Multi-Channel Modulation

Chapter 4

Multi-channel Modulation

Multi-channel modulation methods are generally the best for data transmission channels with moderate or severe intersymbol interference because they readily allow spectral shaping and can allow a simple maximum-likelihood receiver implementation. The time-domain sum of many QAM waveforms also tends to be Gaussian, via central-limit-theorem like arguments, which Chapter 2 showed to be the optimum input distribution for the (possibly filtered) AWGN channel. The combination of good codes used across all the carriers has consequently become the modulation format of choice for a great many modern communication systems. Additionally in practice, a particular transmission design may have to exhibit acceptable performance over a range of specific transmission channels, and the multicarrier methods often can be correspondingly adapted in real time with realizable complexity. For instance, there are over a billion telephone lines worldwide that exhibit significant variation in channel response and noise power spectral density. Also, the roughly 200 million cable-television coaxial links can also exhibit significant variation from customer to customer. There are literally billions of wireless-channel situations as well that can occur (and vary rapidly with time). Large performance improvement can be attained in these applications when the modulator output signals can be designed as a function of measured channel characteristics. Multi-channel modulation methods allow such modulator adaptation to channel.

Figure 4.1 illustrates two possible channels, and both use Section 2.5's multitone transmission (MT). The basis functions are essentially a set of non-overlapping narrow QAM signals (i.e., "tones"). Figure 4.1(a)'s channel response removes low-frequencies and also attenuates high-frequencies. Situation (a)'s noise is AWGN only. The amount of information carried on each QAM channel roughly follows the channel transfer function. MT designs thus avoid need for Chapter 3's equalization on any "tone" by making each sufficiently narrow that same band's associated channel response appears "flat," so it satisfies Chapter 3's criterion. Figure 4.1(b)'s same MT transmission sees the same DC notch and attenuation with increasing frequency, but also a notch caused by a delayed path with length increment just such that a 180-degree phase shift occurs in band (green portion). Figure 4.1(b) also has radio-frequency noise (red noise). Finally, Figure 4.1(b) has crosstalk noise (colorblue blue noise) from a system using frequencies just above the nominal MT use, but still with some overlap. The transmit basis function are the same, but clearly carry a very different energy allocation and consequently information in these two illustrations. The MT approach avoids Chapter 3's equalization (which would have been very complex in Example (b)). This chapter develops a theory to analyze and ultimately optimize either of Figure 4.1's example channels, or on any stationary linear channel with additive Gaussian noise for any gap Γ.

MT's origins date to the middle of the 20th century. Shannon's 1948 and 1949 information-theory papers [1] construct capacity bounds for transmission on a linearly filtered AWGN channel, effectively using MT. References [4], [2], and [11] and the many references contained therein describe a number of increasingly successful MT-use attempts cover a 30-year period. These early implementations had diminished performance, largely because they depended on analog-signal-processing. The advent of digital-signal-processing implementations ultimately achieved Shannon's full MT-performance promise, but it took time. While the various early wireline multichannel systems did often work better than other

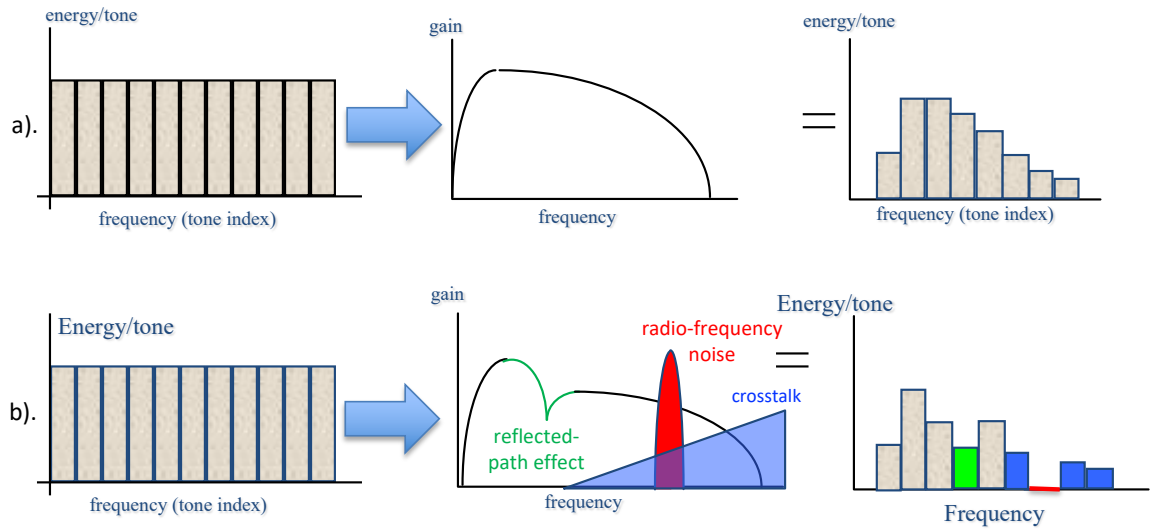


Figure 4.1: Basic concept of adaptive multi-channel ("multitone") transmission.

single-channel modems, a variety of practical problems had earned "multitone" a reputation slandered with skepticism by the early 1990's.

Wireline MT Uses: The first truly successful digital-signal-processing-based effort to popularize MT, with the name "Discrete Multitone" (DMT) emanated from Stanford University[1]. 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. The remaining world's transmission experts anticipated the final downfall of MT. In dramatic reversal of fortune, with measured 10-30 dB improvements (and data rates 4-5 times faster) over the best of the equalized systems on actual transmission lines, the gold-medalist DMT system resurrected MT at 4:10 pm on March 10, 1993[2], when a US (and now world-wide) standard was set for transmitting data over nearly a billion phone lines using multichannel modulation – this is what came to be called "DSL" (digital subscriber line)[3]. Subsequently, various derivative efforts lead to international standards for DMT in Cable (called "DOCSIS 3.1" and 4.0), digital audio broadcast, digital video broadcast, Wi-Fi, and cellular generations 4G and 5G. Indeed the same Olympic modem with minor modifications implemented the world's first "cablemodem" at the December 1993 Anaheim, CA cable show. This chapter provides a general theory and design methodology into which all these

[1]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. While earlier unsuccessful systems had used the name "Orthogonal Frequency Division Multiplexing," this name was avoided to emphasize the highly digital-signal-processing nature of DMT and also because it first used adaptive transmitters based on a feedback control channel.

[2]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 occurring on March 10, 1876, also at 4:10 pm.

[3]It should be noted that a second Bellcore Olympics was held in 2003 as the remnants of the old 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 transmission systems from higher speed DSLs to Cablemodems, to Wi-Fi, and 4G and 5G cellular all now use forms of DMT that are very close to the Olympic Gold designs VDSL - some go under the name of Coded OFDM, which has subtle differences that this Chapter describes.

systems well fit.

Wireless MT Uses: Because the wireline uses succeeded, wireless MT use came next a few months later in 1993. A wireless version of the Olympic modem then participated and again outperformed all other wireless entries (many equalized QAM systems) in the Electronics Industries Assocation (EIA) digital-audio-broadcast (DAB) competition to replace older analog FM radio systems with digital audio[4]. These earlier wireless-DMT transmission systems had no feedback path for an adaptive transmitter, and broadcast a common signal to all users. Thus the transmitter could not adapt. The non-adaptive version instead used highly redundant codes to recover information lost tones caused by Chapter 1's multipath fading. The achieved performance was close to fundamental capacity for the wireless channel, the conditions for which appear in Section 4.4; also the FM-radio band carrier frequencies (and TV frequencies in basically same range) have relatively low Doppler carrier shift and so relative time-variation for the 100's of kHz to few MHz was acceptably low . This was the first real success of **Coded OFDM** and was again standardized internationally for broadcast audio and video (digital TV). As with Chapter 3's CDEF result, it is possible for a constellation size to be constant on the right set of (water-filling) dimensions with good ($\Gamma \to 0$ dB) codes with canonical or optimum performance. Section 4.4 explains this in the context of an equivalent (first noted in Section 2.5) of the MAP, ML, and MMSE estimator for optimum transmission on any filtered AWGN channel.

Wi-Fi Additionally, A wireless local-area-network (WLAN) development called "Hyperlan" subsequently followed the C-OFDM digital broadcast success to specify slightly adaptive MT at wider bandwidths (so time variation appears slower relative to the higher sampling rate needed for wider-band transmission) in small geographic areas (so time variation is limited to people walking basically, not speed vehicles) of 100 meters or less and carrier frequencies of roughly 2.4 GHz and 5 GHz in unlicensed spectrum bands. Wi-Fi's tones are consequently wider by a factor of roughly 10 relative to the earlier wireless systems, but avoid significant distortion through the cyclic-extension methods of Section 4.6 and 4.7. Thus, Wi-Fi uses some feedback and adaptive transmitter, but not as much as wireline (which has much less time variation). Section 4.4 addresses the level of adaptivity, and really the minimum level of feedback necessary to attain nearly optimal performance, possible and the use of **modulation coding schemes (MCS)** in place of the full Shannon-like MT transmitter adaptation. Again, there was good success and this C-OFDM system's 54 Mbps transmission came to be standardized by the IEEE as IEEE 802.11(a) at carrier frequencies between 5 GHz and 6 GHz, and then at a lower 2.4 GHz carrier frequency as the popular IEEE 802.11(g). This Wi-Fi C-OFDM today transmit data rates as high as 10 Gbps, known as 802.11n, 11ac, 11ad, 11.ay (now known also as Wi-Fi 4, 5, and 6 and also the emerging 7, or 11be).

Cellular: Following Wi-Fi's success and exploiting also wider bandwidths but with tone bandwidths closer to the original designs, 4G/LTE and 5G[5] use the same C-OFDM methods with more sophisticated digital tracking systems that exploit various pilot and embedded training signals to update rapidly both the transmitter and the receiver, again attempting to use minimal feedback (indeed less than Wi-Fi uses) through various codebook compression schemes that augment Wi-Fi's MCS methods. These appear in Sections 4.6 and 4.7.

Powerline Communication: Power-line standards vary in transmission format, but many use DMT (a few coded-OFDM) and also time-indexed frequencies over the 17ms cycle of 60 Hz current often on the same transmission channel (which causes the noise to vary periodically over that same cycle). The ITU's "G.hn" standards use DMT in a format similar to "G.fast" high-speed DSL standards, and can carry information on twisted pairs, coax, and/or power lines and combinations thereof. These

[4]This system was again supplied by Stanford spin-out Amati, and special thanks are due in this case again to Mark Mallory, John Bingham and also then-student, now Dr. Sara Kate Wilson, who all helped develop the wireless version.

[5]Fourth and Fifth Generation (4G & 5G) wireless transmission standards are written by the 3rd-Generation Partnership Project (3GPP) that is recognized by the International Telecommunications Union (ITU) standards organization to create global wireless licensed-spectrum transmission standards. LTE is formally "Long-Term Evolution" that has multiple revisions, and progressed from 3G to 4G to 5G and likely beyond in the future.

standards also allow, like the DOCSIS cable versions, for several drop points to add/extract information upstream/downstream on a single transmission medium.

MIMO and Vector Coding: Space-Time matrix or MIMO channels decompose through Chapter 2's vector coding, which Section 4.6 more completely develops, into parallel sets of channels, or multi-channel, also as illustrated throughout this chapter. Some of the earliest recognition of the multi-channel space-time concept dates to purely theoretical work in the 1970's by Brandenburg and Wyner of Bell Labs [5] on multidimensional wireline systems, which was coincident with independent work on arrays of antennas in the general military complex for applications like radar and sensing. This author [7] also investigated codes for such MIMO channels for multiple head/antenna arrays in magnetic recording, and this was later more fully studied by Dr. Paul Voois in his 1994 Stanford dissertation, projecting (parallel) data rate/density increases of roughly 40x the conventional SISO systems (noting specifically the earlier Brandenburg Wyner work as motivational). An early effort to use this same basic concept was patented by Paulraj and Kailath of Stanford in 1991 [6], essentially popularizing the wireless transmission version of MIMO. Paulraj noted that the likelihood of independent spatial transmission channels with multiple coordinated transmit and receive antennas needed only spacing of a fraction of a wavelength between the antennas, meaning space could be used to increase significantly transmission speeds by creating independent sub-channels on the same frequency but isolated electronically in space (similar to the multiple head/antennas use of space). G. Foschini of Bell Laboratories in 1996 [8] also began to observe MIMO wireless data-rate increase, actually citing the same 40x as Cioffi/Voois earlier, with several parallel antennas at both a transmitter and a receiver, extending the Paulraj work, in what was called "BLAST" (Bell Labs Array of Space Time). In 1994, Cioffi and Forney first published their work on application of a "Generalized Decision Feedback" theory to any set of parallel channels, introducing concepts like water-filling in space-time [10].

MU-MIMO: The GDFE allows extension to solve Chapter 2's multi-user problems optimally in space and time (and frequency) as shown in Chapter 5. Cioffi and Ginis followed the GDFE work patented and published on the so-called "Vectored" DSLs that extended the GDFE and first identified a diagonal dominance concept that now is the basic behind massive MIMO systems (where very large numbers of spatial dimensions create the diagonal dominance in wireless and wireline). This chapter builds a basic understanding that then is helpful to address more generally multidimensional problems in Chapter 5.

