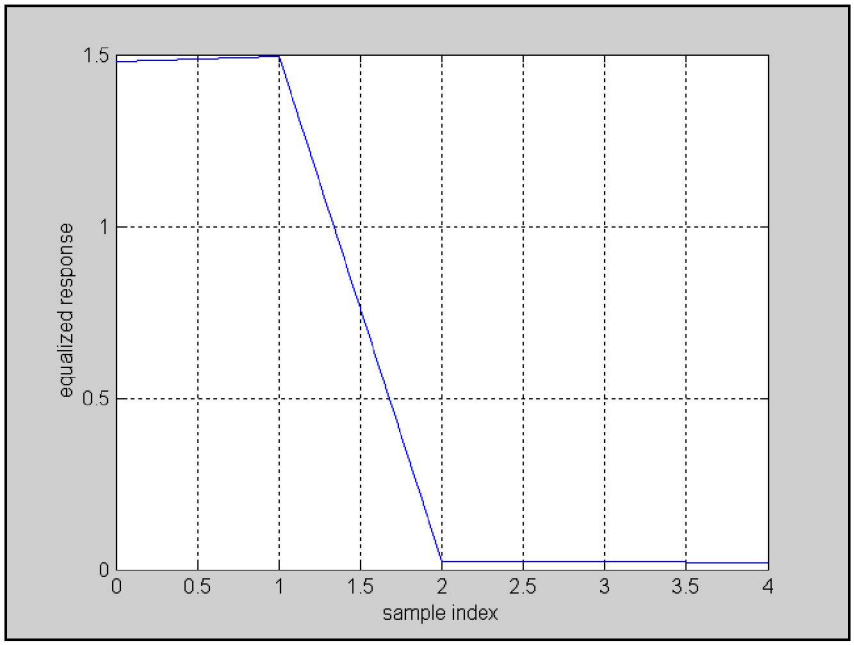


Figure 4.35: Equalized response for 3-tap TEQ with $1/(1-.9D)$ channel.

The ratio of signal energy in the first two terms to distortion is then $1.4803^2 \cdot (1+1.0084^2)/.3988$ or 10.4 dB. Equivalently, the calculation from Equation (4.339) is

$$\text{SNR}_{TEQ,U} = \frac{1 \cdot .9168^2}{.0828 - (1-.9168)^2 \cdot 1} = 10.4 \text{ dB}. \tag{4.355}$$

This signal to noise ratio cannot be significantly improved with longer equalizers, which can be verfied by trial of larger lengths, nor is $\nu > 1$ necessary as one can determine by comparing with an infinite-length MT result for this channel.

This new channel plus distortion can be compared against the old for any value of N in VC or DMT. The equalized channel is shown in Figure 4.35 where it is clear the first two positions contain virtually all the energy. Figure 4.36 shows the power spectrum of the error, which in this case is almost flat (see the vertical scale) so a designer could apply loading directly to the subchannels as if the noise were "white" and of variance .3988 without significant error. Without this "TEQ," the loss in performance is much higher at -4.6 dB from Table 4.6 versus $\frac{5.26-.3988}{5.26}$ = -0.3 dB with the TEQ.

The TEQ performance increase over no equalizer can indeed be much greater than 4.3 dB on channels with more severe responses. (See for instance "Salvekar's Marathon" in Problem 4.25.) The TEQ is not a DFE and does not maintain white noise nor channel magnitude in the finite-length case.

The following second example shows a near equivalence of a DFE and an equalized DMT system when both use exactly the same bandwidth, a concept explored and explained further in Chapter 5.

EXAMPLE 4.8.2 (Full-band TEQ) A 7-tap FIR channel [.729 .81 −.9 2 .9 .81 .729] is shortened to $\nu = 3$ (4 taps) with an 11-tap equalizer in this example and then DMT applied. The channel is such that all 128 tones of the DMT system are used. This is compared against a MMSE-DFE with 20 total taps (14 feedforward, 6 feedback using Chapter 3's Matlab DFE program). The matlab commands are inserted here and the reader can then follow the design directly:

```
>> [snr,w]=dfecolor(1,p,14,6,13,1,[.1 zeros(1,13)])
```

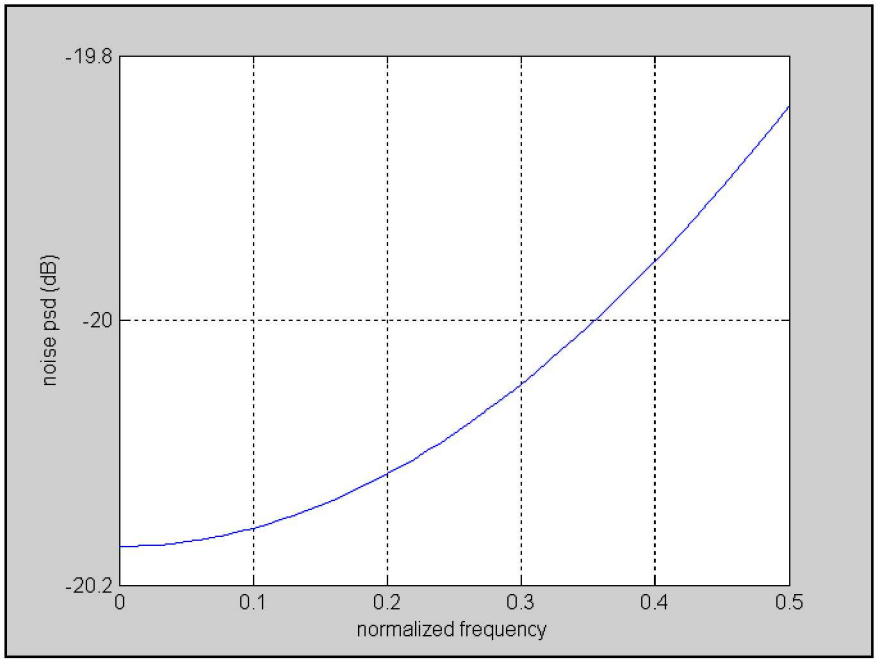


Figure 4.36: Distortion Spectrum for 3-tap TEQ on $1/(1-.9D)$ channel with white noise at $\sigma^2 = .1$.

```
snr = 16.8387
w =  -0.0046    0.0056    0.0153   -0.0117   -0.0232    0.0431    0.0654   -0.0500   -0.0447
      0.2263    0.2748   -0.0648    0.1642   -0.1479

feedback is -0.0008   -0.7908    0.0001    0.0473    0.0001    0.1078
-----------------------------------------------------------------------------------------------
>> p=[-.729 .81 -.9 2 .9 .81 .729 ];
>> np=norm(p)^2
np = 7.9951
>> P=toeplitz([p(1) , zeros(1,10)]',[p, zeros(1,10)]);
some entries in P are:
P =  Columns 1 through 8
   -0.7290    0.8100   -0.9000    2.0000    0.9000    0.8100    0.7290         0
         0   -0.7290    0.8100   -0.9000    2.0000    0.9000    0.8100    0.7290
         0         0   -0.7290    0.8100   -0.9000    2.0000    0.9000    0.8100
         0         0         0   -0.7290    0.8100   -0.9000    2.0000    0.9000
         0         0         0         0   -0.7290    0.8100   -0.9000    2.0000
         0         0         0         0         0   -0.7290    0.8100   -0.9000

>> ryy=P*P'+.1*eye(11);
```

The delay Δ after some experimentation was found to be $\Delta = 10$ for an 11-tap TEQ on this particular channel. This example continues then in Matlab according to:

```
>> rxy=[ zeros(4,10) eye(4) zeros(4,3)]*P'
rxy =   Columns 1 through 8

         0         0         0         0    0.7290    0.8100    0.9000    2.0000
```

```
         0         0         0         0         0    0.7290    0.8100    0.9000
         0         0         0         0         0         0    0.7290    0.8100
         0         0         0         0         0         0         0    0.7290

  Columns 9 through 11

   -0.9000    0.8100   -0.7290
    2.0000   -0.9000    0.8100
    0.9000    2.0000   -0.9000
    0.8100    0.9000    2.0000

>> rle=eye(4)-rxy*inv(ryy)*rxy'

rle=0.0881    0.0068   -0.0786   -0.0694
    0.0068    0.1000   -0.0045   -0.1353
   -0.0786   -0.0045    0.1101    0.0464
   -0.0694   -0.1353    0.0464    0.3666

>> [v,d]=eig(rle)
v =-0.2186   -0.5939   -0.7658   -0.1144
   -0.3545    0.2811   -0.2449    0.8575
    0.1787    0.7319   -0.5695   -0.3287
    0.8914   -0.1806   -0.1710    0.3789

d = 0.4467         0         0         0
         0    0.1607         0         0
         0         0    0.0164         0
         0         0         0    0.0410

>> b=sqrt(np)*v(:,3)'
b =  -2.1653   -0.6925   -1.6103   -0.4834

>> w=b*rxy*inv(ryy)
w = 0.0101 0.0356 -0.0771 -0.1636 0.0718 0.1477 -0.4777 -0.7924 -0.0078 -0.2237 0.1549
```

The α is .89836 and the $\text{SNR}_{\text{MFB}} = 17.7868$ dB corresponding to a biased error energy of .1288 and unbiased at .1331. A TEQ program written by 2005 student Ryoulhee Kwak executes this procedure and appears in Subsection 4.8.4 and at the web page. By running the program a few times, one can find that by using $\nu = 3$ and 14 taps in the equalizer with delay 9, the SNR_{MFB} increases to 18.9 dB. By then running water-filling, one finds an SNR in the range of 15 to 16 dB for various TEQ's of different length on this channel.

Instead, if no TEQ is used with FFT size 1024, then all tones are used and

```
>> [gn,en_bar,bn_bar,Nstar,b_bar]=waterfill(b,19.2,1,1024,0);
>> Nstar = 1024

>> b_bar = 2.8004

>> 10*log10(2^(2*b_bar) -1)
ans = 16.8
```

The performance is the same as the DFE system earlier. Thus, this second example illustrates that sometimes the TEQ is best not used and perhaps a larger symbol size is better. The no-TEQ DMT receiver has total complexity $\log_2(1024) = 10$ per sample in each of the

transmitter and receiver, while the DFE has complexity 20. Both get the same performance for about the same total complexity (transmitter and receiver) in the case that entire set of tones is used (a curious result developed more in Chapter 5).

4.8.3 Zero-forcing Approximations of the TEQ

There are several alternative approaches to design of the TEQ, one of which is the ZF TEQ. The ZF TEQ method does not consider MMSE and instead looks at the energy of an equalized response $W(D) \cdot P(D)$ within a "window" of $\nu + 1$ samples and the energy outside that window in the other $N - 1$ samples, which are called the "wall." The heurestic rationale is that energy in the equalizer response in the wall should at a minimum relative to the energy within the "window." First, this "window/wall" approach tacitly assumes that the channel input power spectral density is white $S_x(D) = \bar{\mathcal{E}}_{\boldsymbol{x}}$ (or else this criterion makes no sense). Furthermore, noise is clearly ignored in the approach. For finite-length TEQ (or any equalizer) design, zero-forcing solutions cannot exactly force a specific $\boldsymbol{b}$. The MMSE approach to equalizer design becomes zero forcing for all lengths when the SNR parameter is set to an infinite value. In this case, the mean-square energy of the "wall" is then necessarily minimized relative to the total energy of $\bar{\mathcal{E}}_{\boldsymbol{x}} \cdot c = \cdot \|p\|^2$ in the MMSE-TEQ when the channel input is white. As Ishai astutely noted, the problem of maximizing window to wall energy is the same as maximizing total equalized pulse energy to wall energy. Thus, all these approaches (which eventually result in a generalized eigenvalue problem that must be solved) are the same as the MMSE-TEQ when the designer assumes white input spectrum $S_x(D) = \bar{\mathcal{E}}_{\boldsymbol{x}}$ and sets $\frac{\mathcal{N}_0}{2} = 0$. The MMSE-TEQ considers this solution a spossible among those considered for optimization, and thus is always the same or better.

EXAMPLE 4.8.3 (Return to $1/(1-.9D)$ channel example of Section 4.8.2) A designer might decide to use one of the various zero-forcing TEQ designs instead on the $\frac{1}{1-.9D}$ channel. Clearly if noise is ignored, then a best equalizer is $w(D) = \sqrt{5.2636} \cdot (1-.9D)$, which flattens the channel so that even $\nu = 0$ is sufficient. In fact all zero-forcing equalizers for $\nu = 1$ are given by the class $W(D) = (a+b \cdot D) \cdot (1-.9D)$ where $a^2+b^2 = \|p\|^2$. The first zero-forcing equalizer has smallest enhanced noise given by .9527 relative to $\|p\|^2 = 5.2636$ for a ratio of 7.4 dB, about 3 dB worse than including the effect of the white noise. A better choice of ZF-TEQ (since there are many in this special case) would be $W(D) = (1.65 - 1.59 \cdot D) \cdot (1 - .9D)$.[32], and has ratio instead of 10 dB, which is still less than 10.4 dB. In fact, no higher ratio than 10 dB can be found for the ZF case. Thus, all ZF-TEQ's would all have worse SNR. However, the astute reader might observe loading should be executed with the correct noise power spectral density (which is not close to white in this example). This observation that it is really SNR_{dmt} that is important then leads to the next subsection.