The sections in order: 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, as a time-frequency channel. Section 4.2 investigates a channel with flat frequency response, but with respective gains 1 and 0.9 on a main line of sight path and a crosstalk path for a space-time version of this same example. Section 4.2 also investigates the generic parallel-channel situation and includes a review of the "SNR gap" of earlier chapters as well as an elaboration of Shannon's water-filling concept in a design (rather than capacity theory) context. The water-fill concept appears mathematically and pictorially. Discrete loading algorithms that optimize the parallel channels' transmit bit and energy distribution appear in Section 4.3, while those with the same constellation size appear in the following Section 4.4. Section 4.4 makes particular use of Section 2.3's infinite block-length AEP/LLN observation that MAP, MMSE, and ML detectors are all the same on the AWGN, which allows significant reduction in the feedback channel necessary in all multi-dimensional methods. Section 4.5 details channel partitioning, the construction of parallel independent channels from a channel with intersymbol interference. With isolated transmit symbols, achieved by waiting for trailing intersymbol interference to decay to zero, Section 4.6 describes the optimum "vector coding," and investigates also its space-time realization in MIMO systems. Section 4.7 introduces slightly suboptimum DMT, OFDM, and related partitioning methods. Subsequent sections address various implementation theories for the practical design: Equalization methods to control delay in MT methods with necessarily finite block length appear in Section 4.8, while Section 4.9 describes windowing methods for out-of-band energy reduction. Section 4.10 analyzes and develops methods to reduce MT methods nominally high peak-to-average power ratio.

4.1 Basic Multi-Channel Transmission

This section first investigates the area of single-input-and-single-output (SISO) transmission channels that are sub-divided mathematically into a number of frequency-indexed tones in Subsection 4.1.1. The section progresses then by extending the basic concepts to MIMO systems where the channel is physically and mathematically divided into a number of space-indexed dimensions in Subsection 4.1.2.

4.1.1 Time-Frequency "Multi-tone" Methods

Figure 4.2's **MultiTone (MT)** modulation uses two or more coordinated passband (QAM or QAM-like) signals or **tones** to carry a single bit stream over a communication channel. These tones are complex dimensions, $\widetilde{N} = 2$. This subsection holds $L_x = L_y = 1$ and the tone-dimensional index is $n = 0, ..., \overline{N}$ where the special case of the zero-frequency $n = 0$ and the Nyquist frequency $n = \overline{N}$ are single real-dimensions; The total is thus as N real dimensions, and as always $N = \widetilde{N} \cdot \overline{N}$. Chapter 2 also used $\overline{N}$ to measure successive subsymbol uses of a channel, but that broader application of N and $\overline{N}$, effectively twice - once in frequency and once over all symbols, does not continue here. However, it will be implied in any context where the analysis assumes good codes with $\Gamma \to 0$ dB, and thus the Chapter-2-style implied asymptotics reapply here in Chapter 4 tacitly. The receiver independently demodulates each of these passband tones and then re-multiplexes them into the original bit stream. MT's heuristic motivation is that if each tone's bandwidth is sufficiently narrow, then no ISI occurs within that dimension. The individual passband dimensions/tones may carry data equally or unequally. Usually, the passband signal(s) with largest channel output signal-to-noise ratio carry a larger fraction of the digital information, consistent with Section 1.6's "gap approximation" that[6] $\bar{b}_n = \frac{1}{2}\log_2\left(1 + \frac{\text{SNR}_n}{\Gamma}\right)$.

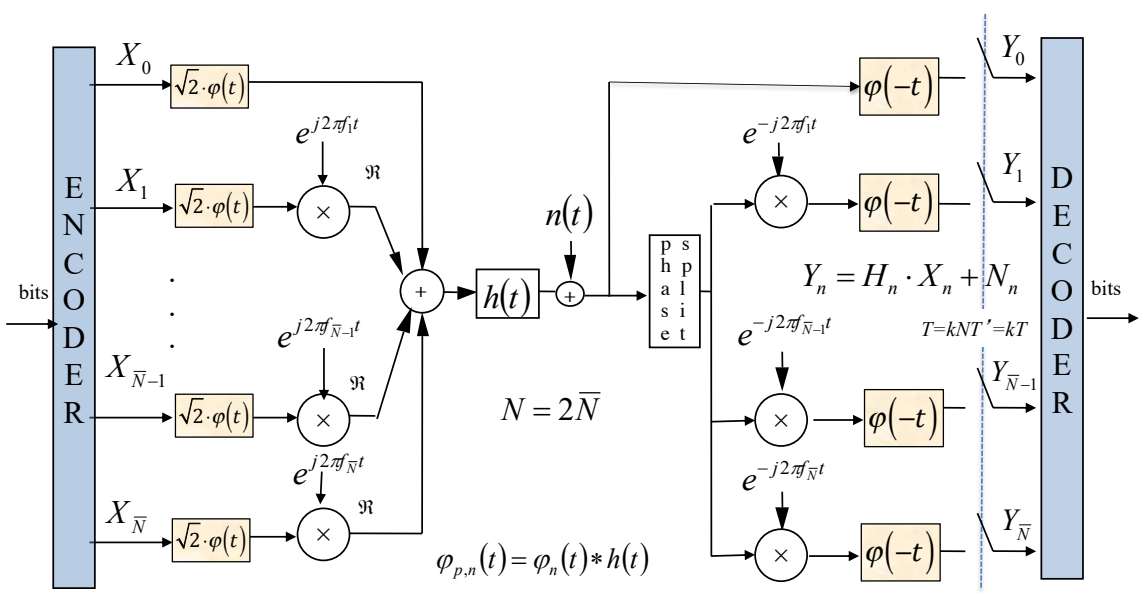


Figure 4.2: Basic MultiTone (MT) Modulation.

Figure 4.2 shows a simple MT system. $\overline{N}$ QAM-like modulators, along with possibly one DC/baseband

[6] 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 2). When $\Gamma = 1$ (0 dB), the data rate is highest (and this is the lowest possible gap). 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.

PAM modulator, transmit $\overline{N}+1$ symbol elements X_n, $n=0,1,...,\overline{N}$.[7] ($\overline{N} \triangleq N/2$, and $N \in 2\mathbb{Z}^+$ is assumed to be even in this chapter.) MT uses capital letters X_n instead of Chapter 2's $\tilde{\boldsymbol{x}}$ because a code may also be used on each (set of) frequency-indexed successive transmissions. This system is essentially the same as Chapter 1's general modulator. X_0 and $X_{\overline{N}}$ are real one-dimensional **elements** while X_n, $n=1,...,\overline{N}-1$ can be two-dimensional ("complex") elements. Each symbol represents one of 2^{b_n} messages that can be transmitted on dimension n. The carrier frequencies for the corresponding dimensions are $f_n = n/T$, where T is again the symbol period. The baseband-equivalent basis functions are $\varphi_n(t) = \sqrt{2} \cdot e^{\jmath 2\pi \cdot f_n \cdot t} \cdot \varphi(t)$,[8] where $\varphi(t) = \frac{1}{\sqrt{T}} \cdot \text{sinc}\left(\frac{t}{T}\right)$. The entire transmitted is (up to) $\overline{N}+1$ dimensions as Figure 4.3 indicates with frequency-indexed bands.

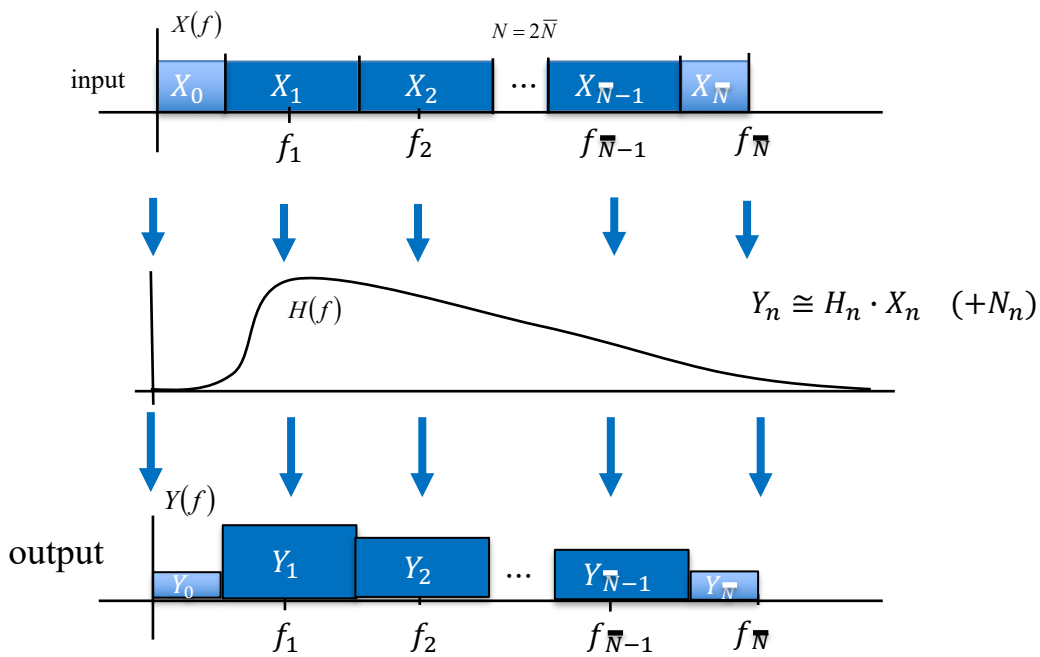


Figure 4.3: Illustration of frequency bands for multitone transmission system (See also Section 2.5).

The MT signal passes through an ISI/AWGN channel to Figure 4.2's demodulators. The receiver separately demodulates each dimension by first quadrature decoupling with a phase splitter and then baseband demodulating with a matched-filter/sampler combination. With this particular ideal-transmitter basis-function choice, the channel output basis functions $\varphi_{p,n}(t)$ also are an orthogonal set. There is thus no interference between the dimensions. Each dimension may have ISI, but as $N = 2 \cdot \overline{N} \to \infty$, this ISI vanishes. Thus, (sub)symbol-by-(sub)symbol detection independently applies to each tonal dimension and implements an overall ML detector. Chapter 3's equalizer/filtering is not necessary (large N) to implement the maximum-likelihood detector. Thus, MT's maximum-likelihood detection is the set of independent simpler ML/dimension detectors. Chapter 3's equalization is consequently unnecessary if each tone's bandwidth is sufficiently narrow to make its ISI negligible. Thus Chapter 3's high-speed modulator, complex equalizer, and suboptimum performance yields to MT's simple and optimal $\overline{N}+1$ low-speed modulators and demodulators in parallel, with the optimality guaranteed as $\overline{N} \to \infty$. Choice of $\overline{N}$ may introduce some practical concerns that Sections 4.5 and 4.6 address.

MT typically uses a $\overline{N}$ value that ensures that the ISI channel's pulse response appears almost

[7] 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.

[8] The passband basis functions are actually the real and imaginary parts of $\sqrt{2/T} \cdot \text{sinc}(\frac{t}{T}) \cdot e^{\jmath 2\pi f_n t}$ with receiver scaling tacitly referred to the transmitter, consistent with in Chapter 1.

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. Earlier channel's (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

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

Equation (4.1)'s approximation becomes increasingly exact as $\overline{N} \to \infty$. Figure 4.3 illustrates the scaling of H_n at each channel's dimension. Each such dimension scales the input X_n by the gain H_n. Each dimension has an AWGN with power-spectral density σ_n^2. Thus, each MT dimension has $\text{SNR}_n = \frac{\bar{\mathcal{E}}_n \cdot |H_n|^2}{\sigma_n^2}$.

Aggregate Bit Rate: Each MT dimension carries b_n bits per element (complex 2 dimensions or real 1 dimension) or $\bar{b}_n$ bits per real dimension (note both $n = 0$ and $n = \overline{N}$ are by definition one-dimensional "subchannels" with subchannel $\overline{N}$ viewed as single-side-band modulation)[9]. The MT system carriers a total number of bits

$$b = \sum_{n=0}^{\overline{N}} b_n \text{ bits/symbol}^{10}, \tag{4.2}$$

and the corresponding data rate is then (with T as the symbol period)

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

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

MT's ML Detection: $\overline{N} + 1$ simple one-dimensional detectors implement overall ML detection with sufficiently large N for an uncoded system using for instance simple PAM and/or QAM. This detector set need not search all combinations of $M = 2^b$ possible transmit symbols; instead each detector operates independently on a simple AWGN. More generally with a code of gap Γ (and again realizing this means a larger outer "N" applies to such codes with each MT symbol becoming a subsymbol or really set of subsymbols for that larger outer code) has ML detector as an independent set of detectors that each apply to one tone's successive dimensions for the outer applied code on that dimension. This maximum-likelihood detector is less complex because of the MT basis-function choice that well matches transmission on the filtered AWGN. Broader bandwidth basis functions, as might be found in a wider bandwidth single-carrier (for instance wide-band QAM) system, often poorly match transmission on the filtered AWGN channel, thus inevitably requiring Chapter 3's complicated equalizers and suboptimum detectors to try to correct the poor transmit basis-function choice. Chapter 3's Sections 11 and 12 address transmit basis-function optimization that leads to a minimum set of QAM signals, each with some equalization. This can be an intermediate option to MT (although it often remains more complex as Chapter 5 shows). Section 2.5 showed MT's combination with a code of gap Γ on each AWGN dimension comes as close to its theoretical capacity as that same code would come to capacity on an AWGN. That is, for sufficiently large N, MT basis functions are indeed optimum for transmission on an ISI channel. This chapter focuses on specifics of practical implementable (with finite N) approximations to Subsection 2.5's theoretical capacity-achieving MT bound.

> **EXAMPLE 4.1.1** *[$1 + .9 \cdot D^{-1}$]* This text repeatedly uses the example of a channel with (sampled) pulse response $1 + .9 \cdot D^{-1}$ to compare performance of various receiver-design structures. MT on this example, as well as on any linear filtered AWGN channel, achieves the highest performance. This continuous MT channel has $H(f) = 1 + .9 \cdot e^{\jmath 2\pi f}$ $(T = N)$. This channel increasingly attenuates higher frequencies, so prudent design might silence dimension

[9] Typically, subchannel 0 and subchannel $\overline{N}$ are not used in actual real baseband systems, although 0 is used in complex baseband systems.

$n = 4$ ($X_4 = 0$). $N = 8$ is not sufficiently large for the approximation in (4.1) to hold, but larger N would make this example intractable. So, this example uses $N = 8$ for easy concept illustration.

Table 4.1 summarizes the dimesional gains and SNR's:

$1 + .9D^{-1}$ 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	$1 + .9e^{\jmath\pi/4} = 1.76\angle 21.3^o$	19.5 (12.9 dB)	2×1.5	9.2
2	16/7	$1 + .9e^{\jmath\pi/2} = 1.35\angle 42.0^o$	11.4 (10.6 dB)	2×1.2	9.1
3	16/7	$1 + .9e^{\jmath 3\pi/4} = .733\angle 60.25^o$	3.4 (5.3 dB)	$2 \times .5$	10

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

The phase choice is irrelevant to achievable performance, and the example would be the same for $1 + .9 \cdot D^{-1}$ as for $1 + .9 \cdot D$. A convenient expression sometimes used with PAM/QAM constellations is to rewrite the Q-function argument

$$\arg(Q) = \sqrt{\frac{3 \cdot SNR}{2^{2\bar{b}} - 1}} \tag{4.4}$$

in terms of $\bar{b}$

$$\bar{b} = \frac{1}{2}\log_2\left(1 + \frac{SNR}{\frac{\arg^2(Q)}{3}}\right) \quad , \tag{4.5}$$

which then implies by Chapter 1's familiar gap formula that the gap can be also written as

$$\Gamma = 20 \cdot \log_{10}\left[\frac{\arg(Q)}{\sqrt{3}}\right] \text{ dB} \quad . \tag{4.6}$$

This relates Table 4.1's last column to the gap, which Section 4.2 reviews. Table 4.1's SNRs assume that $\frac{\mathcal{N}_0}{2} = .181$ – that is, Chapter 3's SNR_{MFB}= 10 dB; and further assume that $b = 8$ and thus consistently, $\bar{b} = 1$. Of course, SNR_{MFB} is a function of the modulated signals and in this example has meaning only in performance comparisons against this example's previous invocations. This example design's energy per dimension places non-zero equal energy per dimension on the 7 energized dimensions. With $T = N = 8$, this energy choice of 8/7 per used real dimension consistently retains the transmit power of 1 (that is one unit of energy per dimension and symbol period 1 as in previous investigations/examples) because the transmit power is $\sum_n \mathcal{E}_n/T = 8/8 = 1$. For this simple example, 4 independent detectors operate in parallel and exceed the performance of Section 3.6's MMSE-DFE, coming very close to the MFB for binary PAM transmission. The data rate remains constant at $\bar{b} = 1$ (or $b = 8$), and the power also is constant at $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$, so comparison against the Section 3.6's DFE is fair. Section 3.11's transmit-optimized MMSE-DFE, as later sections show, can effectively match the best MT performance. 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 each dimension's (or tone's) constellation.

For this example, $N = 8$ is not sufficiently high for all approximations to hold and a more detailed analysis of each dimension's exact SNR might lead to slightly less optimistic performance projection. On the other hand, this example did not yet optimize the energy per dimension nor the bit distribution. For a large N, the best performance will reduce 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 (Section 3.9), 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.

Example 4.1.1 illustrates MT's concept has merit. Noteworthy also is the parallelism inherent in MT's receiver and transmitter, allowing a very high-speed implementation on a number of parallel processing devices. The $1 + .9 \cdot D^{-1}$ channel has mild ISI (but is easy to work with mathematically without computers) – Example 4.1.1's gain of 1.7 dB can be much larger on channels with severe ISI.

4.1.2 Space-Time decompositions into parallel channels

The space-time (MIMO) generalization first retrieves the notation $\widetilde{N} = 2L_x$ that views a matrix AWGN with no ISI and only spatial crosstalk. Then a code, as in Chapter 2, with $N = \overline{N} \cdot \widetilde{N}$ might be in use with $\Gamma \to 0$ dB if the code is capacity achieving. Figure 4.4 illustrates a simple 2×2 ($L_x = 2$) wireless MIMO channel with two transmit antennas and two receive antennas. There may energy transfer between the received dimensions, or "spatial" interference. A matrix H can characterize that spatial interference. Nominal transmission design would treat the spatial interference as "noise" that in addition to any channel noise would act to reduce performance between transmission from transmitter 1 to receiver 1, or from transmitter 2 to receiver 2. The spatial interference need not be detrimental though, and can effectively increase data rate as the following Example 4.1.2 illustrates:

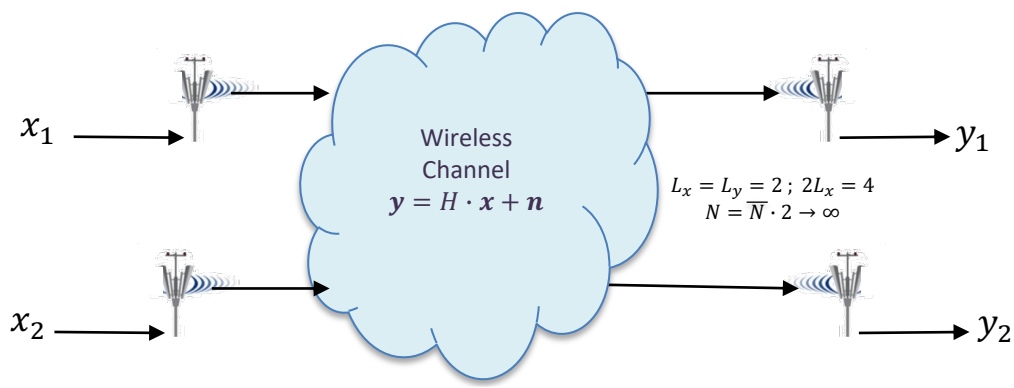


Figure 4.4: Basic space-time wireless matrix AWGN channel 2×2.

EXAMPLE 4.1.2 *[MIMO Spatial Matrix AWGN Channel]* Referring to Figure 4.4, the matrix H has spatial interference (non-zero off-diagonal elements) so that

$$H = \begin{bmatrix} 1 & .9 \\ -.8 & 1 \end{bmatrix} \quad . \tag{4.7}$$

While H is real, the inputs are complex dimensions, $x_1 \in \mathbb{C}$ and $x_2 \in \mathbb{C}$. The output $\boldsymbol{y}$ then would be

$$\boldsymbol{y} = H \cdot \boldsymbol{x} + \boldsymbol{n} \quad , \tag{4.8}$$

an example of a matrix AWGN. This example lets the noise be spatially independent (white), and each complex dimension's noise has variance 0.1 and zero mean. If $x_2 = 0$, then $y_1 = x_1 + n_1$ and $\mathcal{E}_1 = 1$. Then, $\text{SNR}_1 = 10$ or 10 dB. With a gap of $8.8dB$, then $\bar{b} = 0.4$ is attainable with $P_e = 10^{-6}$. However, if $x_2 \neq 0$ and $\mathcal{E}_2 = 1$, then the SNR is $\text{SNR}_1 = 1/(.91) = 1.098$. The attainable bit rate then drops to $\bar{b} = 0.2$.

A similar calculation could occur for the 2-to-2 path with a similar low data rate consequent to crosstalk. A designer might thus abandon the 2nd antenna as useless hardware that causes

a problem. If the second dimension's energy reallocates to the first dimension, increasing that dimension's energy by 3 dB (doubling it), then SNR_1 could increase to 13dB, and a bits/dimension of roughly 0.9 bit/dimension. A more clever designer would try to use a linear-combination at the receiver (like an equalizer that inverts the channel, but enhances noise) and improves the data rate. The result would be about 0.8 bits/dimension.

However, suppose the transmitter uses a 2×2 matrix M that is determined by singular value decomposition (SVD) of $H = F\Lambda M^*$, as per Subsection 2.3.5. F and M are unitary matrices (see more in Section 4.5 on SVD). The matlab commands below show how to calculate these 3 matrices.

```
>> [F, Lambda, Mstar]=svd(H)
F = 0.9076    0.4197
    0.4197   -0.9076
Lambda = 1.3624     0
          0     1.2624
Mstar = 0.4197    0.9076
        0.9076   -0.4197
>> F*Lambda*Mstar = 1.0000    0.9000
                   -0.8000    1.0000
```

The use of M as a transmit matrix does not increase the transmit energy total (both dimensions) of 2 units since $M^*M = I$. The receiver could also use F^* as a receive matrix (which does not change the white noise since $FF^* = I$). Then, there remains two complex dimensions characterized by

$$y_1 = 1.3624 \cdot x_1 + n_1 \tag{4.9}$$

$$y_2 = 1.2624 \cdot x_2 + n_2 \tag{4.10}$$

The corresponding SNR's are $\text{SNR}_1 = 18.5$ and $\text{SNR}_2 = 15.9$, which correspond to bit rates of $\tilde{b}_1 = 1.8$ and $\tilde{b}_2 = 1.6$ for a total of $\tilde{b} = 3.4$ bits/symbol. This is more than 2x faster than any of the approaches mentioned above. Generally speaking, the wireless case can be made to look in many cases like the situation where there are 2 independent wires, and thus twice the data rate, which doubling would clearly and practically be an upper bound (best situation would be 2 independent wires effectively). This of course depends on the channel H and how "rich" it is.

Example 4.4 shows that joint transmitter and reciver optimization both transmitter and receiver in the matrix AWGN channel can be profound. In fact, the interested reader could try switching the sign of the term -0.8 to $+0.8$ in H and rework this example to see the data rate reduces significantly with respect to the 3.4 bits/subsymbol (roughly 1/2) because this alternate H is not as "rich." (Hint: the size and spread of the singular values is important.)

4.2 Parallel Channels

Section 2.5's and Section 4.1's MT system inherently exhibits $\bar{N}+1$ independent complex-dimension subchannels[11]. Generally, multichannel systems have N independent single-real-dimensional subchannels (where in the MT system, several real-dimensional pairs have identical gains and can be considered as two-real-dimensional subchannels or as single-complex-dimensional subchannels). This section studies the general communication channel that decomposes into an equivalent set of N real parallel subchannels. Of particular importance is performance analysis and optimization for the entire set of subchannels.

The SNR-gap concept is often important, so Subsection 4.2.1analyzes a single-channel gap and then Subsections 4.2.2 and 4.2.3 use these single-channel results to establish equivalent broader results for parallel channels.

4.2.1 Single-channel gap analysis

Chapter 1's "gap" analysis is reappears here briefly in preparation for the study of parallel channels and loading algorithms.

An AWGN channel with gain H_n, input power spectral density $\mathcal{E}_n$, and noise power spectral density σ_n^2 has thus SNR_n is $\frac{\mathcal{E}_n \cdot |H_n|^2}{\sigma_n^2}$. This channel has maximum data rate or capacity[12] of

$$\bar{\mathcal{C}}_n = \frac{1}{2} \cdot \log_2(1 + \text{SNR}_n) \tag{4.11}$$

bits/dimension. More generally,

$$\bar{b}_n = \frac{1}{2} \cdot \log_2(1 + \frac{\text{SNR}_n}{\Gamma}) \quad . \tag{4.12}$$

As in Section 2.3, any reliable and implementable system must transmit at a data rate at least slightly below capacity. The gap facilitates analysis when $\bar{b} < \bar{\mathcal{C}}$. Most practical signal constellations (e.g., PAM, QAM) have a constant gap for all $\bar{b} \geq .5$. For $\bar{b}_n < .5$, code systems exist that exhibit the same constant gap[13] as for $\bar{b} \geq .5$. For any given coding scheme and a given target symbol-error probability P_e, the SNR gap is

$$\Gamma \triangleq \frac{2^{2\bar{\mathcal{C}}} - 1}{2^{2\bar{b}} - 1} = \frac{\text{SNR}}{2^{2\bar{b}} - 1} \quad . \tag{4.13}$$

To illustrate such a constant gap, Table 4.2 repeats achievable $\bar{b}$ for uncoded QAM schemes using square constellations.