4.8.4 Kwak TEQ Program

Former student Ryoulhee Kwak has provided the TEQ program at the web site that works for both FIR and (all pole) FIR channel inputs. The program is

```
function [P,ryy,rxy,rle,v,d,w,b,mmse,SNRmfb_in_dB,c]=teq(p,L,mu,delta,sigma,Ex_bar,filtertype)
% programmed by Ryoulhee Kwak
%
% filtertype 1=FIR , 2= IIR(just for one pole filter)
% p=channel impulse response
% in case of FIR, p= 1+0.5D+2.3D^2->p=[1 0.5 2.3]
% in case of IIR,  p=1/(2.4-0.5D)-> p=[2.4 -0.5]
% sigma =>noise power in linear scale
% delta =>delay
% L => number of taps
```

[32] The author found this by minimizing output noise power over choices for a and b, which results in a quadratic equation for b with solutions $b = -1.59$ or $-.626$ ($a = 2.21$).

```
% c=> w*p
% v=>eigen  vectors
% d=>eigent values
%mu => for b

if filtertype==1% FIR

    norm_p=norm(p);
    [temp size_p]=size(p);
    P=toeplitz([p(1), zeros(1,L-1)]',[p,zeros(1,L-1)]);

    ryy=Ex_bar*P*P'+sigma*eye(L,L);
    rxy=[zeros(mu+1,delta) eye(mu+1) zeros(mu+1,L+size_p-2-mu-delta)]*P'*Ex_bar;
    rle=eye(mu+1)*Ex_bar-rxy*inv(ryy)*rxy';
    [v d]=eig(rle);
    d_temp=diag(d)';
    [m,n ]= min(d_temp);
    b=norm_p*v(:,n)';
    w=b*rxy*inv(ryy);
    conv(w,p);
    mmse=b*rle*b';
%by using easy way to get error energy
    c=conv(w,p)
    alpa=c(delta+1)./b(1)
    biased_error_energy=norm_p^2*(m-Ex_bar*(1-alpa)^2)
    unbiased_error_energy=norm_p^2*(m-Ex_bar*(1-alpa)^2)/alpa^2

    SNRmfb=norm_p^2*Ex_bar/unbiased_error_energy;
    SNRmfb_in_dB=10*log10(SNRmfb)

else %IIR
    %in case of IIR filter P matrix is infinite dimesion but there is a trick to get exact rxy, ryy
    norm_p=sqrt(1/(p(1)^2*(1-(p(2)/p(1))^2 )));
    [temp size_p]=size(p);
    ptemp=[(-p(2)/p(1)).^(0:1:L-1)]/p(1); %-1 factor!
    P=toeplitz([ptemp(1) zeros(1,L-1)]',ptemp);
    Ptemp=toeplitz(ptemp',ptemp);

    ryy=Ex_bar*Ptemp*norm_p^2+sigma*eye(L,L);
    rxy=[zeros(mu+1,delta) eye(mu+1) zeros(mu+1,L-1-mu-delta)]*P'*Ex_bar;
    rle=eye(mu+1)*Ex_bar-rxy*inv(ryy)*rxy';
    [v d]=eig(rle);
    d_temp=diag(d)';
    [m,n ]= min(d_temp);
    b=norm_p*v(:,n)';
    w=b*rxy*inv(ryy);
    c=conv(w,p);
    sum(conv(w,p).^2)-norm(b)^2*(w(1)/b(1))^2;
    mmse=b*rle*b';
 %by using easy way to get error energy
    alpa=c(delta+1)./b(1)
    biased_error_energy=norm_p^2*(m-Ex_bar*(1-alpa)^2)
    unbiased_error_energy=norm_p^2*(m-Ex_bar*(1-alpa)^2)/alpa^2
```

```
    SNRmfb=norm_p^2*Ex_bar/unbiased_error_energy;
    SNRmfb_in_dB=10*log10(SNRmfb);
end
```

This program can be used to design MMSE-TEQ's.

4.8.5 Joint Loading-TEQ Design

Usually TEQ's are designed (or trained) before loading in DMT or VC systems, and an assumption of initially flat input spectrum $S_x(D) = \bar{\mathcal{E}}_{\boldsymbol{x}}$ is then made. For OFDM, this assumption is always true, but when loading is used in DMT, then it will likely not be true once loading is later computed. In fact, an optimum design would maximize SNR_{dmt} over the TEQ settings as well as the $\mathcal{E}_n$ for each subchannel. Al-Dhahir was the first to approach this problem and attempt solution. A better solution emanated from Evans (see December 2001 IEEE transactions on DSP). While Evans beautifully poses the problem, he notes at the end that it requires very difficult and advanced optimization procedures to be exactly solved. He further notes that such solution is not feasible in actual implementation. Evans then reverts to the MMSE solution and shows it is often close to the MAX-SNR_{dmt} solution. Some later work by Evans and Johnson finds simpler algorithms, but often the MMSE-TEQ is sufficient.

Thus for a non-white (non-diagonal) $\boldsymbol{R}_{\boldsymbol{xx}}$, this subsection poses the following solution:

1. Initially design MMSE-TEQ for flat or white input.
2. Execute loading (water-filling or LC) and compute SNR.
3. Redesign the MMSE-TEQ using the new $R_{\boldsymbol{xx}}$.
4. Re-loading the channel and see if SNR_{dmt} has significantly improved.
   (a) If improved, return to step 3 with the latest $R_{\boldsymbol{xx}}$ from loading.
   (b) If not, use the TEQ with best SNR_{dmt} so far determined.

Clearly with dynamic loading, the movement of energy could cause the TEQ to be somewhat deffi-cient. This would be particularly true if narrow-band noise occurred after initial training. The above procedure can then be repeated. The TEQ change would need to occur on the same symbol that synchro-nizes the change of energy and bit tables in the corresponding modems, so thus should be synchronized to bit swapping.

4.8.6 Block Decision Feedback

The overhead of the extra ν dimensions of vector coding and and DMT/OFDM is negligible for large N. When large N leads to unacceptable delay in the transmission path, smaller N may have to be used. In this case, the extra ν samples between symbols (blocks of samples) may be eliminated by what is known as Kasturia's Block Decision Feedback Equalizer.

Block Decision Feedback uses the decisions from the last block of samples to reconstruct and cancel the interference from previous symbols. The pulse response is known in the receiver so that the effect of the last ν columns of P can be subtracted as in Figure 4.37. Assuming previous decisions are correct, the resultant vector channel description is

$$\boldsymbol{y} = \tilde{P}\boldsymbol{x} + \boldsymbol{n} \quad , \tag{4.356}$$

where $\tilde{P}$ is the $N \times N$ matrix containing the left-most N columns of P. P must be converted to a minimum-phase (MMSE sense) channel by a conventional MMSE-DFE feedforward filter for best performance. Vector coding is applied to this matrix channel with SVD performed on the matrix $\tilde{P}$. While simple in concept, the subtraction of previous symbols in practice can be complex to implement.

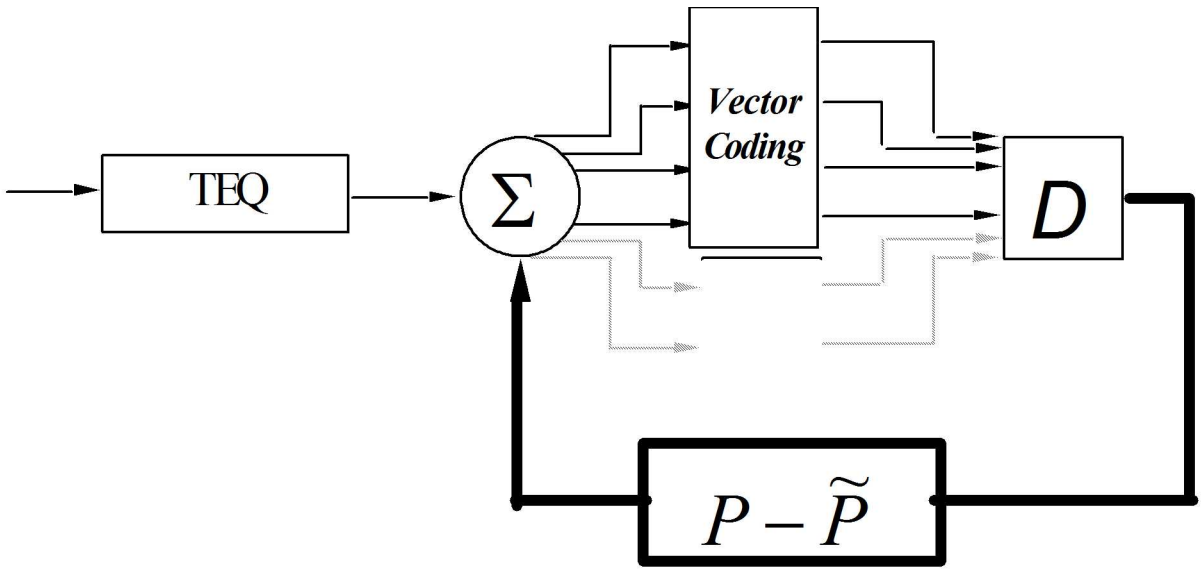


Figure 4.37: Kasturia's Block DFE.

4.8.7 Cheong's Cyclic Reconstruction and Per-tone Equalizers

K. W. Cheong in his 2000 dissertation and earlier results used the concept of cyclic reconstruction (first introduced for echo cancellers by Ho and Cioffi) to simplify the concept of a full $N \times N$ block equalizer that would be equivalent to a separate TEQ for each tone.[33] Van Acker and Mooenen (and van de Wiel and Pollet) later renamed this matrix equalization a "per-tone equalizer" and used the equivalent of cyclic reconstruction to simplify the amount of computation (apparently unaware of Cheong's earlier work and the equivalence to cyclic reconstruction). The method requires a slightly higher complexity than a TEQ in real-time, but does not require eigen-vector computation. This method actually then leads to an easy adaptive implementation that maximizes product SNR.

The key observation in cyclic reconstruction is that the cyclic prefix could be faked to look longer if at each sampling instant that is not already part of the cyclic prefix the value from the previous symbol were somehow replaced by the sample from the current symbol. Essentially, the cyclic prefix becomes infinite.

If the receiver saves the last symbol of time-domain samples, and they were basedon correct decisions, then the ISI from the last sample could be subtracted. Also, an ISI equivalent to the repeated samples could be added. Cheong used this in a basic Block DFE structure to allow simplification and retain use of the FFT's (unlike Kasturia's earlier block DFE). At some further increase in complexity, it is possible that Cheong's equalizer can be implemented also in the time domain using instead the channel outputs. A truly cyclic channel would have outputs that are also cyclic. Any deviations from such cyclic behavior would be completely represented by the differences between points separated by N samples.

Thus a set of differences can be computed by the receiver or

The basic observation is the nominal FFT has N values

$$(y_{N+i-1} - y_{i-1})\ i = 1, ..., N\ . \tag{4.357}$$

Any deviation from perfect cyclic channels must be a linear combination of these differences on a linear channel. Thus, one implementation is shown in Figure 4.38 The coefficients are adapted according to the same algorithm as in (4.134) using the same error signal $\tilde{U}_n$ but with each FEQ coefficient augmented by a set of N taps that have the time-domain differences. This remains the complexity of

[33] Clearly having a separate equalizer for each tone allows the individual SNRs to be maximized and then also the product SNR to be maximized given a certain input energy distribution, essentially decoupling the loading problem from the equalization problem.

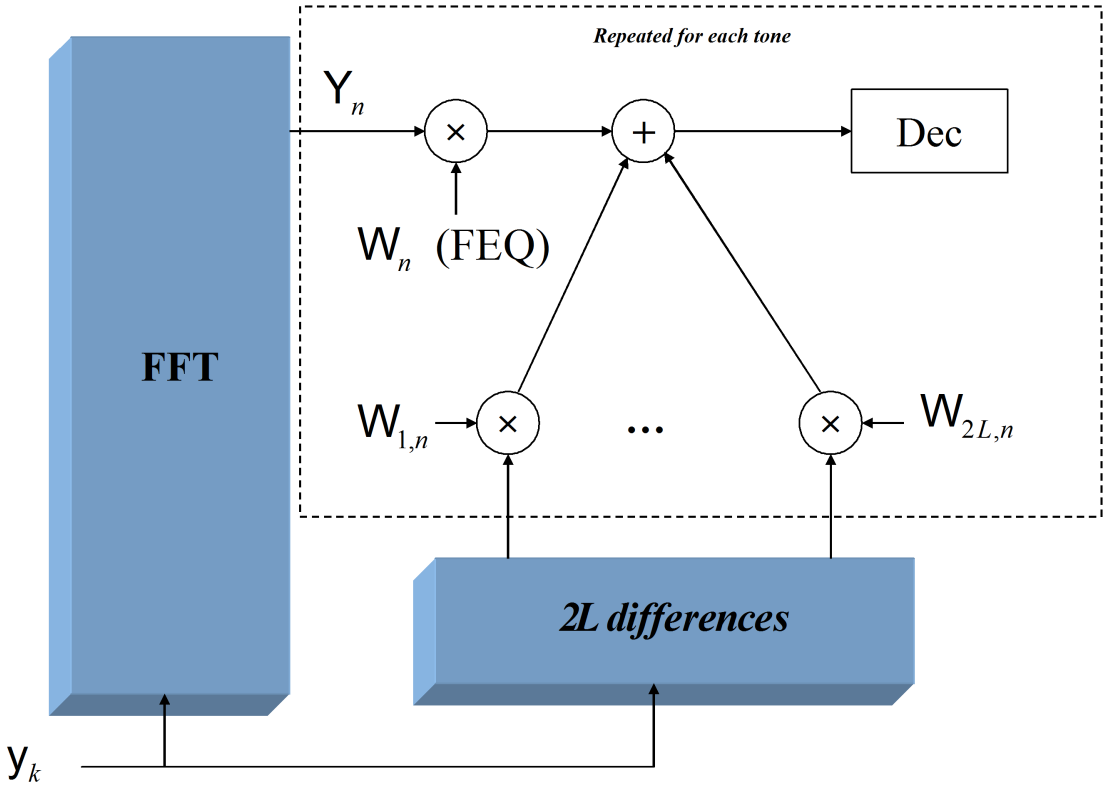


Figure 4.38: Cheong's Cyclic Reconstruction (sometimes called a "per-tone" equalizer).