[11] Actually, 2 corresponding to index $n = 0$ and $n = \overline{N}$ are single-real-dimensional, but could be thought of as one two-dimensional subchannel, although likely with different gains on each of those real dimensions. This special case only occurs for purely real baseband channels.

[12] 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 $\bar{b} \leq \bar{c}$.

[13] 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 $\bar{P}_e = 10^{-6}$.

Again for $\bar{b} < 1$, constant gap presumes mild additional coding (see Chapters 9 - 11) in the "uncoded" system when $\bar{b} = .5$, or even lower.[14] 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 2 and 9 - 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 4.5 plots $\bar{b}$ versus SNR for various gaps. Smaller gap indicates better coding. The curve for $\Gamma = 9$ dB approximates Chapter 1's uncoded PAM or QAM transmission at symbol-error probability $P_e \approx 10^{-6}$ (really 8.8 dB). A gap of 0 dB means the theoretical maximum bit rate has been achieved, which requires infinite complexity and delay - see Chapter 2.)

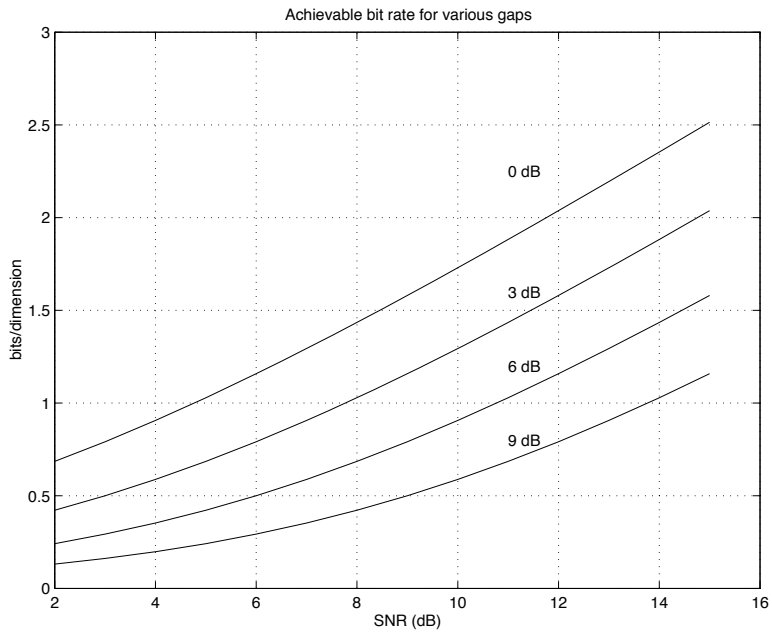


Figure 4.5: Illustration of bit rates versus SNR for various gaps.

The remainder of this chapter and Chapter 5 assume the same constant gap for each and every subchannel's coding.

Margin repeated from Section 1.5: 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.

Definition 4.2.1 [Margin] *The* **margin**, γ_m, *for transmission on an AWGN (sub)channel with a given SNR, for a given number of bits per dimension* $\bar{b}$, *and for*

[14] Thus, Table 4.2 assumes for $\bar{b} = .5$ that instead of simple BPSK transmission of $\left[\pm\sqrt{2}\ \ 0\right]$ for an energy/symbol of 2, $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$, and $d^2_{min} = 8$ that instead the system transmits one of these 4 symbol values for two successive QAM channel inputs $2/\sqrt{3} \cdot [1\ 1\ 1\ 0]$, $2/\sqrt{3} \cdot [-1\ \ -1\ 1\ 0]$, $2/\sqrt{3} \cdot [1\ \ -1\ \ -1\ 0]$, or $2/\sqrt{3} \cdot [-1\ 1\ \ -1\ 0]$ for also an energy per dimension of $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$ and $\bar{b} = 0.5$, but with a minimum distance squared of $d^2_{min} = 32/3$. This improvement is 32/24= 1.3 dB. Using a few more dimensions like this can provide another .4 dB fairly easily, so up to 1.7 dB. So while pure BPSK would have a gap of 10.5 dB at $P_e = 10^{-6}$, this mildly coded system has a gap of 10.5-1.7=8.8 dB, rendering constant the QAM gap of 8.8 dB all the way down to $\bar{b} = 0.5$ as shown. Similar things can be done for coded designs with lower gaps (than 8.8 dB) when $\bar{b} = .5$, and in particular with a little more effort a gap of 8.8 dB can be maintained simply down to 0.25 bits/dimension although not shown here.

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 an error probability at or below the target P_e.

Margin follows through the use of the gap formula as

$$\gamma_m = \frac{2^{2\bar{b}_{max}} - 1}{2^{2\bar{b}} - 1} = \frac{\text{SNR}/\Gamma}{2^{2\bar{b}} - 1} \quad . \tag{4.14}$$

A negative margin in dB means that the SNR must improve by the magnitude of the margin to attain the target P_e. The margin relation is equivalent to

$$\bar{b} = .5 \log_2 \left(1 + \frac{\text{SNR}}{\Gamma \cdot \gamma_m} \right) \quad . \tag{4.15}$$

EXAMPLE 4.2.1 [AWGN with SNR = 20.5 dB] An AWGN has SNR of 20.5 dB. The channel capacity is then

$$\bar{\mathcal{C}} = \frac{1}{2} \cdot \log_2 (1 + \text{SNR}) = 3.5 \ \text{ bits/dim} \quad . \tag{4.16}$$

With a $P_e = 10^{-6}$ and 2B1Q (4 PAM),

$$\bar{b} = \frac{1}{2} \cdot \log_2 \left(1 + \frac{\text{SNR}}{10^{.88}} \right) = 2 \text{ bits/dim} \quad . \tag{4.17}$$

With concatenated trellis and forward error correction (or with "turbo codes" – see Chapters 9 - 11), and a coding gain of 7 dB at $P_e = 10^{-6}$, then the achievable data rate is

$$\bar{b} = \frac{1}{2} \cdot \log_2 \left(1 + \frac{\text{SNR}}{10^{.18}} \right) = 3 \text{ bits/dim} \quad . \tag{4.18}$$

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

$$\gamma_m = \frac{2^{2\cdot 3} - 1}{2^{2\cdot 2.5} - 1} = \frac{63}{31} \approx \ 3 \text{ dB} \quad . \tag{4.19}$$

This means the noise power would have to be increased (or the transmit power reduced) by more than 3 dB before the target error probability 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

$$\gamma_m = \frac{2^{2\cdot 2} - 1}{2^{2\cdot 1} - 1} = \frac{15}{3} \approx \ 7 \text{ dB} \quad . \tag{4.20}$$

4.2.2 A single performance measure for parallel channels - geometric SNR

Multi-channel transmission designs usually 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
Gamma. 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 Chapter 3's SNR_{MFB}.

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

$$\begin{aligned}
\bar{b} &= (1/N) \cdot \sum_{n=1}^{N} \bar{b}_n = (1/N) \sum_{n=1}^{N} \frac{1}{2} \log_2 \left(1 + \frac{\mathrm{SNR}_n}{\Gamma}\right) && (4.21)\\
&= \frac{1}{2N} \log_2 \left(\prod_{n=1}^{N} \left[1 + \frac{\mathrm{SNR}_n}{\Gamma}\right]\right) && (4.22)\\
&\triangleq \frac{1}{2} \log_2 \left(1 + \frac{\mathrm{SNR}_{m,u}}{\Gamma}\right) \quad . && (4.23)
\end{aligned}$$

Definition 4.2.2 **[Multi-channel or geometric SNR]** *The* **multi-channel SNR** *for a set of parallel channels is defined by*

$$SNR_{m,u} \triangleq \left[\left(\prod_{n=1}^{N} \left[1 + \frac{SNR_n}{\Gamma}\right]\right)^{1/N} - 1\right] \cdot \Gamma \quad . \tag{4.24}$$

The subscript u abbreviates "unbiased" and should not be confused with Chapter 2's user index variable.

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 4.6.

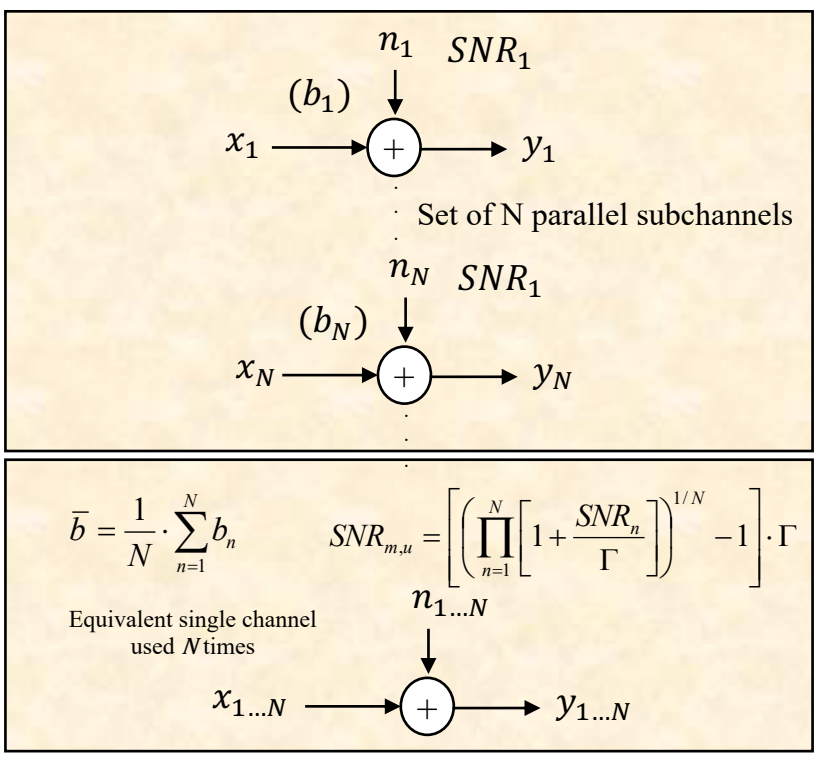


Figure 4.6: Equivalent single channel characterized by Multi-channel SNR.

The multichannel SNR in (4.24) can be directly compared with Chapter 3's detection SNR of an equalized single-channel system. The bit rate is given in (4.23) as if the aggregate set of parallel channels were a single AWGN with $\mathrm{SNR}_{m,u}$, used N times (or $\overline{N}$ times if complex).

When the terms involving +1 can be ignored, the multichannel SNR is approximately the geometric mean of the subchannel's SNRs:

$$\text{SNR}_{m,u} \approx \text{SNR}_{geo} = \left(\prod_n \text{SNR}_n\right)^{1/N} \quad . \tag{4.25}$$

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

$$\text{SNR}_{m,u} = \left[(22.8+1)(19.5+1)^2(11.4+1)^2(3.4+1)^2\right]^{1/8} - 1 = 8.8 \text{ (dB)} \quad . \tag{4.26}$$

This 8.8 dB is an SNR, and just coincidentally happens to be the same as the gap for 10^{-6} error rate. 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.[15] $\text{SNR}_{m,u} = 9.7$ dB for $\Gamma = 8.8$ dB, which is very close to Example 4.1.1's value. This value is artificially high only because the approximation accruing to using too small a value of $\overline{N}$ for illustration purposes - if $\overline{N}$ were large, $\text{SNR}_{m,u}$ would be close to 8.8 dB even with $\Gamma > 1$. Also the gap approximation loses accuracy when $\bar{b} < 1$, so systems with many low-b_n subchannels might avoid gap analysis unless very good codes with $\Gamma \to 0$ dB are in use, in which case the analysis remains accurate.

4.2.3 The Water-Filling Optimization

Data rate, $R = b/T$, maximization for a parallel-subchannel set (at fixed (sub)symbol rate $1/T$) finds the largest achievable bits/dimension

$$b = \frac{1}{2}\sum_{n=1}^{N} \log_2(1 + \frac{\bar{\mathcal{E}}_n \cdot g_n}{\Gamma}) \quad , \tag{4.27}$$

over b_n and $\bar{\mathcal{E}}_n$, where g_n represents the subchannel gain ratio (the SNR when the transmitter applies unit energy to that subchannel). MT has $g_n = |H_n|^2/(\sigma_n^2)$. The gain g_n is a given fixed function of the subchannel, but $\mathcal{E}_n$ can be varied to maximize b, subject to the energy constraint

$$\sum_{n=1}^{N} \bar{\mathcal{E}}_n = N \cdot \bar{\mathcal{E}}_{\boldsymbol{x}} \;\; ; \;\; \bar{\mathcal{E}}_n \geq 0 \; \forall \; n \in \{1, ..., N\} \quad . \tag{4.28}$$

Using Lagrange multipliers, the cost function to maximize (4.27) subject to the constraint in (4.28) becomes

$$\frac{1}{2\ln(2)} \cdot \sum_n \ln\left(1 + \frac{\bar{\mathcal{E}}_n \cdot g_n}{\Gamma}\right) + \lambda \cdot \left(\sum_n \bar{\mathcal{E}}_n - N \cdot \bar{\mathcal{E}}_{\boldsymbol{x}}\right) \quad . \tag{4.29}$$

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

$$\frac{1}{2\ln(2)} \cdot \frac{1}{\frac{\Gamma}{g_n} + \bar{\mathcal{E}}_n} = -\lambda \quad , \tag{4.30}$$

leading to the Water-Filling Theorem

Theorem 4.2.1 [Water-Filling Energy Distribution] *The data rate (4.27) is maximized, subject to (4.28), when*

$$\bar{\mathcal{E}}_n + \frac{\Gamma}{g_n} = \textit{ constant} \quad . \tag{4.31}$$

Proof: *The preceding equations.* **QED.**

[15] Somewhat by coincidence because the continuous optimum bandwidth from Chapter 2 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 accurately.