$N \times N$. However, differences far away from the cyclic prefix will have negligible contribution. If the (FIR) TEQ for any (or all) tones were L taps long, then clearly it has only L degrees of freedom and could be affected by at most $2L$ differences.

Thus the complexity is essentially double that of an L-tap TEQ but also has the advantage of essentially being a different equalizer for each tone output. Moonen and Acker developed what they call a "per-tone" equalizer well after the Kasturia and Cheong work, but did not realize the equivalent use of cyclic reconstruction, thus the apparent different name. However, the concept is clearly equivalent when viewed from the perspective of cyclic reconstruction. These $2L$ samples are the same for all tones but do depend on the delay δ chosen. In a system with no delay, they would be the $2L$ samples closest to the previous symbol. With a delay delta, the $2L - \delta$ samples closest to the previous symbol and the δ samples at the end of the symbol would be used. Such a system requires collection of the next symbol to compute the differences (or at least δ output samples from the next symbol) and so has between δ samples and one symbol of additional delay required for implementation.

An adaptive implementation would use the ZF algorithm used for the FEQ on each tone with the single complex coefficient now being replaced by $1 + 2L$ difference terms.

4.9 Windowing Methods for DMT

While cyclostationary[34] DMT/OFDM signals are not deterministically periodic over all time simply because the transmitted input message is not the same in each symbol period. Deterministically for any tone, the corresponding output signal for a given symbol period is

$$x_n(t) = X_n \cdot e^{\jmath \frac{2\pi}{N} \cdot \frac{N+\nu}{T} \cdot nt} \cdot w_T(t) \quad , \tag{4.358}$$

where $w_T(t)$ is a window function. In the DMT/OFDM partitioning of Section 4.6, the rectangular window function was always

$$w_T(t) = \begin{cases} 1 & t \in [0, T) \\ 0 & t \notin [0, T) \end{cases} \quad . \tag{4.359}$$

Letting

$$\bar{f} \triangleq \left(1 + \frac{\nu}{N}\right) \cdot \frac{1}{T} = \cdot \frac{1}{NT'} \quad , \tag{4.360}$$

the product in (4.358) corresponds to convolution in the frequency domain of the impulse $X_n \cdot \delta(f - \bar{f})$ with $W_T(f)$. The fourier transform of the rectangular window in (4.359) is

$$W_T(f) = \text{sinc}\left(\frac{f}{\bar{f}}\right) \cdot e^{-\jmath \pi f T} \quad , \tag{4.361}$$

which has the magnitude illustrated in Figure 4.39. While there is no interference between tones of DMT because of the zeros in this function, at intermediate frequencies ($f \neq n \cdot \bar{f}$), there is non-zero energy. Thus the DMT spectrum is not a "brick-wall" bandpass system of tones, but decays rather slowly with frequency from any tone. Such slow decay may be of concern if emissions from the present channel into other transmission systems[35] are of concern. Even other DMT transmissions systems that might sense crosstalk from the current DMT system would need exact synchronization of sampling rate $1/T'$ to avert such crosstalk. A slight difference in DMT sampling clocks, or just another modulation method, on the other transmission system might lead to high interference. For this reason, some DMT/OFDM modulators use a non-rectangular window to reduce the "sidelobes" with respect to those in Figure 4.39.

Non-rectangular windowing of the transmit signal in DMT could re-introduce interference among the tones, so must be carefully introduced. Figure 4.40 illustrates two methods for introducing windows without reintroduction of intra-symbol interference through the use of extra cyclic prefix and suffix. The cyclic suffix used for Zippered duplexing in Section 4.6 is also useful in windowing. Extra prefix samples beyond those needed for ISI (that is larger than the minimum ν required) and extra suffix samples (beyond either 0 with no Zippering, or larger than the minimum Δ required with Zippering) are used for the shaped portion of the window. The minimum $N + \nu + \Delta$ samples within the symbol remain unwindowed (or multiplied by 1, implying no change) so that DMT or OFDM partitioning works as designed when the receiver FFT processes only N of these samples. The remaining samples are shaped in a way that hopefully reduces the sidelobes of the window function, and thus makes the frequency response of the modulated signal fall more sharply. This sharper fall is achieved by having as many continuous derivatives as is possible in the window function. The windows from the extra prefix of the current symbol and the extra suffix of the last symbol can overlap without change in performance if the sum of the two windows in the overlapping region equals 1. Such overlap of course reduces the increase in excess bandwidth introduced by the window.

The most common choice of window in practice is a raised-cosine window of $2L + 1$ samples of extra cyclic extension (extra prefix plus extra suffix). This function is given in the time domain by

$$w_{T,rcr}(t) = \frac{1}{2} \cdot \left\{1 + \cos\left(\pi \left[\frac{t}{LT'}\right]\right)\right\} \forall t \in (-LT', LT') \tag{4.362}$$

where time $t = 0$ is in the middle of the extra extension. The actual implementation occurs with samples $w_k = w_{T,rcr}(kT')$ for $k = -L, ..., 0, ..., L$. Letting

$$\bar{\alpha} = \frac{L}{N + \nu + \Delta} \quad , \tag{4.363}$$

[34] DMT/OFDM signals are stationary at symbol intervals, but cyclic over a symbol period.

[35] That may share the same channel or unfortunately sense radiation from it.

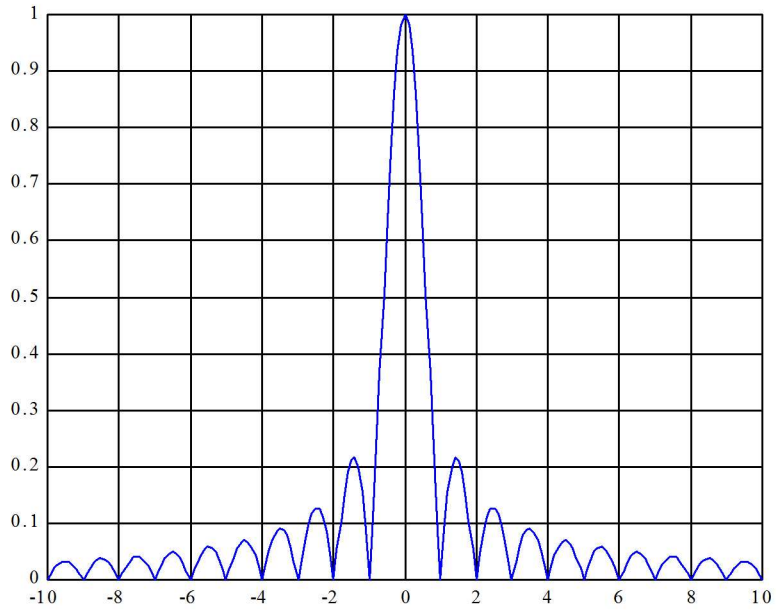


Figure 4.39: Rectangular window fourier transform versus $\frac{f-\bar{f}}{f}$.

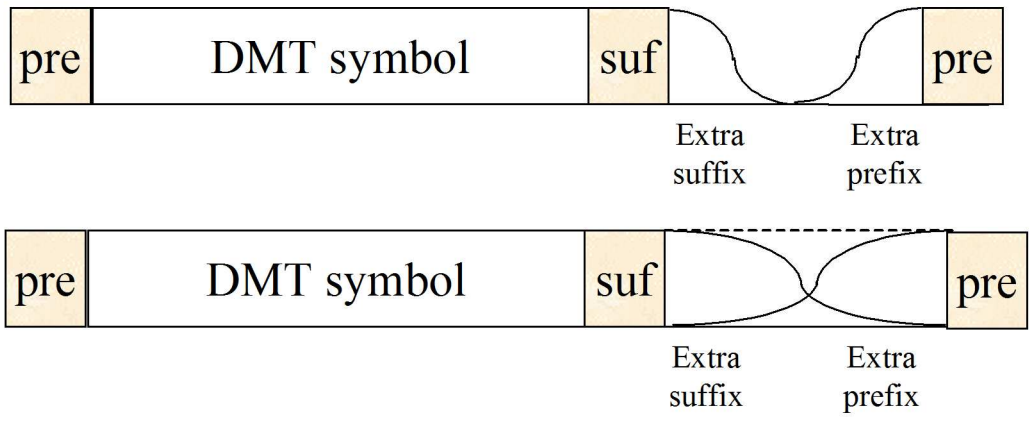


Figure 4.40: Illustration of extra prefix and suffix and non-overlapped and overlapped windows.

the fourier transform of this window becomes

$$W_{T,rcr}(f) = \frac{L}{\bar{\alpha}} \cdot \text{sinc}\left(\frac{\pi f}{\bar{f}}\right) \cdot \frac{\cos\frac{\bar{\alpha}\cdot f}{\bar{f}}}{1 - \left(\frac{2\bar{\alpha}\pi f}{\bar{f}}\right)^2} \quad . \tag{4.364}$$

Figure 4.41 illustrates the sidelobes of the raised cosine function for 5% (blue) and 25% (red) along with those of $W_T(f)$ from Figure 4.39 (this time in decibels). Emissions into other systems can be significantly reduced for between 5% and 25% excess bandwidth. The FMT systems of Section 4.6 offer even greater reduction (or can be combined) if required.

4.9.1 Receiver Windowing

Also of interest are any non-white noises that may have autocorrelation of length greater than $\nu T'$. Extremely narrow band interference noise, sometimes called "RF interference" is an example of such a noise. DMT block length N may not be selected sufficiently large and the TEQ of section 4.8 may not adapt quickly enough to "notch" or reduce such noise, or the TEQ may not be sufficiently long to reduce the noise level. The same extra prefix and suffix interval in the receiver can again be used to apply a window (perhaps the same as used in the transmitter). Windowing at the receiver again corresponds to frequency-domain convolution with the transform of the window function. Low sidelobes can reduce any narrow-band noise and improve all the g_n (and particularly those close to the narrowband noise). Whether used for transmitter, receiver, or both, Figure 4.41 shows that the first few sidelobes are about the same no matter how much excess bandwidth (extra cyclic extension) is used. However, for somewhere between 5% and 25% excess bandwidth, 20 dB and more rejection of noise energy coupling through sidelobes is possible, which may be of considerable benefit in certain applications. Of course, this extra bandwidth is the same as any excess bandwidth used by the transmitter (so excess bandwidth does not increase twice for transmitter and receiver windowing).

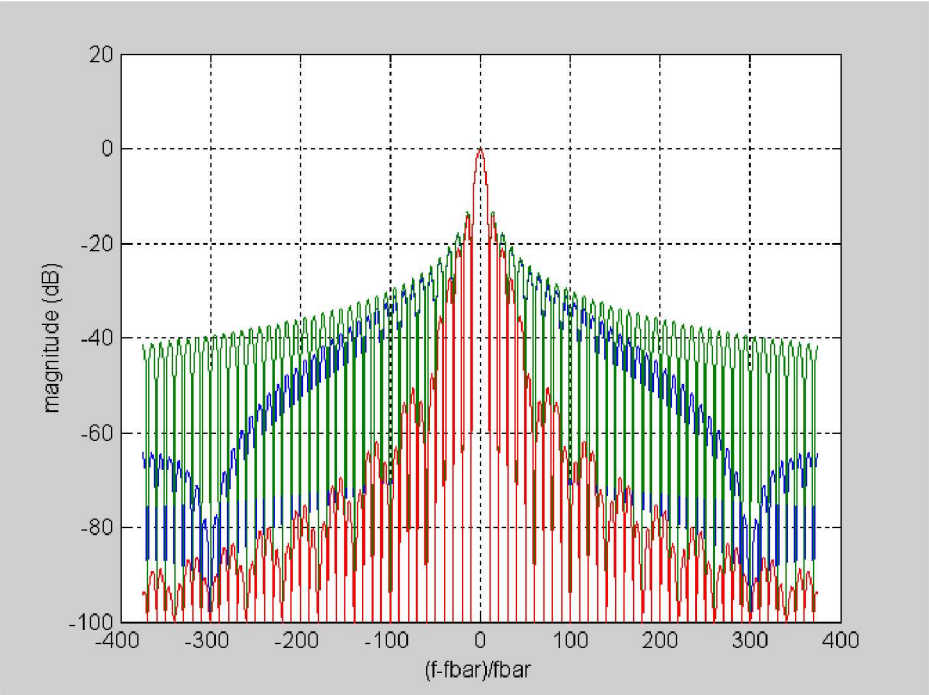


Figure 4.41: Illustration of transforms of rectangular window (green) and raised-cosine with 5% (blue) and 25% (red) excess bandwidth versus $(f - \bar{f})/\bar{f}$.