When $\Gamma = 1$ (0 dB), water-filling attains maximum data rate, or capacity, of the parallel subchannel set. As in Chapter 2, the solution is called the "water-filling" solution because the solution depicts graphically as a curve, or "bowl," of inverted channel signal-to-noise ratios being filled with energy (water) to a constant line. as in Figure 4.7. Figure 4.7 depicts water-filling for a transmission system with 6 subchannels with $g_n = \frac{|H_n|^2}{\sigma_n^2}$. When $\Gamma \neq 1$ (> 0 dB), the form of the water-fill optimization remains the same (as long as Γ remains 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 (possibly) more narrow (fewer used subchannels) optimized transmit band than when capacity ($\Gamma = 1$) is achieved. The number of bits on each subchannel is then

$$b_n = .5 \cdot \log_2 \left(1 + \frac{\bar{\mathcal{E}}_n \cdot g_n}{\Gamma} \right) \quad . \tag{4.32}$$

For MT, the optimum water-filling transmit energies then satisfy

$$\bar{\mathcal{E}}_n + \Gamma \cdot \frac{\sigma_n^2}{|H_n|^2} = \text{ constant } \quad . \tag{4.33}$$

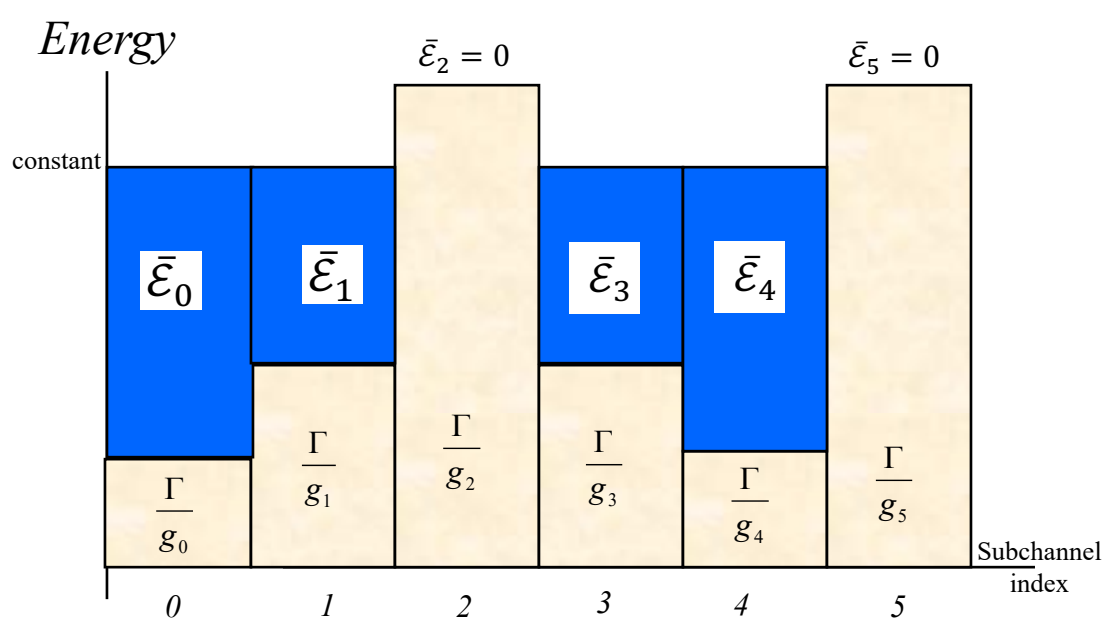


Figure 4.7: Illustration of discrete water-filling for 6 subchannels.

Water-filling first solves Equation (4.33) for the constant K 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 4.7 have positive energies, while 2 carry no energy rather than solve (4.33) with negative energy. Equivalently these zeroed subchannels have a noise power that exceeds the "water" line. The 4 used subchannels have energy that makes the sum of input-referred noise (inverse subchannel gain) and transmit energy constant. Subsection 4.3 provides water-filling algorithms.

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. The subchannel index n index is fairly general in all developments. It is possible to execute water-filling over any parallel set of independent channels that may be indexed in frequency, time, and/or space. Water-filling's optimality remains for the sum data rate over all such independent subchannels.

4.2.4 Margin Maximization (or Power Minimization)

For many transmission systems, variable data rate is not desirable. In this case, the best design instead maximizes 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)

$$\mathcal{E}_x = \sum_{n=1}^{N} \bar{\mathcal{E}}_n \tag{4.34}$$

subject to fixed data rate (and coding gap) as per

$$\sum_{n=1}^{N} \frac{1}{2} \cdot \log_2 \left(1 + \frac{\bar{\mathcal{E}}_n \cdot g_n}{\Gamma} \right) = b \quad . \tag{4.35}$$

The best margin is then

$$\gamma_m = \frac{N \cdot \overline{\mathcal{E}}_{\boldsymbol{x}}}{\sum_{n=1}^{N} \bar{\mathcal{E}}_n} \quad . \tag{4.36}$$

The energy minimizatio's solution after correct differentiation is again water-filling:

$$\bar{\mathcal{E}}_n + \frac{\Gamma}{g_n} = \text{ constant} \quad . \tag{4.37}$$

However, in this case water/energy pours only until the number of bits per symbol (which is computed at each subchannel according to (4.32) and then summed over all subchannels) is equal to the given fixed rate in (4.35). Then the maximum margin follows from (4.36). Each subchannel increases its energy uniformly with respect to the minimized energye sum by a factor that is the maximum margin in (4.36). This energy-minimizing form of the bit/energy-loading problem is known in mathematics as "the dual form" of the rate-maximizing formulation. Sometimes in multi-user channels, see Chapter 5, the solution with minimum energy for a given data rate and given margin is of interest in what is called "iterative water-filling." Energy minimization is equivalent to margin maximization because any other bit distribution would have required more energy to reach the same given data rate, and the P_e is constant (at the level corresponding to Γ) for all subchannels/dimensions.

Wireless, and sometimes wireline cable, uses of multi-carrier usually have external modulators as in Section 1.3.6. In these MT cases, each tone is really complex and has two equal-gain real dimensions and a phase, or more simply the channel adjusts gain and phase. Only the gain is important in water-filling. In these cases, water-filling can be directly applied to the gains and the input energy powers can be determined. The average energy of the "QAM" inputs to each tone is then the amount determined and the bits carried is $\tilde{b}_n = \log_2 (1 + \text{SNR}_n/\Gamma)$ (that is, without the 1/2 in front) with $\tilde{N} = 2$ and $N = 2\overline{N}$.

4.3 Loading Algorithms with Variable Constellations

Loading algorithms compute values for the bits b_n and energy $\mathcal{E}_n$ for each and every subchannel (dimension) in a parallel set with $N < \infty$. One example of a loading algorithm is Section 4.2's optimum water-filling algorithm that solves a set of linear equations with boundary constraints, as this section describes further. The water-filling solution for large N may produce b_n that have fractional parts or that are very small. Such small or fractional b_n can complicate practical encoder and decoder implementation. This section's alternative loading algorithm approximates the water-fill solution, but constrains b_n to integer values.

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. This section studies both.

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 (using N real dimensions here):*

$$\max_{\bar{\mathcal{E}}_n} b = \sum_{n=1}^{N} \frac{1}{2} \cdot \log_2 \left(1 + \frac{\bar{\mathcal{E}}_n \cdot g_n}{\Gamma}\right) \tag{4.38}$$

$$\textit{subject to: } N \cdot \bar{\mathcal{E}}_{\boldsymbol{x}} = \sum_{n=1}^{N} \bar{\mathcal{E}}_n \tag{4.39}$$

Complex channels simply treat $N \rightarrow \overline{N}$ with energy then being also per two-dimensional subsymbol.

Definition 4.3.2 [Margin-Adaptive (MA) - Energy-Minimizing- loading criterion] *A* **margin-adaptive** *loading procedure minimizes (or approximately minimizes) the energy subject to a fixed bits/symbol constraint (again with real dimensions):*

$$\min_{\bar{\mathcal{E}}_n} \bar{\mathcal{E}}_{\boldsymbol{x}} = \sum_{n=1}^{N} \bar{\mathcal{E}}_n \tag{4.40}$$

$$\textit{subject to: } b = \sum_{n=1}^{N} \frac{1}{2} \cdot \log_2 \left(1 + \frac{\bar{\mathcal{E}}_n \cdot g_n}{\Gamma}\right) \tag{4.41}$$

The consequent maximum margin is then

$$\gamma_{max} = \frac{N \cdot \bar{\mathcal{E}}_{\boldsymbol{x}}}{\mathcal{E}} \quad . \tag{4.42}$$

When $\bar{\mathcal{E}}_{\boldsymbol{x}}$ remains at its minimized value (and the gap includes any desired margin and coding gain by definition) in Equations (4.40)'s and (4.41)'s solution for the given data rate b, the loading is called **Fixed Margin (FM)**. *Complex channels simply replace N by $\overline{N}$, and then energy becomes per two-dimensional subsymbol.*

FM loading is useful in transmission designs for which power savings is important. For both the RA and MA problems, the constant (across all indices n) gap and could have been absorbed[16] into the energy variable $\mathcal{E}_n \leftarrow \mathcal{E}_n/\Gamma$. The gap is only important to determine the final minimized energy total or to determine the final maximized data rate, so thus to meet the constraint. It is otherwise not fundamental to the different dimensions' relative energy (or bit) allocations.

[16] This single-user case is easier than Chapter 2's multi-user MAC and BC, where such absorption is not helpful.

4.3.1 Computing Water Filling for RA loading

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

$$\begin{aligned}
\bar{\mathcal{E}}_1 + \Gamma/g_1 &= K && (4.43)\\
\bar{\mathcal{E}}_2 + \Gamma/g_2 &= K && (4.44)\\
\vdots &= \vdots && (4.45)\\
\bar{\mathcal{E}}_N + \Gamma/g_N &= K && (4.46)\\
\bar{\mathcal{E}}_1 + ... + \bar{\mathcal{E}}_N &= N \cdot \bar{\mathcal{E}}_{\boldsymbol{x}} \quad . && (4.47)
\end{aligned}$$

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

$$\begin{bmatrix} 1 & 0 & 0 & 0 & ... & -1 \\ 0 & 1 & 0 & 0 & ... & -1 \\ \vdots & \vdots & \ddots & \ddots & \ddots & -1 \\ 1 & 1 & 1 & ... & 1 & 0 \end{bmatrix} \begin{bmatrix} \bar{\mathcal{E}}_1 \\ \bar{\mathcal{E}}_2 \\ \vdots \\ \bar{\mathcal{E}}_N \\ K \end{bmatrix} = \begin{bmatrix} -\Gamma/g_1 \\ -\Gamma/g_2 \\ \vdots \\ -\Gamma/g_N \\ N \cdot \bar{\mathcal{E}}_{\boldsymbol{x}} \end{bmatrix} \quad , \tag{4.48}$$

which has solution by matrix inversion. Matrix inversions of size $N+1$ down to 1 may have to be performed iteratively until all dimensions have nonnegative energies.

An alternative simpler solution sums the first N equations to obtain:

$$K = \frac{1}{N} \left[N \cdot \bar{\mathcal{E}}_{\boldsymbol{x}} + \Gamma \cdot \sum_{n=1}^{N} \frac{1}{g_n} \right] \quad , \tag{4.49}$$

and

$$\bar{\mathcal{E}}_n = K - \Gamma/g_n \;\; \forall\, n = 1, ..., N \quad . \tag{4.50}$$

If one or more of $\mathcal{E}_n < 0$, then water-filling eliminates most negative and continues again with the new (4.49) and (4.50) until all energies are nonnegative, with $N \rightarrow N-1$ and the corresponding g_n term eliminated on each successive step. Water-filling preferably preorders the equations in terms of channel signal-to-noise ratio with $n = 1$ corresponding to the largest-SNR subchannel, $g_1 = \max_n g_n$, and with $n = N$ corresponding to the smallest-SNR subchannel, $g_N = \min_n g_n$. Equation (4.49) then becomes at the ith step of the iteration $(i = 0, ..., N)$

$$K = \frac{1}{N-i} \cdot \left[\mathcal{E}_{\boldsymbol{x}} + \Gamma \cdot \sum_{n=1}^{N-i} \frac{1}{g_n} \right] \quad , \tag{4.51}$$

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

$$\bar{\mathcal{E}}_n = K - \Gamma/g_n \;\; \forall\, n = 1, ..., N^* = N - i \quad . \tag{4.52}$$

Figure 4.8's flow chart illustrates the water-filling algorithm.