4.10 Peak-to-Average Ratio reduction

The peak-to-average ratio **of an N-dimensional constellation** was defined in Chapter 1 as the ratio of a maxium magnitude constellation point to the energy per dimension. That definition is useful for $N = 1$ and $N = 2$ dimensional constellations. For the higher dimensionality of DMT and OFDM signals (or vector coding or any other partitioning), the maximum time-domain value is of practical interest:

$$x_{max} = \max_{k\ ,\ k=0,\ldots N+\nu-1} |x_k| \quad . \tag{4.365}$$

This value determines the range of conversion devices (DACs and ADCs) used at the interfaces to the analog channel. The number of bits needed in the conversion device often determines its cost and also its power consumption. Thus, smaller is usually better (especially at high speeds). A higher ratio $x_{max}/\bar{\mathcal{E}}_{\boldsymbol{x}}$ leads to more bits in such a device. Also, subsequent analog circuits (amplifiers, filters, or transformers) may exhibit increased nonlinearity as x_{max} increases. Finally, the power consumption of linear amplifiers depends on the peak power, not the average power. In the analog portion of the channel between conversion devices, actually x_{max} should be redefined to be

$$x(max) = \max_{t\in[0,T)} |x(t)| \quad , \tag{4.366}$$

which can exceed the discrete-time x_max. High-performance transmission systems necessarily increase the peak voltages in theory, so the PAR for the transmitted signal $x(t)$ (and/or x_k) can become important to designers.

This section investigates this peak value and shows that additional digital signal processing can reduce the large peak.

Subsection 4.10.1 begins with some fundamentals on the probability of peaks and their relationship to data rate, showing that indeed very little data rate need be lost to eliminate large peaks in transmission. Subsection 4.10.2 then progresses to describe the most widely used and simplest PAR-reduction method known as "tone reservation" (developed independently by former 379 students and their colleagues!) where a few tones are used to carry "peak annihilating" signals instead of data. This method essentially requires no coordination between transmitter and receiver and thus is often preferred. The loss of "tones" is somewhat objectionable, so for more complexity and some receiver-transmitter understanding, the equally effective "tone-injection" method appears in Subsection 4.10.3. This method causes no data rate loss and instead cleverly inserts signals invisible to the data connection, but that annhilate peaks. Subsection 4.10.4 discusses the most complicated nonlinear-equalization methods that take the perspective of allowing clipping of signals to occur and then essentially remove the distortion caused by the clips at the receiver through an ML detector. All the approaches presented in this section reduce the probability of nonlinear distortion to a level that can be ignored in practice.[36]

A series of coded approaches have also been studied by various authors, where points in the N-dimensional constellation that would lead to time-domain large x_{max} are prohibitted from occuring by design. Unfortunately, these approaches while elegant mathematically are impractical in that they lose far too much data rate for effective performance, well above theoretical minimums. This book will not consider them further.

4.10.1 PAR fundamentals

This book chooses to focus on real signals x_k so that $X^*_{N-i} = X_i$ for each symbol. For complex signals, the theory developed needs to be independently applied to the real dimension and to the complex dimension.[37]

[36] That is a level at which outer forward error-correction codes, see Chapters 10 and 11, will remove any residual errors that occur at extremely infrequent intervals.

[37] In which case, the signal processing is independently applied to the even part of $\boldsymbol{X}$, $X_{e,i} \triangleq .5 \cdot (X_i + X^*_{N-i})$ that generates the real part of x_k and then again to the odd part of $\boldsymbol{X}$, $X_{o,i} \triangleq -.5 \cdot (X_i - j \cdot X^*_{i-i})$ that generates the imaginary part of x_k.

First, this subsection considers the discrete-time signal x_k and generalizes to over-sampled versions that approximate $x(t)$. The time-domain signal x_k is again

$$x_k = \sum_{n=0}^{N-1} X_n \cdot e^{-\jmath \frac{2\pi}{N} nk} \quad , \tag{4.367}$$

which is a sum of a large number of random variables. Losely speaking with the basic Central Limit Theorem or probability, such a sum is Gaussian in distribution and so has very small probability of very large values. For the Gaussian distribution, a value with 14.2 dB of peak-to-average ration ($Q(14.2dB)$) has a probability of occurrence of 10^{-7} and a PAR of 12.5 dB has occurrence 10^{-5}. When such a peak occurs, with high-probability large "clip" noise occurs if the conversion devices or other analog circuits cannot handle such a large value. So, with probabilities on the order of the probability of bit-error, entire DMT/OFDM symbols may have many bit errors internally unless the PAR in the design of such circuits accommodates very high PAR. Typically, designers who use no compensation set the PAR at 17 dB to make the occurrence of a peak clip less likely than the life-time of the system (or so small that it is below other error-inducing effects).

One might correctly note that the same high-peak problem occurs in any system that uses long digital filters (i.e., see the transmit-optimizing filters for QAM and PAM systems in Chapter 5) will have each output sample of the filter equal to a sum of input random variables. For effective designs, the filter is long and so the Gaussian nature of actual transmitted signal values is also present. Furthermore aside from filters or summing effects, best coding methods (particularly those involving shaping) provide transmit signal and constellation distributions that approximate Gaussian (see Chapter 8). Thus, one can find any well-designed transmission system having the same high PAR problem. However, solutions to this problem are presently known only for the DMT/OFDM systems (which with the solutions, actually then have a lower PAR than other systems).

At first glance, it might seem prospects for high performance in practice are thus dim unless the designer can afford the higher dyanmic range and power consumption of conversion and analog circuits. However, let us assume a PAR of say 10 dB is acceptable (even sinusoids in practice require 3-4 dB so this is say one bit of range[38] more than a non-information-bearing signal would require. Such a peak in a Gaussian distribution occurs with probability 10^{-3}. Let us suppose that internal to the data transmission stream, a small amount of overhead data rate was allocated to capturing the size of the clip that occurs in analog circuits and sending a quantized version of that size to the receiver. No more than 20 bits per clip would be required to convey a very accurate description, and perhaps just a few bits if designs were highly efficient. Those twenty bits would represent on average only 2% of the data rate. Clearly not much is lost for imposing maximums on the transmitted range of quantities, at least from this trivial fundamental perspective. Such an approach might have significant delay since the overhead bit stream would be very low with respect to other clocks in the design. Nonetheless, it fundamentally indicates that significant improvement is possible for a small loss in overall data rate.

This is exactly the observation used in the three following approaches of this section.

4.10.2 Tone Reservation

Tone reservation allows the addition of a **peak annihilator** $\boldsymbol{c}$ to the transmit signal $\boldsymbol{x}$ on each symbol. Thus, $\boldsymbol{x} + \boldsymbol{c}$ is cyclically extended and transmitted. The peak-annihilator attempts to add the negative of any peak so that the signal is reduced in amplitude at the the sampling instant of the peak. The peak-annihilation vector $\boldsymbol{c}$ is constructed by frequency-domain inputs on a few tones in set $\mathcal{I} = \{i_1, i_2, ..., i_r, ..., N - i_r, ..., N - i_1\}$. Typically $r << N/2$. The determination of this set is deferred to the end of this subsection.

Thus,

$$\boldsymbol{x} + \boldsymbol{c} = \mathcal{Q}^* \left[\boldsymbol{X} + \boldsymbol{C}\right] \tag{4.368}$$

and

$$C_n = \begin{cases} 0 & n \notin \mathcal{I} \\ \text{selected} & n \in \mathcal{I} \end{cases} \quad . \tag{4.369}$$

[38] Conversion devices basically provide 6 dB of dynamic range increase for each additional bit used to quantize quantities.

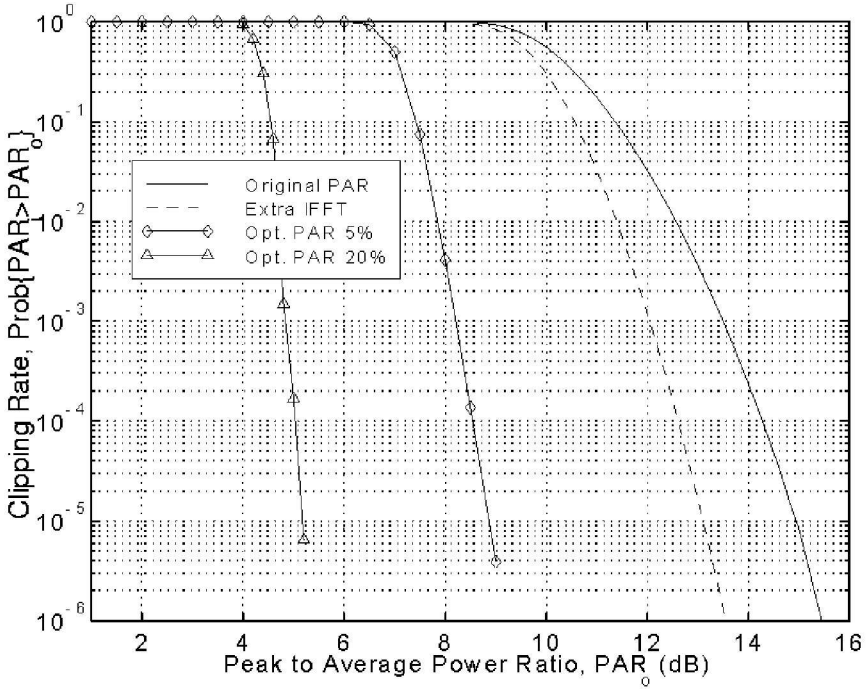


Figure 4.42: Probability of peak above specified PAR level for $N = 128$ OFDM/DMT systems with 5% and 20% of tones reserved. The "extra FFT" curve represents a double-pass of the transmitter IDFT where the phase of some signals has been selectively altered on the second pass to reduce PAR for comparison.

Thus, $X_n \neq 0$ when $n \notin \mathcal{I}$, leaving $N - r$ tones available for data transmission and the others reserved for selection and optimization of the peak annihilator.

The problem can be formulated as

$$\min_{\boldsymbol{C}} max_k |\boldsymbol{x} + \mathcal{Q}^* \boldsymbol{C}| \tag{4.370}$$

with the constraints that $C_n = 0$, $\forall n \in \mathcal{I}$ and $\|\boldsymbol{x} + \mathcal{Q}^* \boldsymbol{C}\|^2 \leq N\bar{\mathcal{E}}_{\boldsymbol{x}}$. This problem can be recognized with some work as equivalent to a linear-programming problem for which algorithms, although complex, exist for solution. This subsection shows the performance of such exact solution to see potential benefit, but rapidly progress to an easily implemented algorithm that approximates optimum and is used in practice. Figure 4.43 illustrates the reduction in probability of a peak exceeding various PAR level. For reservation of 5% of the tones, a 6 dB reduction (savings of one bit is possible). For 20% reservation (perhaps an upper limit on acceptable bandwidth loss), a 10 dB reduction is possible. Indeed, the PAR achieved for this OFDM/DMT system is well below those of systems that perform worse. However, the complexity of the solution used would be burdensome. Thus, the alternative method of minimizing squared clip noise is subsequently derived.

The alternative method makes use of the function

$$clip_A(x_k) = \begin{cases} x_k & |x_k| < A \\ A \cdot \mathrm{sgn}(x_k) & |x_k| \geq A \end{cases} , \tag{4.371}$$

where $A > 0$, and x_k and A are presumed real. The "clip noise" for a symbol vector $\boldsymbol{x}$ is then

$$\delta^2_{\boldsymbol{x}} = \|\boldsymbol{x} - clip_A(\boldsymbol{x})\|^2 = \sum_{k=0}^{N-1} [x_k - clip_A(x_k)]^2 \quad . \tag{4.372}$$

With the peak-annihilation in use

$$\delta^2_{\boldsymbol{x}+\boldsymbol{c}} = \|\boldsymbol{x} - clip_A(\boldsymbol{x} = \mathcal{Q}^*\boldsymbol{C})\|^2 \quad . \tag{4.373}$$

The gradient of the "clip-noise" can be determined when a peak-annihilator is used:

$$\nabla_{\boldsymbol{c}}\delta^2_{\boldsymbol{x}+\boldsymbol{c}} = 2 \cdot \sum_{k \ni |x_k+c_k|>A \& i \in \mathcal{I}} \mathrm{sgn}(x_k + c_k) \cdot [|x_k + c_k| - A]\, \mathcal{Q}^*_{\mathcal{I}} \cdot \bar{\boldsymbol{q}}^*_k \tag{4.374}$$

where $\mathcal{Q}^*_{\mathcal{I}}$ is the $N \times 2r$ matrix of columns of $\mathcal{Q}^*$ that correspond to the reserved tones, and $\bar{\boldsymbol{q}}_k$ is an $2r \times 1$ column vector that is the hermetian transpose of the k^{th} row of $\mathcal{Q}^*_{\mathcal{I}}$. The $N \times 1$ vectors

$$\boldsymbol{p}_k = \mathcal{Q}^*_{\mathcal{I}}\bar{\boldsymbol{q}}_k \tag{4.375}$$

can be precomputed as they are a function only of the IFFT matrix and and the indices of the reserved tones.[39] These vectors are circular shifts of a single vector. They will have a "peak" in the k^{th} position and thus changing the sign of this peak and adding it to the original $\boldsymbol{x}$ to be transmitted reduces the PAR. Such an operation could occur for each and every k such that $|x_k + c_k| > A$. That is essentially what the gradient above computes – a direction (and magnitude) for reducing the clip noise in those dimensions where it occurs.