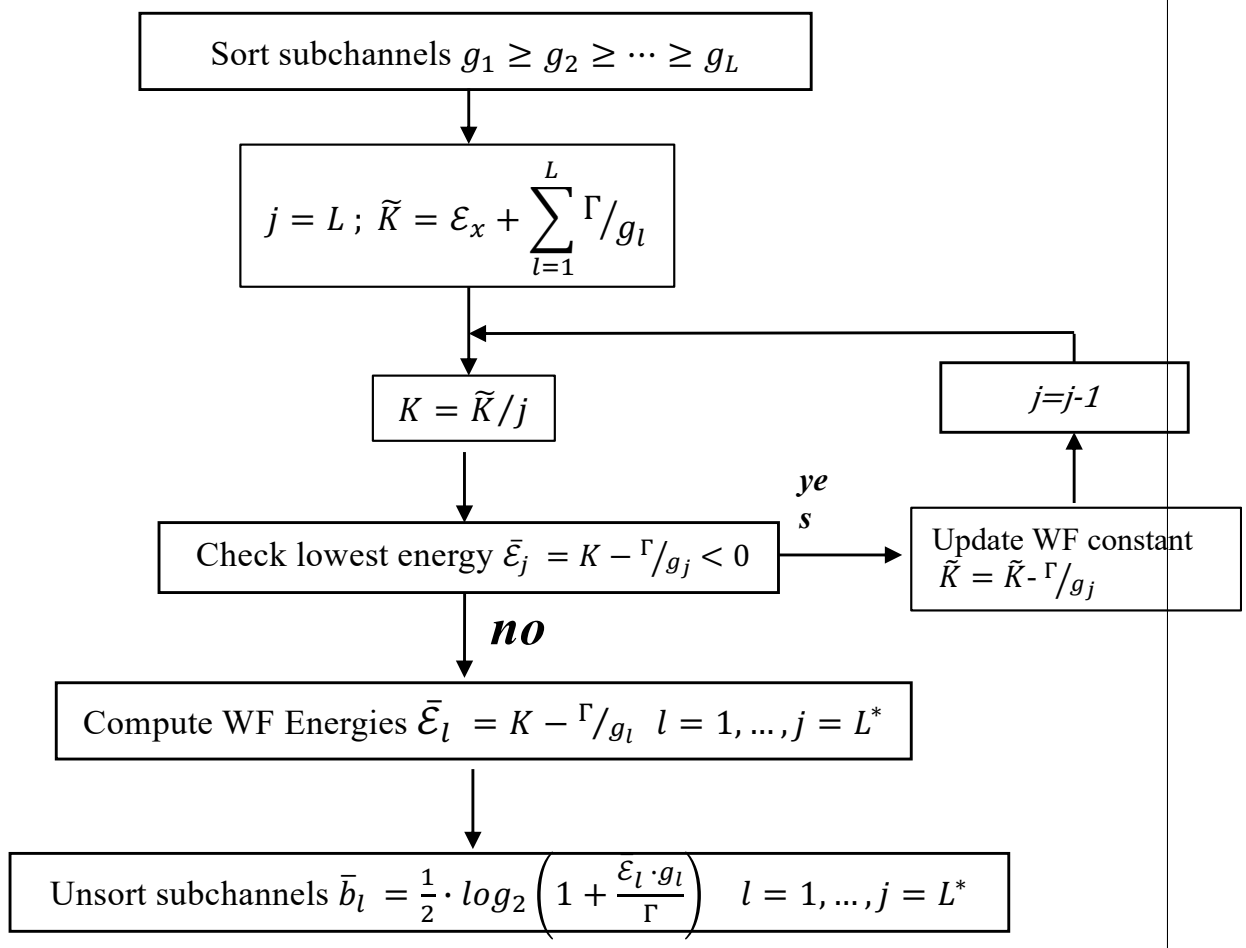


Figure 4.8: Flow chart of Rate-Adaptive Water-Filling Algorithm.

The sum in the expression for K in (4.49) is always over the used subchannels. The following simplified formulas, apparently first observed by J. Aslanis in his dissertation [12], then determine the number of bits and number of bits per dimension as

$$b_n = \frac{1}{2} \cdot \log_2 \left(\frac{K \cdot g_n}{\Gamma} \right) = \frac{1}{2} \cdot \log_2 \left(1 + \frac{\bar{\mathcal{E}}_n \cdot g_n}{\Gamma} \right) \quad . \tag{4.53}$$

and

$$\bar{b} = \frac{1}{2} \cdot \log_2 \left(\frac{K^{N^*/N} \cdot (\prod_{n=1}^{N^*} g_n)^{1/N}}{\Gamma^{N^*/N}} \right) = \frac{1}{N} \cdot \sum_{n=1}^{N^*} b_n \tag{4.54}$$

$$= \frac{1}{2} \cdot \log_2 \left(1 + \frac{\mathrm{SNR}_{m,u}}{\Gamma} \right) \quad , \tag{4.55}$$

respectively. Equation (4.54) illustrates a curious MMSE-biased (see Appendix D.2) SNR (when $\Gamma = 1$, 0 dB) $K \cdot g_n$ for each dimension: That is the biased SNR is the water-level times the channel gain g_n. The SNR for the set of parallel channels is therefore

$$\mathrm{SNR}_{m,u} = \Gamma \left\{ \left(\frac{K}{\Gamma} \right)^{N^*/N} \left(\prod_{n=1}^{N^*} g_n^{1/N} \right) - 1 \right\} \quad . \tag{4.56}$$

The water-filling-level constant K becomes the ratio

$$K = \Gamma \cdot \left(\frac{2^{2b}}{\prod_{n=1}^{N^*} g_n} \right)^{1/N^*} \quad . \tag{4.57}$$

The capacity water-filling level is the geometric average over the used subchannels of a constant constellation size, $M = 2^{2\bar{b}^*}$, to subchannels' geometric-gain ratio. Readers may recognize this as Chapter 2's average asymptotic equipartition over the subchannels. Also the constant gap enters the optimization only in scaling linearly the constant K.

EXAMPLE 4.3.1 **[$1 + .9D^{-1}$ again]** The water-fill solution for $\Gamma = 1$ on the $1 + .9D^{-1}$ channel appears here. First, the subchannels have gains

$$g_0 = \frac{1.9^2}{.181} = 19.94 \tag{4.58}$$

$$g_1 = \frac{1.7558^2}{.181} = 17.03 \tag{4.59}$$

$$g_2 = \frac{1.3454^2}{.181} = 10.00 \tag{4.60}$$

$$g_3 = \frac{.7329^2}{.181} = 2.968 \tag{4.61}$$

$$g_4 = \frac{.1^2}{.181} = .0552 \tag{4.62}$$

Using (4.49) with all subchannels ($N = 8$ or actually 5 subchannels since 3 are complex with identical g's on real and imaginary dimensions), one computes

$$K = \frac{1}{8}\left[8 + 1 \cdot (\frac{1}{19.94} + \frac{2}{17.03} + \frac{2}{10.0} + \frac{2}{2.968} + \frac{1}{.0552})\right] = 3.3947 \quad , \tag{4.63}$$

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

$$K = \frac{1}{7}\left[8 + 1 \cdot (\frac{1}{20} + \frac{2}{17} + \frac{2}{9.8} + \frac{2}{3})\right] = 1.292 \quad , \tag{4.64}$$

The corresponding subchannel energies, $\bar{\mathcal{E}}_n$, on the subchannels are 1.24, 1.23, 1.19, and .96, which are all positive, attaining the best solution with all largest set of all-positive-energy subchannels/dimensions. The corresponding number of bits per each subchannel is $\bar{b}_n = .5 \log_2(K \cdot g_n)$, 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 Section 2.5's known capacity of this channel[17]. Figure 4.9 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 error probability for the selected transmission rate of $\bar{b} = 1$ in Chapter 3's earlier study of this example channel.

The capacity computation's accuracy increases somewhat by further increasing N – the eventual large-N-asymptotic value is approximately 1.55 bits/dimension. Then, $\text{SNR}_{m,u} = 2^{2(1.55)} - 1 = 8.8$ dB. This is the highest detection-point SNR that any receiver can achieve. This is better than Chapter 3's 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 MT and Chapter 3's MMSE-DFE can be very large, especially when the MMSE-DFE uses a single constant transmit band.

[17] An Example in Section 2.5 found this channel to have a capacity of 1.55 bits/dimension.

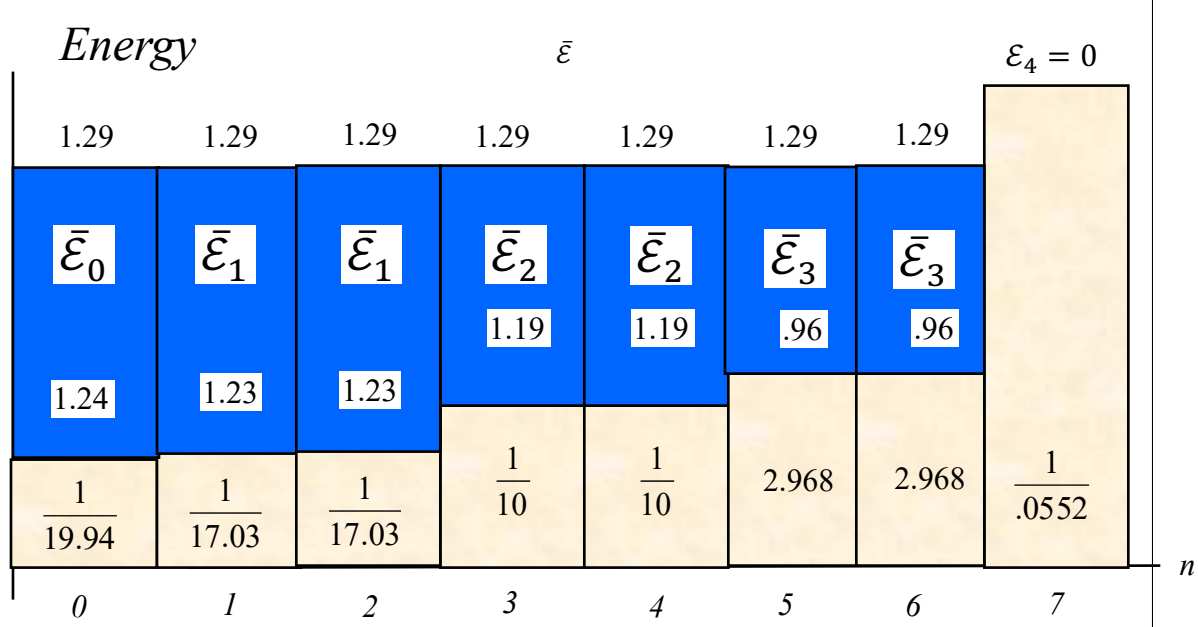


Figure 4.9: Water-filling energy distribution for $1 + .9D^{-1}$ channel with $N = 8$.

4.3.2 Computing Water-Filling for MA loading

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

$$\bar{\mathcal{E}}_n = K_{ma} - \frac{\Gamma}{g_n} \tag{4.65}$$

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

$$\begin{aligned} b &= \frac{1}{2} \cdot \sum_{n=1}^{N^*} \log_2 \left(1 + \frac{\bar{\mathcal{E}}_n \cdot g_n}{\Gamma}\right) && (4.66) \\ &= \frac{1}{2} \cdot \sum_{n=1}^{N^*} \log_2 \left(\frac{K_{ma} \cdot g_n}{\Gamma}\right) && (4.67) \\ &= \frac{1}{2} \cdot \log_2 \left(\prod_{n=1}^{N^*} \frac{K_{ma} \cdot g_n}{\Gamma}\right) && (4.68) \\ 2^{2b} &= \left(\frac{K_{ma}}{\Gamma}\right)^{N^*} \cdot \prod_{n=1}^{N^*} g_n && (4.69) \end{aligned}$$

Thus, the constant for MA loading is (rearranging (4.54) because now b is known)

$$K_{ma} = \Gamma \cdot \left(\frac{2^{2b}}{\prod_{n=1}^{N^*} g_n}\right)^{1/N^*} . \tag{4.70}$$

Also the constant K scales linearly with the gap Γ.