A steepest-descent gradient algorithm for estimating the best $\boldsymbol{c}$ then sums a (small) scaled sum of gradients successively computed. Defining

$$\epsilon_k = \mathrm{sgn}(x_k + c_k) \cdot [|x_k + c_k| - A] \;\; \forall k \ni |x_k + c_k| > A \tag{4.376}$$

as a type of successive error signal, then the gradient algorithm becomes

$$\boldsymbol{c}_{j+1} = \boldsymbol{c}_j - \mu \cdot \sum_{k \ni |x_k+c_k|>A} \epsilon_k(j) \cdot \boldsymbol{p}_k \quad , \tag{4.377}$$

where j is an algorithm index reflecting how many iterations or gradients have been evaluated. This algorithm converges to close to the same result as the optimum linear-programming solution.

After approximately 10 iterations, the PAR has been reduced by 6 dB. After 40 iterations, the algorithm is within .5 dB of best performance with 5% of tones used. These calculations were for $N = 128$ and represent resulting from averaging over a variety of data vectors(and of course peaks only occur for a small fraction of those random data vectors).

For most applications, the indices in $\mathcal{I}$ can be randomly chosen among those tones that have non-zero energy allocated in loading (use vectors that have zero energy at the channel output usually leads to a peak reproduced at the output of the channel even if the input had no such peak, which means the ADC will saturate in the receiver). Tellado has studied methods for optimizing the set $\mathcal{I}$ but came to the conclusion that simple random selection the indices of to get 5% of the used tones is sufficient. (And if the resultant vector to be circularly shifted does not have a "peak" in it, a second random selection is then made.)

This method, largely due to Tellado, but also studied at length by Gatherer and Polley, is the predominant method used for PAR reduction in practice today. Its lack of need for coordination between transmitter and receiver (the transmitter simply forces the receiver to off-load the tones it wants by sending "garbage" data on those tones that forces swaps away from them). Thus, there is no need to specify the method in a DMT standard. However, in OFDM standards the set of used tones may need agreement before hand. Indeed since peaks do not occur often, the tones used for pilots in OFDM may also carry the very infrequent peak-reduction signals without loss of performance of the (well-designed) receiver PLLs.

In continuous time, peaks can occur at time instants other than those corresponding to the DMT sampling rate. Searches for peaks then may interpolate the IFFT output samples in the transmitter and the reduction waveform might also be interpolated when designed. All follows in a straightforward manner at 2 to 4 times the DMT/OFDM sampling rate in digital signal processing, but the interpolation can add complexity to this signal processing.

[39] These terms arise in the gradient vector because of the restriction to zero of the non-reserved tones and would be deleted from the gradient altogether if there were no restrictions on $\boldsymbol{C}$ or $\boldsymbol{c}$.

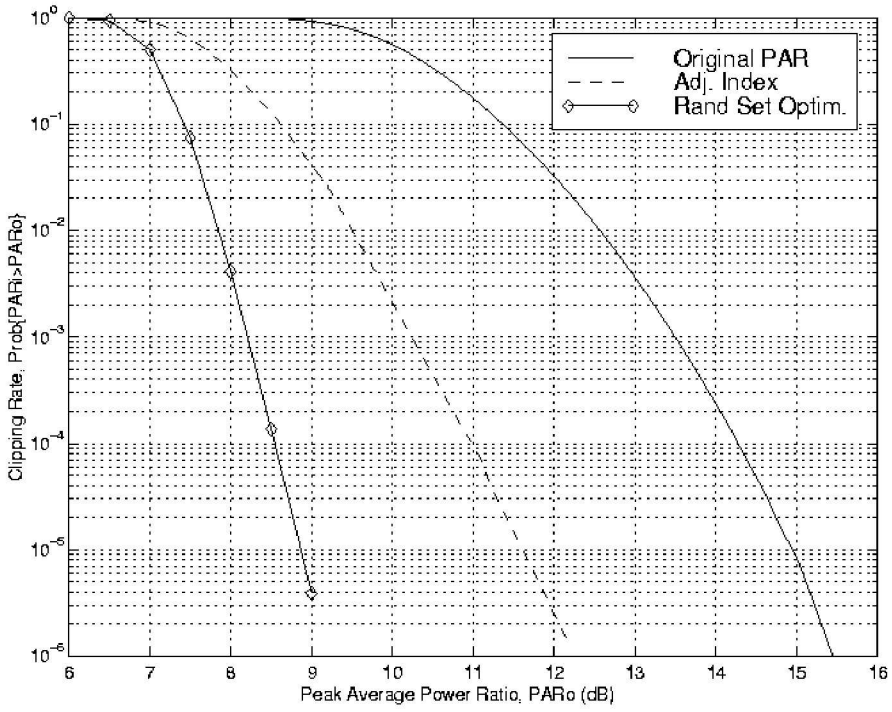


Figure 4.43: Probability of peak above specified PAR level for successive iterations of the gradient algorithm.

4.10.3 Tone Injection

A fraction of reverved tones that exceeds 5% is considerably in excess of the minimums theoretical requirement of bandwidth loss to reduce PAR substantially. Tellado, and independently Kschichang, Narula, and Eyuboglu of Motorla, found means of adding peak-annihilation signals to the existing data-carrying tones.

In tone injection, no tones are reserved for only peak-reduction signals. Instead, signals are added to the data symbols already present on some tones in a way that is transparent to the receiver decision process. The basic concept is illustrated in Figure 4.44. There a selected constellation point called A can be replaced by any one of 8 equivalent points if the 16QAM constellation shown were repeated in all directions. Each of these 8 points has greater energy and increases the transmitted energy for that particular tone if one of these is selected. The basic idea of tone injection is to search over points like the 8 shown for several tones to find an added signal that will reduce the peak transmitted. Because peaks do not occur often and because the extra energy only occurs on a few tones, the penalty of tone injection is a very small increase in transmit power, typically .1 dB or less on average. A knowledgeable receiver design would simply decode the original point or any of the 8 possible replacement points as the same bit combination. The offset in either horizontal or vertical direction is $\bar{d}$ units.

Thus, peaks in time domain are traded for peaks in the frequency domain where they have little affect on the transmitted signal energy, and in particular a very large power-reducing effect on the consumed power of the analog driver circuits that often depend on the peak signal transmitted.

More formally, the time domain DMT or OFDM signal (avoiding DC and Nyquist, which are simple extensions if desired and letting $R_n \triangleq \Re\{X_n\|$ and $I_n \triangleq \Im\{X_n\|$) is

$$x(t) = \frac{2}{\sqrt{N}} \cdot \sum_{n=1}^{N/2-1} R_n \cdot \cos(\frac{2\pi}{NT'}nt) - I_n \cdot \sin(\frac{2\pi}{NT'}nt) \tag{4.378}$$

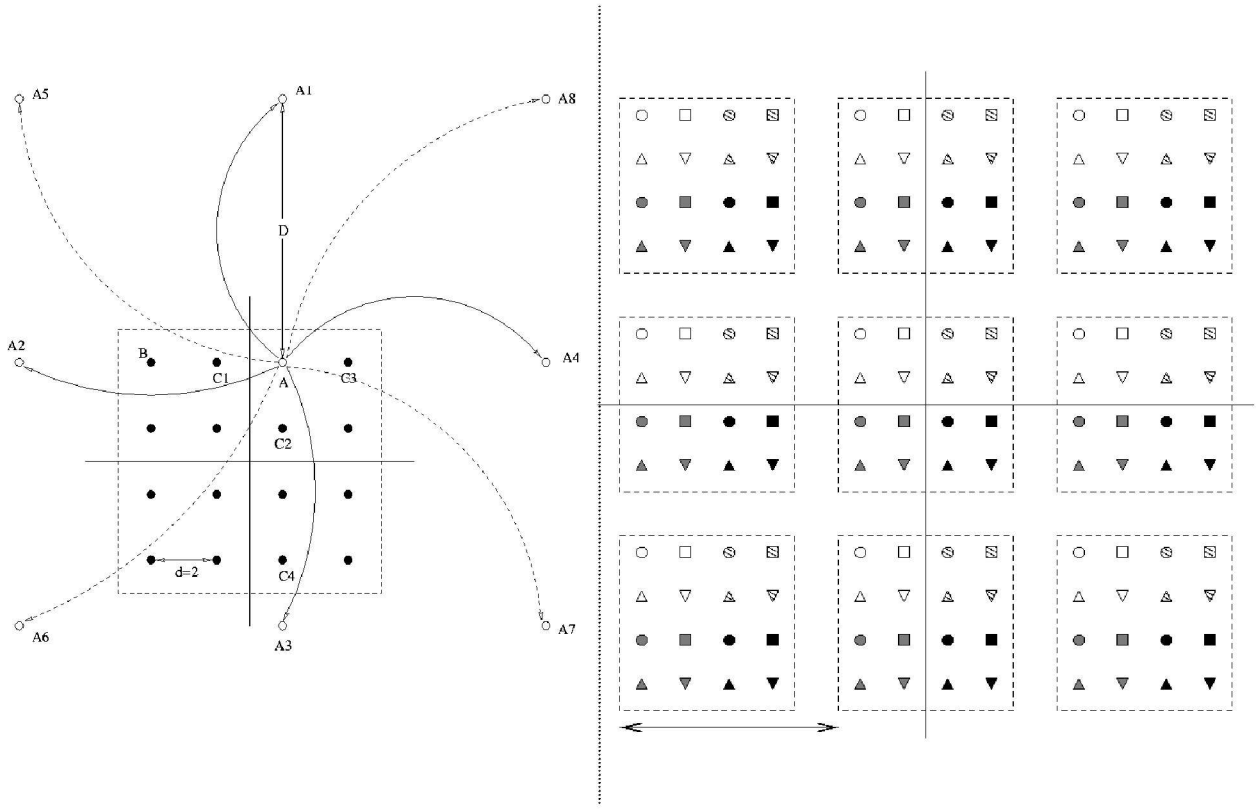


Figure 4.44: Illustration of constellation equivalent points for 16QAM and Tone Injection.

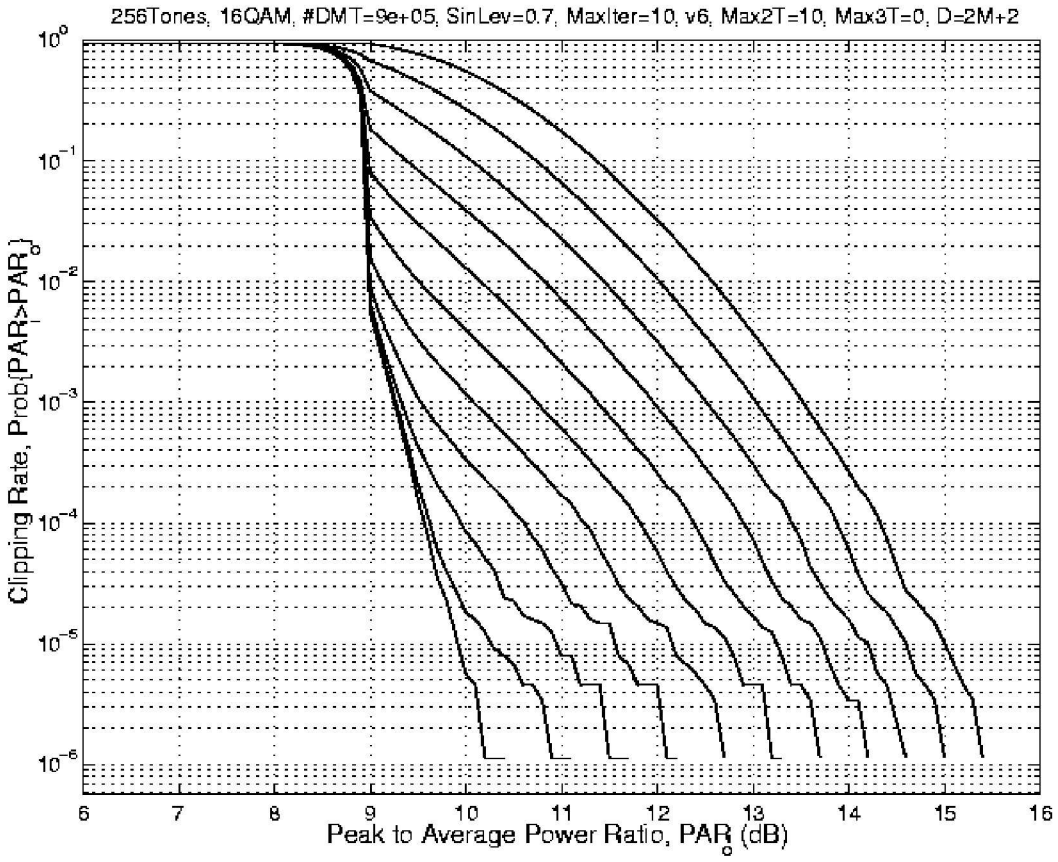


Figure 4.45: Illustration of successive iterations of tone injection.

over one symbol period ingnoring the cyclic prefix. $x(t)$ may be sampled at sampling period $T' = T/(N+\nu)$ or any other higher sampling rate, which is denoted $x_k = x(kT')$ here. Over a selected set of tones $n \in \mathcal{I}$, x_k can be augmented by injecting the peak-annihilation signal

$$\tilde{x}_k = x_k \pm \frac{2 \cdot \bar{d}}{\sqrt{N}} \cos(\frac{2\pi}{NT'} nkT') \pm \frac{2 \cdot \bar{d}}{\sqrt{N}} \sin(\frac{2\pi}{NT'} nkT') \ \ \forall n \in \mathcal{I} \quad . \tag{4.379}$$

It is not necessary often to add both the cos and sin terms, as one is often sufficient to reduce the peak dramatically. Algorithms for peak injection then try to find a small set of n indices at any particular symbol for which an injected peak signal is opposite in sign and rougly equal in magnitude to the largest peak found in the time-domain symbol. An exhaustive search would be prohibitively complex. Instead, typical algorithms look for indices n that correspond to large points on the outer ranges of the constellation used on tone n_0 having been selected by the encoder and for which also the cos or sin is near maximum at the given peak location k_0. Then, roughly,

$$|\tilde{x}_{k_0}| = |x_{k_0}| - 2\bar{d}/\sqrt{N} = |x_{k_0}| - \frac{2}{\sqrt{N}} \cdot \sqrt{\frac{6 \cdot 2^{2\bar{b}_{n_0}}}{2^{2\bar{b}_{n_0}} - 1}} \quad . \tag{4.380}$$

On each such iteration of the algorithm, a peak is reduced at location k_i, $i = 0, ...$ until no more reduction is possible or a certain PAR level has been achieved. Figure 4.45 illustrates successive steps of such tone reduction applied to an OFDM system with 16 QAM on each tone. About 6 dB of PAR reduction is obtained if enough iterations are used. The transmit power increase was about .1 dB.

4.10.4 Nonlinear Compensation of PAR

Tellado also studied the use of receivers that essentially store the knowledge of nonlinear mapping caused by peaks that exceed the linear range of the channel in a look-up table. He then proceeded to implement a maximum likelihood receiver that searches for most likely transmitted signal. This method is very complex, but has the advantage of no bandwidth nor power loss, and is proprietary to recievers so need not be standardized as would need to be the case for at least the silenced tones in tone reservation and the modulo-reception-rules for tone injection.

Exercises - Chapter 4

4.1 *Short Questions*

a. (1 pt) Why is the X_0 subsymbol in Figure 4.2 chosen to be real?

b. (1 pt) Why is the $X_{\bar{N}}$ subsymbol in Figure 4.2 chosen to be one-dimensional?

c. (3 pts) Given ideal sinc transmit filters, $\phi_n = (1/\sqrt{T})$ sinc (t/T), why are the channel output basis functions of multitone, $\phi_{p,n}$, orthogonal for all possible impulse responses $h(t)$?

4.2 *A simple example on Multitone*

(3 pts) Redo Example 4.1 with $H(f) = 1+0.5e^{j2\pi f}$ (Suppose $T = N$). The rest of the specifications are : $\frac{\bar{\mathcal{E}}_{\boldsymbol{x}} \cdot \|h\|^2}{\sigma^2} = 10$dB, design target arg. of Q-function = 9dB, $\bar{E}_x = 1$, $N = 2\bar{N} = 8$. As in the example, the last sub-channel is silenced. Also, attempt a design with $N = 2\bar{N} = 16$. The bit rate of your new design will go down if correctly designed. Why?

4.3 *GAP Analysis*

a. (1 pt) While the SNR gap may be fixed for often-used constellations, this gap is a function of two other parameters. What are these two parameters?

b. (2 pts) Based on your answer to part (a), how can the gap be reduced?

c. (2 pts) For an AWGN channel with SNR = 25dB and QAM transmission, $\bar{P}_e = 10^{-6}$, and $b = 4$. What is the margin for this transmission system?

d. (2 pts) Let $b = 6$ for part (c), how much coding gain is required to ensure $\bar{P}_e = 10^{-6}$?

4.4 *Geometric SNR*

a. (1 pt) For the channel in Problem 4.2, calculate the $SNR_{m,u}$ using the exact formula and the approximate one, SNR_{geo} (for a gap, use $\Gamma = 8.8$dB).

b. (1 pt) Are these two values closer than for the channel $1 + .9D^{-1}$? Explain why.

c. (3 pts) Compare the difference of the SNR_{MFB} to the exact $SNR_{m,u}$ for the $1 + .5D^{-1}$ and $1 + .9D^{-1}$. Which one is closer? Why?

4.5 *Water-Filling Optimization*

Repeat Problem 4.2 using water-filling optimization. Given spec is $H(f) = 1 + .5e^{j2\pi f}$, $SNR_{MFB} =$ 10dB, $\Gamma = 1$(0dB). (Note that all the subchannels may be used in this problem)

a. (2 pts) Calculate the optimal distribution of energy for the sub-channels and the maximum bit rate assuming that the gap, $\Gamma = 1$(0dB).

b. (1 pt) Calculate the gap for PAM/QAM that produces an arg. of the Q-function equal to 9dB. (The gap for $\bar{b} \geq 1$ is the difference between the SNR derived from capacity and the argument of the Q-function for a particular probability of error)

c. (2 pts) Calculate the optimal distribution of energy for the sub-channels and the maximum bit rate using the gap found in (b).

4.6 *Margin Maximization*

For the channel with $H(f) = 1+0.5e^{j2\pi f}$, as in Problem 4.2 maximize the margin using water-filling. (All the subchannels may be used in this problem). Let N=8 for this problem.

a. (2 pts) Is transmission of uncoded QAM/PAM at $P_e = 10^{-7}$ at data rate 1 possible? (i.e. is the margin ≥ 0)? If so, what will the margin be?

b. (1 pt) For data rate 1, what gap provides zero margin? For uncoded PAM/QAM, what $\bar{P}_e$ corresponds to zero margin?

c. (2 pts) What is the margin if $\Gamma = 1$(0dB)?

d. (1 pt) What is the margin if Γ = margin in part (c) (dB)?

4.7 *Creating a DMT Water-filling Program*

Attach the appropriate program for this problem. On plots versus N, you may elect to plot only for even N values in this program. The sampling period is presumed to be $T' = 1$. If you use the already provided programs in the text, then indicate in the margins of the print-out the various mathematical formulas that correspond to the programming steps.

a. (4 pts) Write or use a program to do DMT-RA water-filling on an arbitrary FIR channel with the pulse response given as a row vector (ex. $h = [111 - 1]$ for $1 + D + D^2 - D^3$ or $1 + D^{-1} + D^{-2} - D^{-3}$ channel). ν will be determined as the length of your FIR channel-input vector. The input parameters should be $\frac{\mathcal{N}_0}{2}$, $\bar{\mathcal{E}}_x$, the number of subchannels and the gap, Γ. (Hint, try using the flow charts in Figures (4.7) - (4.9)). What are $\bar{b}$ and the energy distributions for RA loading on the $1 + .9D^{-1}$ channel? (N = 8, $\Gamma = 1$ (0dB), $\bar{E}_x = 1$ and $\frac{\mathcal{N}_0}{2} = .181$)?

b. (2 pts) plot $\bar{b}$ vs. N for the channel $1 + D - D^2 - D^3$. ($\Gamma = 1$ (0dB), $\bar{E}_x = 1$ and $\frac{\mathcal{N}_0}{2} = .1$)

c. (2 pts) Modify your program (i.e. write another program) to do MA water-filling on the same FIR channel. What is the margin and the energy distribution for MA loading on the $1 + .9D^{-1}$ channel? (N = 8, $\Gamma = 1$ (0dB), $\bar{b} = 1$, $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$ and $\frac{\mathcal{N}_0}{2} = .181$)?

d. (2 pts) plot margin vs. N for the channel $1 + D - D^2 - D^3$ with $\bar{b} = 1$. ($\Gamma = 1$ (0dB), $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$ and $\frac{\mathcal{N}_0}{2} = .1$)

4.8 *Levin-Campello Algorithms*

Please show work. This problem executes the Levin-Campello loading algorithm for the $1 + 0.5D^{-1}$ channel with PAM/QAM, $\bar{P}e = 10^{-6}$, $SNR_{MFB} = 10dB$, N=8, and $\bar{E}_x = 1$. Use $\beta = 1$.)

a. (2 pts) Create a table of $e(n)$ vs. channel. (You may truncate the table sufficient to answer part (b) below).

b. (3 pts) Use the EF algorithm to make $\bar{b}$=1.

c. (1 pt) Use the ET algorithm to find the largest $\bar{b}$.

d. (1 pt) If the design were to reduce the b by 2 bits from part (c), use the EF and BT algorithms to maximize the margin. What is this maximum margin?

4.9 *Creating a DMT Levin-Campello Program*

Attach the appropriate program for this problem. On plots versus N, you may elect to plot only for even N values in this program. The sampling period is presumed to be $T' = 1$.

a. (4 pts) Examine, use, and/or understand the text's programs to do RA LC (EF and ET) on an arbitrary FIR channel with the pulse response given as a row vector (ex. $h = [111 - 1]$ for $1 + D + D^2 - D^3$ or $1 + D^{-1} + D^{-2} - D^{-3}$ channel). The input parameters should be $\frac{\mathcal{N}_0}{2}$, $\bar{\mathcal{E}}_x$, the number of subchannels and the gap, Γ. If you use the ones from the text, write formulas in the margin where appropriate that describe the associated mathematical actions in the text. What are $\bar{b}$ and the energy distributions for RA loading on the $1 + .9D^{-1}$ channel? (N = 8, $\Gamma = 1$ (0dB), $\bar{E}_x = 1$ and $\frac{\mathcal{N}_0}{2} = .181$)?

b. (2 pts) plot $\bar{b}$ vs. N for the channel $1 + D - D^2 - D^3$. ($\Gamma = 1$ (0dB), $\bar{E}_x = 1$ and $\frac{\mathcal{N}_0}{2} = .1$)

c. (2 pts) Modify your program (i.e. write another program) to do MA LC (EF and BT) on the same FIR channel. What is the margin and the energy distribution for MA loading on the $1 + .9D^{-1}$ channel? (N = 8, $\Gamma = 1$ (0dB), $\bar{b} = 1$, $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$ and $\frac{\mathcal{N}_0}{2} = .181$)?

d. (2 pts) plot margin vs. N for the channel $1 + D - D^2 - D^3$ with $\bar{b} = 1$. ($\Gamma = 1$ (0dB), $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$ and $\frac{\mathcal{N}_0}{2} = .1$)

4.10 *Sensitivity Analysis*

This problem looks at the sensitivity of E_n and $P_{e,n}$ (P_e for each sub-channel), to variations in the values of g_n for water-filling.

a. (1 pt) $S_{E_{ij}} = \frac{\delta E_i / E_i}{\delta g_j / g_j}$. Show that $S_{E_{ij}} = \frac{\Gamma(\delta_{ij} - 1/N^*)}{g_j E_i}$, where N^* is the number of sub-channels being used and δ_{ij} is the Kronecker delta. (Hint. Solve for the closed form expression for E given that N^* is known and then differentiate it.)

b. (2 pts) Calculate the worst case sensitivity for the $1 + 0.9D^{-1}$ and the $1 + 0.5D^{-1}$ for multitone with N=16 and $\Gamma = 3$dB. Which one do you expect to be more sensitive? Why?

c. (2 pts) Now, calculate bounds for the sensitivity calculated in (a). Show that

$$\frac{\Gamma(\delta_{ij} - 1/N^*)/g_{MAX}}{N^* \bar{E}/N^* + \Gamma(1/g_{harm} - 1/g_{MAX})} \leq S_{E_{ij}} \leq \frac{\Gamma(\delta_{ij} - 1/N^*)/g_{MIN}}{N^* \bar{E}/N^* + \Gamma(1/g_{harm} - 1/g_{MIN})}$$

where g_{harm} is the harmonic mean of g's.

(Hint: Replace $\mathcal{E}_i$ by its closed form expression and manipulate the expression.)

d. (1 pt) Show that $\delta A_n / \delta g_n \geq A_n / g_n$, where A_n is the argument of the Q-Function for the n^{th} sub-channel assuming that the bit-rate b_n is held constant. Interpret the result.
(Hint. Work with closed form expression for $g_n E_n$ instead.)

e. (2 pts) Find the sub-channel which has the worst sensitivity in terms of $P_{e,n}$ for $1 + 0.9D^{-1}$ and $1 + 0.5D^{-1}$ channels. Use $\Gamma = 3dB$, N=16, SNR_{MFB}=10dB for multitone.(Assume $\delta A_n / \delta g_n = A_n / (g_n)$).

4.11 *Complexity of Loading*

This problem uses 100 parallel channels(N=100) and evaluates different algorithm complexities: To compute a water-filling energy/bit distribution, we must solve (**??**) with the additional constraint of positive energies. An operation is defined as either an add, multiply, or logarithm. The word complexity denotes the number of operations.

For parts (a) and (b), this problem is primarily interested in the complexity as an order of N and N^*, (i.e. is it $O(N^2)$?) Also, comparison of part (a) with part (b) provides an idea of how much less complex Chow's algorithm is, so detailed complexities are also of interests, although an exact number is not necessary. Please use a set of reasonable assumptions (such as sorting takes $O(N \log N)$). Chow's algorithm here refers to Chow's on/off primer. For (c), assume that we start with $\boldsymbol{b} = [000...0]$, then do E-tight. The complexity would depend on the total number of b's, and β. So, you can express your answer in terms of b and β. It is difficult to compare (c) with (a) or (b), as they really solve different problems. Nevertheless, some reasonable estimates would be helpful.

a. (2 pts) Formulate an efficient algorithm that computes the water-filling energy/bit distribution. Compute the number of operations for water filling. How does this complexity depend on N^*?

b. (2 pts) What is the complexity for Chow's algorithm?

c. (2 pts) What is the complexity for LC algorithm?

d. (1 pt) Compare the different complexities calculated in (a),(b),(c).