4.3.2.1 MA water-fill algorithm

1. Order g_n largest to smallest, $n = 1,..., N$, and set $i = N$.
2. Compute $K_{ma} = \Gamma \cdot \left(\frac{2^{2b}}{\prod_{n=1}^{i} g_n}\right)^{1/i}$ (this step requires only one multiply for $i \leq N-1$).
3. Test subchannel energy $\bar{\mathcal{E}}_i = K_{ma} - \frac{\Gamma}{g_i} < 0$? If yes, then $i \leftarrow i-1$, go to step 2, otherwise continue
4. Compute solution with $n = 1, ..., N^* = i$

$$\bar{\mathcal{E}}_n = K_{ma} - \frac{\Gamma}{g_n} \tag{4.71}$$

$$b_n = \frac{1}{2} \cdot \log_2\left(\frac{K_{ma} \cdot g_n}{\Gamma}\right) \tag{4.72}$$

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

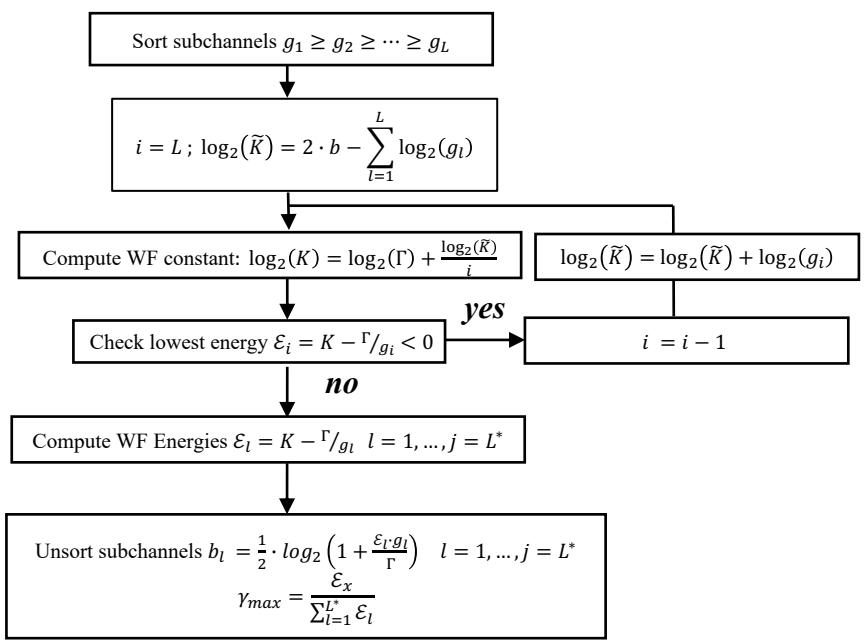


Figure 4.10: Flow chart of margin-adaptive water filling.

Figure 4.10 illustrates margin-adaptive water-filling. Figure 4.10 recognizes that the numerator and denominator terms in Equation (4.70) could be quite large while the quotient remains reasonable and so uses logarithms to reduce the precisional requirements inside the loop.

EXAMPLE 4.3.2 Returning to the $1 + .9 \cdot D^{-1}$ 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

$$K_{ma} = 10^{.88} \cdot \left(\frac{2^{16}}{19.945 \cdot 17.032^2 \cdot 10^2 \cdot 2.968^2 \cdot .05525}\right)^{.125} = 6.3228 \tag{4.73}$$

$$\mathcal{E}_4 = 6.3228 - 10^{.88}/.0552 = -131 < 0 \quad i \to 4 \tag{4.74}$$

2. step $i = 4$ subchannels

$$K_{ma} = 10^{.88} \cdot \left(\frac{2^{16}}{19.845 \cdot 17.032^2 \cdot 10^2 \cdot 2.968^2} \right)^{1/7} = 4.0727 \tag{4.75}$$

$$\bar{\mathcal{E}}_3 = 4.0727 - 10^{.88}/2.968 = 1.5169 > 0 \;\; \mathcal{E}_3 = 3.0337 \tag{4.76}$$

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

$$\gamma_{max} = \frac{8}{3.0337 + 6.6283 + 7.2547 + 3.6827} = (1/2.57) = -4.1 \text{ dB} \tag{4.77}$$

The negative margin indicates that this channel needs coding or more energy (at least 4.1 dB more) to achieve a $P_e = 10^{-6}$, even with MT. However, gap/capacity arguments assure 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 at $P_e = 10^{-6}$, to have acceptable performance on this channel. The amount of additional coding necessary is a minimum with MT (when $N \to \infty$).

Revisiting earlier MIMO matrix-AWGN example for MA loading:

EXAMPLE 4.3.3 *[MIMO Loading]* Returning to Example 4.1.2 and referring to Figure 4.4, the crosstalking channel matrix was

$$H = \begin{bmatrix} 1 & .9 \\ -.8 & 1 \end{bmatrix} \quad . \tag{4.78}$$

The channel's SVD has the channel gains as

$$\Lambda^2 = \begin{bmatrix} (1.3624)^2 & 0 \\ 0 & (1.2624)^2 \end{bmatrix} \quad . \tag{4.79}$$

For $b = 4$ bits/symbol on this channel, MA water-filling produces the water-level constant:

$$K_{ma} = 10^{.88} \left\{ \frac{2^4}{\left(\frac{1.3624^2}{0.1}\right) \cdot \left(\frac{1.2624^2}{0.1}\right)} \right\}^{1/2} = 1.77 \quad . \tag{4.80}$$

The two corresponding energies are then (for minimum energy to attain $b = 4$) are then

$$\mathcal{E}_1 = K_{ma} - \frac{\Gamma}{g_1} = 1.77 - \frac{10^{.88}}{18.49} = 1.36 \tag{4.81}$$

$$\mathcal{E}_2 = K_{ma} - \frac{\Gamma}{g_2} = 1.77 - \frac{10^{.88}}{15.88} = 1.29 \quad , \tag{4.82}$$

Leaving a total energy of $\mathcal{E} = 2.64$ and thus a margin of $\gamma_m = -1.23$ dB. This means the design needs a small amount of coding gain (more than 1.23 dB) to achieve the desired $b = 4$ bits per symbol.

4.3.3 Loading with Discrete Information Units

Water-filling algorithms have bit distributions where b_n can be any non-negative 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*

$$b_n = B_n \cdot \beta \quad , \tag{4.83}$$

where $B_n \geq 0$ is an integer $B_n \in \{\mathbb{Z}^+ \bigcup 0\}$

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.

EXAMPLE 4.3.4 *[DSL rate adaption]* Severak international DSL standards (ITU G.992.1 or ADSL1), ITU G.992.5 or ADSL2+, G.993.2 or VDSL2, G.993.5 Vectored VDSL, G.9901 "G.fast", and/or G.9711 "G.mgfast") use a form of MT described more fully in Section 4.6. These allow high-speed digital transmission on phone lines with one transceiver in the telephone company central office (or some fiber-fed "optical line terminal" in the neighborhood or basement that effectively becomes the copper end/start point) and the other transceiver at the customer. These all have a rate-adaptive operation mode for internet access. The symbol rate is 4000 Hz in these methods, meaning the tone width is close to (see Sections 4.6 and 4.7 4000 Hz. G.fast and G,mgfast use 48kHz instead of 4kHz. These standards use $\overline{N} \leq 8192$. and have granularity $\beta = 1$ bit per two-dimensional subchannel. The ADSL1 standard uses 256 tones in one direction of transmission known as "downstream" and 32 tones upstream, the 8:1 asymmetry being a match to usually asymmetric file transfers and acknowledgment packet-length ratios in IP traffic for internet use. ADSL2+ and VDSL2 use larger number of tones, as do G.fast and G.mgfast with $N \leq 8192$. Section 4.7 investigates duplexing methods.

Thus the data rate can be any multiple of 4kbps (48kbps) 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. Data rates depend on length of channel (gain/attenuation) and can be anywhere from 1 Mbps to 5 Gbps.

Standard	Symbol Rate	$\bar{N}$ down up	N down up	b_{max}	max down data rate up
ADSL1 (G.992.1)	4000 Hz	256 32	512 64	15	8Mbps 800kbps
ADSL2+ (G.992.5)	4000 Hz	512 32	1024 64	15	24Mbps 1.5Mps
VDSL2 (G.993.2)	4000 Hz	4096 4096	8192 8192	15	150Mbps 30Mps
Vec VDSL (G.993.5)	4000 Hz	8192down 8192up	16384 16384	12	150Mbps 30Mps
G.fast (G.9901)	48000 Hz	4096 4096	8192 8192	14	1Gbps 1Gbps
G.mgfast (G.9711)	48000 Hz	8192 8192	16384 16384	14	5Gbps 5Gbps
G.hn (G.9961)	24000 Hz	4096 4096	8192 8192	15	500Mbps 500Mps

The examples are approximate on data rates listed, and there are various spectral (or in G.fast and G.hn both frequency and time) restrictions that apply, and further coding overheads may reduce speeds listed below a simple calculation of assuming maximum bits on all tones. G.hn is used also on power-line copper cables and on coaxial cables. Nonethelss, the basics are illustrative. All these methods use adaptive loading over the subchannels.

4.3.3.1 Optimum Discrete Loading Algorithms

Optimum discrete loading algorithms recognize the discrete-loading problem as an instance of "greedy optimization" in mathematics. The basic concept is that each increment of additional information that a multi-channel transmission system transports should be on the subchannel that requires the least incremental energy for it. Such algorithms are optimum for loading when the information granularity β is the same for all subchannels, which is usually the case.[18]

Several definitions are necessary to formalize discrete loading.

[18]The $1+.9 \cdot D^{-1}$ 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

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*

$$\boldsymbol{b} = [b_1 \; b_2 \; ... \; b_N]^t \quad . \tag{4.84}$$

The sum of the elements in $\boldsymbol{b}$ equals 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[19]. 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 (B_n) of information units corresponding to $b_n = B_n \cdot \beta$ bits on each subchannel can be notationally generalized to the* **energy function**

$$\mathcal{E}_n \rightarrow \mathcal{E}_n(b_n) \quad , \tag{4.85}$$

which explicitly shows the symbol energy's dependence on the number of bits transmitted, b_n. The **incremental energy** *to transmit b_n bits 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*

$$e_n(b_n) \triangleq \mathcal{E}_n(b_n) - \mathcal{E}_n(b_n - \beta) \quad . \tag{4.86}$$

EXAMPLE 4.3.5 [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

$$\mathcal{E}(b_n) = \begin{cases} \frac{2^{b_n}-1}{6} d_n^2 & b_n \text{ even} \\ \frac{2^{b_n+1}-1}{12} d_n^2 & b_n \text{ odd} \end{cases} \quad . \tag{4.87}$$

The incremental energy is then

$$e_n(b_n) = \begin{cases} \frac{2^{b_n}-1}{12} d_n^2 & b_n \text{ even} \\ \frac{2^{b_n}+1}{12} d_n^2 & b_n \text{ odd} \end{cases} \quad . \tag{4.88}$$

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 per 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	$\frac{d_n^2}{4}$	$\frac{d_n^2}{4}$	$\frac{3d_n^2}{4}$	$\frac{5d_n^2}{4}$	$\frac{11d_n^2}{4}$	$\frac{21d_n^2}{4}$	$\frac{43d_n^2}{4}$	$\frac{85d_n^2}{4}$

This table's distance d corresponds to a given desired error probability (and there is thus no gap approximation use).

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.

[19] Such systems can occur in multi-user systems that have essentially suchannels per user as well as frequency. Chapters 2 and 5 particularly address loading for multi-user channels.

Montonic increase of $\mathcal{E}(b_n)$ with b_n is important and clearly realistic because sending more bits should require more energy for a reasonable single-user communications design. Often, and true for QAM in Example 4.3.5, incremental energy $e_n(b_n)$ also increases with b_n. That is, it is increasingly less efficient to load additional bits to any subchannel n in practical constellations (typically twice the existing energy in QAM for each incremental bit). This increasing add-a-bit difficulty leads to the optimality of the discrete loading algorithms that follow:

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

$$\mathcal{E}_n(b_n) = 2 \cdot \frac{\Gamma}{g_n}\left(2^{b_n} - 1\right) \quad . \tag{4.89}$$

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

$$\begin{aligned} e_n(b_n) &= \frac{\Gamma}{g_n}\left(2 \cdot 2^{b_n} - 2^{b_n}\right) && (4.90)\\ &= \frac{\Gamma}{g_n} \cdot 2^{b_n}(2-1) && (4.91)\\ &= \frac{\Gamma}{g_n} 2^{b_n} && (4.92)\\ &= 2 \cdot e_n(b_n - 1) \quad , && (4.93) \end{aligned}$$

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

$$e_n(b_n) = \frac{\Gamma}{g_n} 2^{b_n + .75}\left(2^{.25} - 1\right) \quad , \tag{4.94}$$

and in general for gap-based QAM

$$e_n(b_n) = \frac{\Gamma}{g_n} 2^{b_n + 1 - \beta}\left(2^{\beta} - 1\right) \quad . \tag{4.95}$$

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. The incremental energies are all scaled by the same gap. This will not change their relative sizes in discrete loading, and thus bits will be loaded in the same order for any gap. Thus both capacity-achieving and uncoded systems will add the first bit in the same place, 2nd bit in the same place etc. It is just the total energy (or bit-rate in MA) constraint that determines when the loading stops that would distinguish a coded system from an uncoded system (or more generally two systems with different constant gaps). Smaller gap will imply more bits loaded for any amount of energy, or equivalently less energy to get the same bit rate. For RA loading, smaller gap thus means possibly wider bandwidth (more used tones) while for MA loading, smaller gap causes a more narrow band to be able to achieve the desired data rate. Discrete loading makes this gap interpretation easier to see.