4.12 *Dynamic Loading*

An ISI-channel with AWGN has a multitone transmission system that uses MA LC loading changes from $1 + D - D^2 - D^3$ to $1 + D - .1D^2 - .1D^3$. The energy per dimension is $\bar{\mathcal{E}}_{\boldsymbol{x}} = 2$ and $\bar{b} = 1$ with $\frac{\mathcal{N}_0}{2} = .1$. Use $\beta = 1$ and a gap corresponding to uncoded PAM/QAM with $P_e = 10^{-6}$.

a. (2 pts) Find the number of "bit-swaps" that occur because of this change when $N = 8$ and $N = 12$. (You may assume infinite-length basis functions and that the SNR is approximated by using the fourier transform value in the center of each band as in the examples in Sections 4.1-4.3.)

b. (4 pts) Find the information that is transmitted from the receiver to the transmitter for each of the swaps in part a. Specifically, find the tone indices and the gain-factor changes associated with the add and delete bit positions.

4.13 *VC for the* $1 + 0.5D^{-1}$

For the $1 + .5D^{-1}$ channel, $N = 8$, $\bar{E} = 1$, $\frac{\mathcal{N}_0}{2} = .125$, $T' = 1$, and $\Gamma = 3$dB

a. (2 pts) Solve for the VC subchannels, g_n.

b. (2 pts) Find the optimal RA bit Water-filling distribution. What is the bit rate?

c. (2 pts) Find the optimal MA bit Water-filling distribution for $\bar{b} = 1.25$. What is the margin?

4.14 *DMT for the* $1 + 0.5D^{-1}$

For the $1 + .5D^{-1}$ channel, $N = 8$, $\bar{E} = 1$, $\frac{\mathcal{N}_0}{2} = .125$, $T' = 1$, and $\Gamma = 3$dB

a. (2 pts) Solve for the DMT gains, P_n, and the optimal water-filling RA bit distribution. What is the bit rate?

b. (2 pts) Compare the gains P_n to the gains λ_n of vector coding for the same channel. Which one is better in terms of the ratio of largest to smallest bit assigned? Why is that?

c. (3 pts) Plot the Fourier spectra (i.e. the fft with appropriate zero-padding) of the columns of M for VC and for DMT in case of N=8,16,32. What is the change of pattern in spectra as N increases?

4.15 *Loading - 15 pts - Midterm 2000*

A DMT design is used on the channel with discrete-time response $P(D) = 1 - D^2$ with additive white Gaussian noise that has psd (discrete-time variance) $\sigma^2 = .181$ and the transmit energy per dimension is $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$. Loading will be executed on this channel for $N = 16$ and $\nu = 2$. (Hint think carefully if anything about this channel looks familiar and might save you considerable time in executing the computations - for instance, what do you know about even and odd sample times at the output?)

a. Sketch the magnitude of the Fourier Transform of the channel, $|P(e^{-\jmath\omega T})|$ and find the channel SNR's, g_n, at all frequencies of interest to DMT loading. (3 pts)

b. Execute waterfill rate-adaptive loading for this channel with a 0 dB gap. (3 pts)

c. Round the distribution that you obtained in part b so that the granularity is $\beta = 1$ bit per two dimensions on complex subchannels and one-bit per dimension on one-dimensional subchannels and find the new data rate. (1 pt)

d. Now use the LC algorithm to compute a RA data rate for $\beta = 1$. Be sure to show your incremental energy table. (4 pts)

e. Find a MA bit distribution with $\beta = 1$ for transmission with 8.8 dB gap at a data rate corresponding to $\bar{b} = 1$, and the corresponding margin. (4 pts)

4.16 *Generalized Nqyuist*

a. (2 pts) For what $N \geq 1$ does multitone transmission satisfy the generalized Nyquist criterion and symbol rate $1/T$ on any linear ISI additive white Gaussian noise channel?

b. (2 pts) What could the receiver do to attempt to correct the situations (that is, the N values) for which the GNC in part (a) is not satisfied.

c. (2 pts) How does the N necessary for a channel depend on the channel characteristics?

4.17 *Periodic Channels*

The simplest possible case of finding eigenfunctions is for the periodic channel; for which this problem seeks to prove various interesting facts. These facts will be useful for the investigation of Discrete Multitone Transmission. $r(t)$ is the channel autocorrelation function

a. (2 pts) Suppose $r(t) = r(t+T)$, show $\phi_n(t) = e^{j*\frac{2\pi}{T}nt}$ are eigenfunctions of the channel autocorrelation function. (Hint: express $r(t)$ in a Fourier series.) By eigenfunction, this problem means that if $\varphi(t)$, $t \in [0,T)$, is used for a channel with periodic $r(t)$, the output $y(t)$, $t \in [0,T)$, will be a scaled version of $\varphi(t)$. The scaling factor is the eigenvalue. Note that here, both the input and output signals are restricted to the interval $[0,T)$.

b. (1 pt) What are the associated eigenvalues?

c. (2 pts) If the channel were not periodic, but had finite length (ν), how could the designer create a new set of functions $\hat{\phi}_n(t)$ from $\phi_n(t)$ such that part of the output signal looks periodic? (hint: think of cyclic prefix in DMT.)

Parts (a) and (b) show the eigenfunctions are independent of the channel, which is a very desirable property. However, this only happens with periodic channel. The question is, for $r(t)$ non-periodic, can the designer modify the input so that the output appears is as if the channel is periodic. More precisely, let $r_r(t)$ denote the autocorrelation of a periodic channel, $r(t)$ denote the non-periodic version (i.e. $r(t) = r_r(t)$ for $t \in [0,T)$, $r(t) = 0$ and the ends of $[0,T)$). The relation $\varphi(t) * r_r(t) = k\phi(t)$ is true, where k is the eigenvalue. Can the designer modify $\varphi(t)$ so that $\hat{\varphi}(t) * r(t) = k\phi(t)$, for $t \in [0,T)$? If so, the transmission system can then use the channel-independent eigenfunction. By "part of output signal looks periodic", we mean "the output signal looks as if it were created by a periodic channel." Hint: think guard period and extensions.

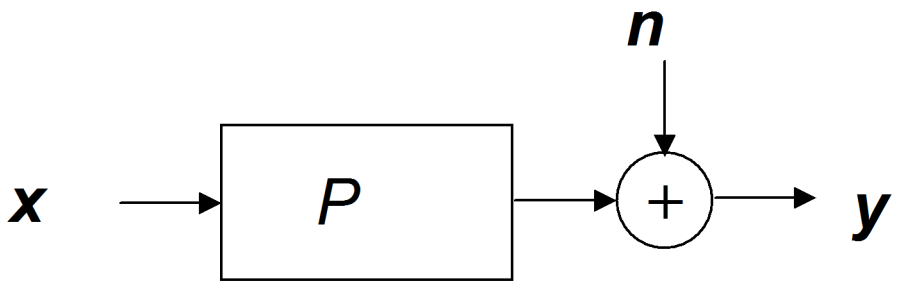
d. (1 pt) If the receiver only looks at the periodic part of the output signal, what would the associated eigenvalues be?

e. If a transmission system uses the channel-independent eigenfunctions developed in this problem, and thus provides the appearance of a periodic channel, how much more bandwidth does the design have to use, or in other words, how much faster does the sampling rate have to be?

4.18 *Vector Coding*

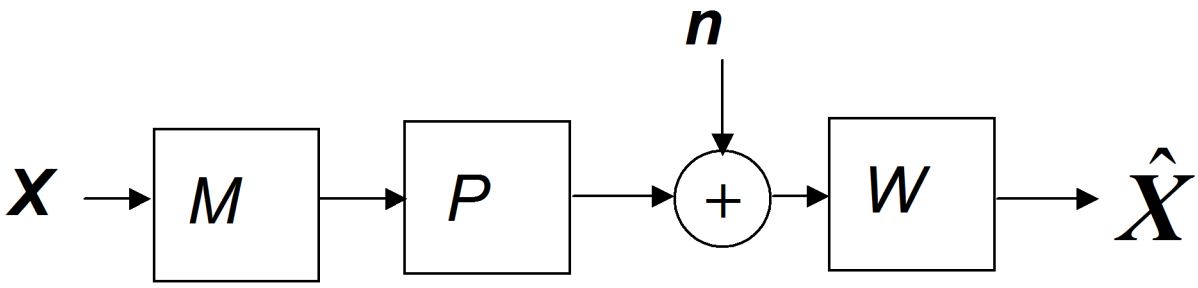
This problem concerns channel partitioning and Vector Coding.

a. (2 pts) Consider the following vector channel :



The system uses a matrix transmit filter, M, and a matrix receiver filter, F^*, such that $FF^* = MM^* = I$ i.e. the matrix filter doesn't change the energy.

Moreover, we want that $Y_n = \Lambda_n \cdot X_n + N_n$
Given that the channel is $1 + 0.5D$, calculate P, M, F^* and $\Lambda's$. Assume $\frac{\mathcal{N}_0}{2} = .125$, $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$, and $N = 8$.



b. (3 pts) Using the value of Λ's, calculate the optimal energy and bit distribution using a loading algorithm with $\Gamma = 3dB$.

c. (1 pt) What is the SNR_{vc} for this vector-coded system?

4.19 *Theoretical questions on Vector Coding*

a. (1 pt) Show that $\boldsymbol{N}$ in vector coding is indeed white and Gaussian with variance $N_0/2$.

b. (3 pts) Calculate $R_{\boldsymbol{yy}}$ and show that $|R_{\boldsymbol{yy}}| = \prod_{n=1}^{N}(\frac{\mathcal{N}_0}{2} + \mathcal{E}_n|\lambda_n|^2)$.

c. (2 pts) Verify $1 + \text{SNR}_{vc} = \left(\frac{|R_{\boldsymbol{yy}}|}{|R_{\boldsymbol{nn}}|}\right)^{1/N+\nu}$ assuming that the gap is 0 dB.

d. Prove that $F^* = \begin{bmatrix} 1 + \frac{1}{SNR_1} & & & 0 & 0 \\ & 1 + \frac{1}{SNR_2} & & & 0 \\ & & & . & . \\ 0 & & & . & \\ 0 & 0 & & . & 1 + \frac{1}{SNR_N} \end{bmatrix} \Lambda R_{\boldsymbol{Xy}} R_{\boldsymbol{yy}}^{-1}$

4.20 *Grasping Vector Coding → DMT*

Assume the channel is $1 + aD^{-1}$. This problem varies a and N to determine their effects on F, Λ and M.

a. (1 pt) For N=4 and a=0.9, compute F, Λ and M. Plot the modulus of the spectra of the eigenvectors of M, that is, take the columns of M as time sequences and calculate their abs(fft). Use at least 32 points for the fft's. Turn in this plot. What shape do these filters have?

b. (2 pts) Repeat part (a) for $N = 4$ and $a = 0.1$. Do you notice any differences in the resulting filter's ffts compared to (a)? What explains this behavior?
(Hint: Part (c) might be helpful)

c. (2 pts) Set $a = 0.9$ and $N = 8$. Calling λ_i the diagonal elements of the Λ matrix, calculate λ_0,$(\lambda_1 + \lambda_2)/2$, $(\lambda_3 + \lambda_4)/2$, $(\lambda_5 + \lambda_6)/2$ and compare to the values of H_n in Example 4.1.1. Explain the similarity.

d. For all the examples tested, matrix F seems to have some special structure. Describe it. How can you use this to reduce the number of operations in the receiver?

4.21 *Partitioning - 2002 Midterm*

A wireless transmission system is shown in Figure 4.46. Uncoded QAM with gap 8.8 dB at $P_e = 10^{-6}$ is used on each of 3 transmit antennae with the same carrier frequency but independent input data streams. Each has symbol rate 100 kHz and the energy per dimension at that rate is unity ($\mathcal{E}_x = 1$). There are also 3 receive antennae. All 9 paths shown are AWGN channels with independent noise of

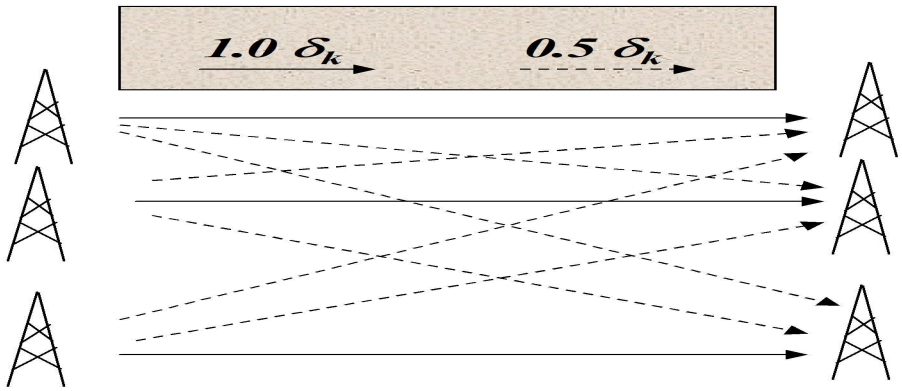


Figure 4.46: Wireless MIMO channel.

two-sided psd .01 on each. The gains are as shown. The "other" signals may be considered as noise for each of the 3 main signal paths from xmit antenna n to receive antenna n or all 3 signals may be jointly transmitted and received. This problem investigates the difference in approaches.

a. (2 pts) If the other two signals are considered to be additive Gaussian noise, what is the possible $\bar{b}$ and b as well as data rate R, according to the gap formula ? (You may assume there is a single receiver with background noise σ^2.) (2 pts)

b. (3 pts) Consider the possibility of co-generation of all three xmit antennae signals and co-reception of all 3 receive antennae signals. Show this can be described by a 3×3 channel matrix P. This is sometimes called a MIMO channel (multiple-input, multiple-output)

c. (1 pt) How many real dimensions are there in this MIMO channel? How many complex dimensions?

d. (5 pts) Use your answer in (b) to design a transmitter and a receiver that will provide 3 independent parallel channels. Show the transmit and receiver processing and specifically provide the entries in any matrix used. Also show subchannel gains g_n.

e. (1 pt) What is the best water-filling energy and corresponding number of bits for each of your channels in the answer to part d?

f. (3 pts) Determine $\bar{b}$ and b as well as R for your system in parts (d) and (e) and compare to your answer in part (a). What does this tell you about partitioning in space?

4.22 *Channel ID*

A DMT system uses the channel ID method of section 4.7. Slobhouse Designs Inc. has decided that 1dB accuracy in SNR measurement is sufficient.

a. (1 pt) Slobhouse decided to measure gain so as to induce less than 1dB error in SNR. How many training symbols (L) does SDI need?

b. (1 pt) If Slobhouse uses the same number (L) of symbols as in part a to do noise estimation, comment on SDI's noise estimation.

c. (3 pts) If L_1 is the number of measurements to estimate the gain, and L_2 is the number of measurements to estimate the noise, approximately estimate the minimum $L_1 + L_2$ such that the overall SNR error is less than 1 dB. There may be several solutions that are close in answer for this part.

4.23 *Channel ID - Midterm 2000*

Channel identification of Section 4.7 estimates channel SNR's for a system with symbol rate 4 Hz on a linear bandlimited AWGN channel. A periodic training signal is used. This problem estimates training time.

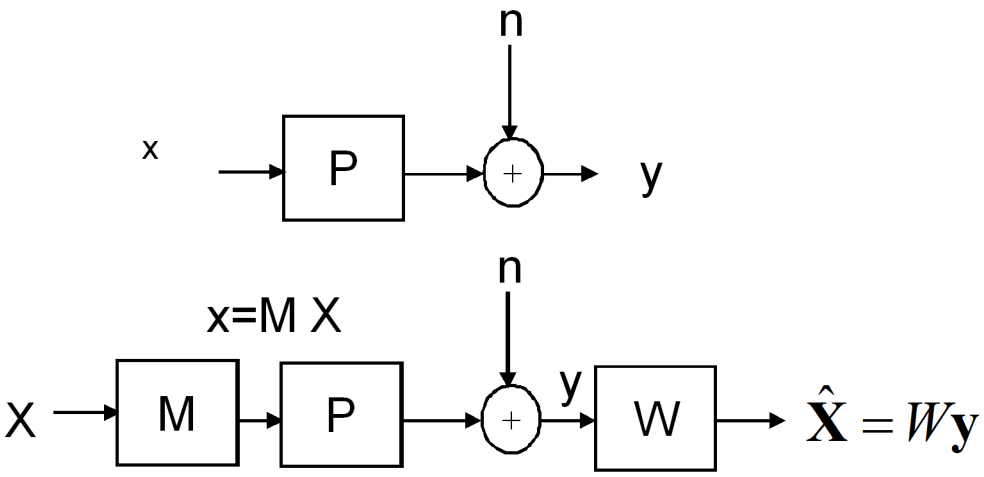
a. How many symbols of training are necessary for an average gain-estimation error of .25 dB (so that the variance of the gain estimation error is less than .25 dB away from the correct answer)? How long will this be? (2 pts)

b. Suppose the FFT size is $N = 8192$, but that the channel length is less than $\nu = 640$ sample periods. If the gain-estimation distortion is distributed evenly in the time domain over all 8192 time samples, but the result was windowed to 640 time-domain samples, what would be the new training time in symbol periods and in absolute time? (2 pts)

c. Suppose .25 dB noise estimation error is tolerable on the average (so that the standard deviation of the noise variance estimate is less than .25 dB away from the true noise). What is the step size of the steady-state FEQ-error based noise estimator, μ', with unit energy transmitted on the corresponding subchannel? (1 pt)

4.24 *Avoiding SVD in Vector Coding.*

Vector coding is used on the AWGN channel.



The receiver uses $W = R_{Xy}R_{yy}^{-1}$, a MMSE estimate for X given y. Then

$$\boldsymbol{X} = W\boldsymbol{y} - \boldsymbol{U} \qquad E\{\boldsymbol{U}\boldsymbol{U}^*\} = R_{\boldsymbol{U}\boldsymbol{U}}$$

a. (5 pts) Find W in terms of ε_n, λ_n, σ^2 and F. Look at your expression for W. What is the equalizer W doing to each sucbchannel? Why is this the same as minimizing MSE on each subchannel individually?

b. (1 pt) Show how the VC receiver W could be implemented adaptively, without explicit knowledge of P or F^* (pictorially and in few words).

Now, the hard part, implementing the transmitter without explicit knowledge of P or M.

c. (2 pts) Pretend $\boldsymbol{x}$ is a channel output and $\boldsymbol{Y}$ is a channel input. Write a vector AWGN expression for $\boldsymbol{x}$ in terms of $\boldsymbol{Y}$ and $\boldsymbol{u}$.

d. (2 pts) What is the MMSE matrix equalizer, call it H, for $\boldsymbol{Y}$ given $\boldsymbol{x}$?

e. (2 pts) If the receiver could, during training, communicate $\boldsymbol{Y}$ and SNR_n over a secure feedback path to the transmitter, show how the vector coding transmit matrix could be determined adaptively. (pictorially and in few words)

f. (1 pt) Do we need to do SVD to implement VC (is there an adaptive way to do this perhaps)?

4.25 *TEQ Analysis in a DMT system - Salvekar's Marathon*
A causal IIR channel has sampled-time response D-transform $\frac{1}{1-.9D}$.

The transmission system is an ISI-free DMT system only if ν goes to infinity. The goal is to reduce the ISI, while making sure that the noise is not enlarged greatly. In general, it may be difficult to derive the R_{xy} and R_{yy} matrices for a general IIR channel. However, we may have a program which solves the TEQ problem for arbitrary FIR channels. So, instead of simply doing all the calculations by hand, we will use the aforementioned programs. If the decay in the IIR channel is fast enough, we just need to take enough taps to ensure that our channel is 'OK'.

a. (3 pt) Check if the results of your program are the same as the results of the example (choose an FIR equivalent of the IIR channel, i.e truncate IIR channel into FIR channel with enough large number of taps such that FIR approximation is VERY close to IIR channel). Report the output of your programs in the form of R_{xy}, R_{yy}, $\boldsymbol{w}$,and $\boldsymbol{b}$.

(b) $\sim$ (h) Convert your program to find the PSD of the distortion. Note, that while your old program may not have computed which frequency bins were actually used, in this case, we need to know which bin is used. This means that the program best uses the fft instead of eig subroutines of matlab, since we are not sure which frequency bin corresponds to which eigenvalue. Furthermore, taking the eigenvalue decomposition of large matrices is very time consuming, especially when we know the eignevectors. For the following problems $\Gamma = 0$.

b. (1 pt) First, write an expression for the output of the channel after the teq. This should be in terms of $P(D)$, $W(D)$, $X(D)$ and $N(D)$.

c. (2 pt) Now, suppose the first $\nu + 1$ are the ones that the DMT system is synchronized with(i.e. the rest of the taps are ISI). Write an expression for the ISI+noise. Call the first $\nu + 1$ taps of $P(D) \cdot W(D)$ as $H'(D)$. The final form of the answer should be: $Err(D) = I(D)X(D) + N'(D)$, where $N'(D)$ is the noise through the system, and $I(D)$ is a function of $H'(D)$.

d. (2 pt) Prove that for a single pole P(D), P(D) = $(\frac{1}{1-aD})$, the I(D) will eventually go as $c * a^n$. When does this behavior begin?

e. (2 pt) If your noise and ISI are uncorrelated, what is the PSD of the error (write it in terms of D transforms)? What is this in the frequency domain, $S_{ee}(e^{jw})$?

f. (2 pt) In a DMT program, we need to specify the noise at each discrete frequency. We want to approximate the noise power in each bin by the sampled PSD. Write a program that will do this in matlab. The inputs into this program should be $\bar{E_x}, \sigma^2, I(D)$,and $W(D)$. As a check, let $\bar{E_x} = 1$, $\sigma^2 = .5$, N=10, $I(D) = .3D^4 + .1D^5$, $W(D) = 1 + .9D$. Using fft can make this very simple. Please present your values.

g. (3 pt) Now, change your DMT program so that it takes in the (sampled) error spectrum which we can get from the program in (f) as an argument. The DMT program will now do waterfilling Find $\bar{b}$ for an $\bar{E_x} = 1$, N=10, P(D)=1+.5D, and $S_{ee}(e^{jw})|_{\frac{2*pi*k}{N}} = |P[k]|^2$, where $S_{ee}(e^{jw})$ is the power spectrum of the error sequence and P[k] is the fft of the pulse response(i.e, what is $\bar{b}$ if the PSD of error is the same as the PSD of pulse response $(= P(e^{jw})|_{\frac{2*pi*k}{N}})$?).

h. (2 pt) Putting (f) and (g) together:
For the following parameters, what is the $\bar{b}$ in the DMT system? Choose the same setup as in example 10.6.1. Now, with $\Delta = 0$, let L=4, $\nu = 3$. Sweep N until we reach asymptotic behavior. Plot $\bar{b}$ vs. N. Also, do the same with the unequalized channel, remembering to include the error (i.e, we do not have TEQ, but still have the same ν for cyclic prefix!).

i. (1 pt.) Do (h) with $\bar{E_x}$=1000. What does this tell you about the teq?

4.26 *DMT Spectra*

This problem concerns the spectrum of a DMT signal. The transmit signal will be assumed to be periodic with period T.

a. (2 pts) Show that the frequency spectrum of this transmit DMT signal is a sum of impulses.

b. (3 pts) Show that multiplying the periodic transmit signal $x(t)$ by a window function $w(t)$ that has the value 1 for $0 \leq t \leq T$ and is zero elsewhere generates the non-periodic DMT signal studied in Section 4.4. Compute the spectrum of this new signal from the spectrum of the original periodic signal and the spectrum of the window.

c. (1 pt) Compare the spectrum of part (b) for two situations where $\nu > 0$ and $\nu = 0$.

d. (2 pts) Suppose a duplex channel has signals traveling in opposite directions on the same media. The designer wishes to separate the two directions by allocating different frequency bands to the two DMT signals, and separating them by lowpass and highpass filtering. Can you see a problem with such a duplexing approach? If yes, what is the problem?

4.27 *Isaksson's Zipper*

For the EPR6 channel $(1 + D)^4(1 - D)$,

a. (1 pt) Find the minimum length cyclic prefix to avoid inter-tone interference in DMT.

b. (2 pts) Find the minimal length of the cyclic suffix in terms of both channel length ν and channel end-to-end delay δ in sampling periods to ensure that both ends have symbols aligned for both the case of no timing advance and with timing advance.

c. (2 pts) Do the receiver or transmitter need any lowpass, bandpass, or highpass analog filters to separate the two directions of transmission (assuming infinite precision conversion devices)? Why or Why not?

d. (2 pts) Suppose two different systems with at least one side co-located (i.e., the transmitter and receiver for one end of each of the channels are in the same box). The two signals are added together because the channel wires "touch" at the co-located end. Could Zipper have an advantage in this situation? If so, what?

4.28 *Kasturia's Block DFE*

Subsection 4.6.7 describes the so-called Block DFE for channel partitioning. This problem investigates the improvement for this guardband-less transmission method for the $1 + .9D^{-1}$ channel of the notes. Use all parameter values as always for this channel, except that we will assume this channel is causal and $1 + .9D$.

a. (2 pts) Find the 8×8 matrix $\hat{P}$ for $N = 8$ with the BDFE, and its associated SVD for vector coding.

b. (2 pts) Using water-filling for loading, what is the SNR for this system of parallel channels? Compare to VC and DMT.

c. (2 pt) Draw a picture of the receiver for the BDFE in this case and indicate where the extra complexity enters.

d. (2 pts) Kasturia found a way in 1988 to derive a Tomlinson-like precoder for this channel. Explain essential features of such a precoder and why it might be difficult to implement. Can you state why the BDFE might be undesirable for partitioning?

4.29 *TEQ Design for the* $1/(1+0.5D)$ *Channel*

For the $1/(1+.5D)$ channel, with white input $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$, AWGN with $\frac{\mathcal{N}_0}{2} = .04$, and $T' = 1$. Force $B(D)$ to have squared norm 4/3 with a 3-tap TEQ.

a. (4 pts) Find the best $B(D)$ with MMSE design for $\nu = 1$ and $\nu = 2$.

b. (2 pts) Find the corresponding MMSEs in part a.

c. (2 pts) Find the corresponding TEQ's ($W(D)$) for part a.

d. (2 pts) Compute the unbiased channel for each $W(D)$ and corresponding distortion energy per sample.

e. (4 pts) Using $\Gamma = 3$ dB, find SNR_{dmt} for this equalized channel with $N = 8$. Repeat for large N to find the limiting value of the SNR_{dmt} for the $\nu = 1$ case. Find also then the largest possible margin for $\bar{b} = 1$.