Clearly, there are many possible bit distributions that all sum to the same total b. A highly desirable bit-distribution property 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*

$$\max_n \left[e_n(b_n)\right] \leq \min_{m \neq n} \left[e_m(b_m + \beta)\right] \quad . \tag{4.96}$$

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-Campello (LC) "Efficientizing" (EF) Algorithm Levin and Campello [3] have independently formalized an iterative algorithm that will translate any bit distribution into an efficient bit distribution:

1. $m \leftarrow \arg\{\min_{1 \le i \le N} [e_i(b_i + \beta)]\}$;
2. $n \leftarrow \arg\{\max_{1 \le j \le N} [e_j(b_j)]\}$;
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) $m \leftarrow \arg\{\min_{1 \le i \le N} [e_i(b_i + \beta)]\}$;
   (d) $n \leftarrow \arg\{\max_{1 \le j \le N} [e_j(b_j)]\}$;

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 produces an efficient bit distribution if the energy function and the incremental energy function are both monotonically increasing with information (as always the case in practical systems). Truncation of a water-filling distribution also produces an efficient distribution, and the total of fractional bits remaining will still need allocation.

EXAMPLE 4.3.6 [Return to $1 + .9D^{-1}$ for EF algorithm] By using the using the uncoded gap approximation at $P_e = 10^{-6}$, the PAM and Nyquist SSB subchannels' energy is given by

$$\mathcal{E}_{PAM,n}(b_n) = \frac{10^{.88}}{g_n}\left(2^{2b_n} - 1\right) \quad n = 0, 4 \ , \tag{4.97}$$

and the QAM subchannels' energies are

$$\mathcal{E}_{QAM,n}(b_n) = 2 \cdot \frac{10^{.88}}{g_n}\left(2^{b_n} - 1\right) \quad n = 1, 2, 3 \ . \tag{4.98}$$

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*

$$0 \leq N \cdot \bar{\mathcal{E}}_{\boldsymbol{x}} - \sum_{n=1}^{N} \mathcal{E}_n(b_n) \leq \min_{1 \leq i \leq N} [e_i(b_i + \beta)] \quad . \tag{4.99}$$

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

The **Levin-Campello E-tightening (ET) algorithm** makes a bit distribution E-tight through this algorithm:

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 $\left(N\bar{\mathcal{E}}_{\boldsymbol{x}} - S \geq \min_{1 \leq i \leq N} [e_i(b_i + \beta)]\right)$
   IF $\left(N\bar{\mathcal{E}}_{\boldsymbol{x}} - S < 0\right)$ THEN
   (a) $n \leftarrow \arg\{\max_{1 \leq i \leq N} [e_i(b_i)]\}$;
   (b) $S \leftarrow S - e_n(b_n)$
   (c) $b_n \leftarrow b_n - \beta$

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

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

EXAMPLE 4.3.7 *[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 halves in this example for E-tightness and the resultant margin is $8/5.32 = 1.8$dB.

Application of the ET algorithm to an efficient distribution solves the RA problem :

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 appears in Appendix G:

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

  Levin Campello's Method for a temporal channel

  Inputs
  p is the time-sampled pulse response
  SNRmfb is the SNRmfb, so EXbar*norm-p^2/sigma^2, in dB
  Ex_bar is the normalized energy
  Ntot is the total number of real subchannels, Ntot>2
      The program user must adjust results for guard-extension losses
  gap is the gap in dB

  Outputs
  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 used subchannels
  b_bar is the bit rate per real dimension

  The first and last tones are PAM, the rest of them are QAM.
```

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*

$$b = \sum_{n=1}^{N} b_n \quad . \tag{4.100}$$

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

B-Tightening Algorithm: 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) $n \leftarrow \arg\{\max_{1 \leq i \leq N} [e_i(b_i)]\}$;

(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 4.3.8 **[Bit tightening on $1 + .9 \cdot D^{-1}$ 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 is -4.3 dB. The total energy is 21.6.

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

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, computation of the incremental energy table includes this redundancy's effect. This can be difficult to compute and depends upon code choice. 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. Apart from the difficult incremental-energy table's calculation, RA LC and MA LC algorithms remain the same. Often a code with some fixed overhead (that is fixed number of total bits, including parity) is chosen independent of loading when MA LC is used. RA's incremental energies $e_n(b_n)$ will assume some fixed lower gap, effectively assuming parity (extra) coding bits to be a fixed percentage of total bits.

A program for MA LC is called MALC.m and is in Appendix G:

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

  Margin Adaptive Levin Campello Loading

  p is the pulse response
```

```
SNRmfb is the SNRmfb, so Ex_bar*norm(p)^2/sigma^2, in dB
Ex_bar is the normalized energy
Ntot is the total number of real subchannels, Ntot>2
  Any guard periods must be addressed by program user
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 (in dB)

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

The same loading algorithms exactly apply to MIMO channels also with their associated set of channel gains g_n for $n = 1, ..., L_x$.

4.3.3.2 Power-Spectral Density Constraints with LC Loading

Levin Campello algorithms also accommodate additional constraints on loading in the incremental-energy table. A common limit is on the power-spectral density (for instance by a national regulatory authority or commission that regulates spectrum and/or emissions levels). The spectrum constraint for each MT tone is then

$$\mathcal{E}_n \leq \mathcal{E}_{max,n} \;\; \forall \, n \;\; . \tag{4.101}$$

Sometimes the constraint is simply "flat" at $\mathcal{E}_{max,n} = \mathcal{E}_{max}$. However in many practical cases, the constraint will vary over the transmission band used, and thus will not be "flat."

Insertion of large (infinite) incremental energies in the incremental-energy table when the corresponding next energy increment would exceed (4.101)'s power-spectral-density (PSD) constraints then optimally allows LC algorithms to avoid exceeding the mask. The loading otherwise proceeds as always by choosing the next smallest-energy tone to add an information unit (bit), but then effectively avoids such increments that violate the power-spectrum limit.

Water-filling under (4.101) fills a bowl with a shaped lid so energy may fill to the lid. This indeed is optimum and satisfies all the Kuhn-Tucker conditions, but may be difficult to compute in that there is no longer a water-filling constant. It is also more difficult to describe the shape of the bowl's lid, but it is possible as follows: The RA and MA (with last step of energy increase in the MA case observing PSD limits) algorithms would proceed as in the earlier flow charts. However, when a negative-energy test is performed for a given value of K on the next tone, that tone needs also to be tested for the PSD constraint. If a PSD violation occurs, the algorithm removes that tone from further consideration and fixes the energy at the PSD limit. Water-filling then continues without further loading this tone and with that tone's energy otherwise set at the PSD limit. All remaining tones follow with the remaining energy, water-filling and checking PSD. Clearly Levin Campello appears look more attractive as it more easily accommodates the PSD constraints in the incremental energy table without the extra power-spectral-density checking ongoing (but does require more work to construct that table).

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. The sort of the vector $\boldsymbol{g}$ (the vector $[g_1 \, , ..., \, g_N]$) is

$$\boldsymbol{g}_{sort} = J \cdot \boldsymbol{g} \tag{4.102}$$

where J is again a permutation/ordering matrix (as in Chapter 2) with one and only one nonzero unit value in each and every row and column. For instance, the $N = 3$ matrix

$$J = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \tag{4.103}$$

takes the bottom row's position and moves it to the top row, moves the top row's position to the middle row and finally places the middle row last. To unsort,

$$\boldsymbol{g} = J^* \cdot \boldsymbol{g}_{sort} \tag{4.104}$$

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

Loading begins with a sort of the channel gains in $\boldsymbol{g}$, so sort g_n. Finding a maximum in a set of N numbers takes $N - 1$ comparisons and then subsequently finding the next maximum takes $N - 2$ comparisons. Continuing in such fashion requires $N^2/2$ comparisons to order the g_n. 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 repeats 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 $Nlog_2(N)$.

Finding the transition point: 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 sort into two subsets greater than or less than or equal to the transition point. Then loading 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.

Run-time bit-swapping complexity: 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 $2N$ comparisons). This number of comparisons could be quite high for each successive bit loaded. Campello further studied sets of subchannels, 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 thus all have indices' 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. 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 search. This is the topic of the next subsection.

4.3.5 Discrete Vector Loading and Variable Gaps

Discrete loading's incremental energy tables' entries can be expanded to correspond to adding information units across more than a single (complex or real) dimension. Such expansion will make the incremental energy table more complex to compute. However, once computed, the LC algorithms proceed as before. With or without dimensional expansion to **discrete vector loading**, the gap may also

become variable for each group or "vector" of dimensions, $\Gamma \rightarrow \Gamma_n$. Variable gap would simply mean the incremental energy table's columns of energies would scale by the different gaps and would correspond to different-gain codes being used on different dimensions (or vectors of dimensions). Such situations can arise when other non-AWGN constraints could be of importance. For instance certain dimensions might be susceptible to periodic or intermittent noises (Gaussian or non-Gaussian) and might benefit from a different code's use (or different gap) on only those dimensions.

Vector loading may also be appropriate where codes based on multidimensional symbols are used. For instance some combined modulation/codes in Chapters 9 - 11(known as trellis codes and lattice codes) are based on 4, 8, 16 or even larger dimensional symbols.

In general, a set of N real dimensions in a multidimensional system might contain groups of N_j dimensions where

$$N = \sum_{j=1}^{J} N_j \quad . \tag{4.105}$$

The jth group or "vector" of N_j dimensions may use the same-gap code ($\Gamma_j = \Gamma$) or different-gap codes (Γ_n not constant with j) than the other groups. The overall symbol period remains T for transmission of signals on N dimensions. The incremental energy tables in vector loading instead have indices j for $j = 1, ..., J$ and thus have incremental energies for adding the bth bit (or information unit) that is $e_j(b)$. The calculation of these incremental energies may be more complex than simple uncoded PAM or QAM. Otherwise LC loading proceeds as before with bits added in places of least next incremental energy and deleted in places of most decremental energy.

The optimality proof of LC loading depends on the successive energy increments' monotonicity; that is $e_j(b+1) \geq e_j(b)$ for all b and j. While uncoded PAM/QAM have this monotoniciy, it may not occur with certain types of codes. Most codes have redundancy, usually explicitly in the form of parity bits. The parity bits (or at least some of them if the code is punctured) will be present even if only 1 bit is loaded to that coded group, so that the **number of code-output bits** is

$$b_{c,j} = b_j + p_j \quad , \tag{4.106}$$

where b_j is the number of information bits in group j and p_j are the parity bits. To load even the first bit so that $b_j = 1$ will require that the parity bit also be loaded to the group/vector. Thus, at least 2 bits (information units), if $p_j = 1$, would need to be the full cost of that first information bit loaded. Subsequent bits may be added without adding additional parity bits. Choices of codes used in loading should ensure that the incremental energies are monotonic to ensure LC's optimality. Thus puncturing may be necessary for smaller b_j to ensure each column in the incremental energy table is monotonically increasing. Usually this is fairly easy to accomplish without reducing the coding gain if the codes are well chosen for the application. (For instance the 4D and 8D Wei codes of Chapter 10 will have this property with judicious puncturing. Most good convolutional codes will also allow this. However, it is not guaranteed with any choice, so designer beware.) An example of such vector loading (4 dimensions) is considered in Section 4.7 for xDSL.

When DMT (and not Coded-OFDM) uses an outer codes, a table of AWGN-equivalent "gross" coding gains can be computed (often by simulation) as Howard Levin did for LC loading with Reed Solomon byte-wise outer codes. Essentially this gain should be removed from the gap when loading to ensure that the energy (RA) or bit (MA) constraint is approximately met correctly for the code. Similar tables could be created for any other type of outer code, including LDPC codes, bit-interleaved codes etc. Coded-OFDM does not have loading (in effect $N_{J=1} = N$), and this issue need not be addressed directly. Instead, it is addressed indirectly as in the next subsection.

4.3.5.1 Embedding Equivalent Channels

An alternative form of such loading views each group as an equivalent AWGN channel with a single geometric SNR. There are in fact N_j such identical single-dimensional channels that represent the group, each carrying an equal number of bits per dimension with this same geometric-equivalent SNR. As the input energy to that group increases, each of the channels can equally carry more information, so an equivalent g_n can represent the group. Adding one bit to the equivalent channel is equivalent to adding

N_j bits. If all the equivalent channels have the same number of dimensions within the group, then LC could be applied to an incremental energy table for the set of N/J equivalent subchannels. There will be then a single equivalent channel that is comprised by the N/J equivalent channels.

Perhaps the simplest example then of this is OFDM where $J = 1$ and $N_{J=1} = N$. The OFDM dimensions all use the same number of bits. Loading is simply determined by the formula

$$b_{ofdm} = \frac{N}{1} \log \left(1 + \frac{\text{SNR}_{ofdm}}{\Gamma} \right) \quad . \tag{4.107}$$

Clearly this is a lower bound on performance if water-filling over the larger set of individual dimensions could have been used for DMT. Always (for same code),

$$\text{SNR}_{dmt} \geq \text{SNR}_{ofdm} \tag{4.108}$$

trivially follows. The Wi-Fi and LTE examples of Section 4.7 will observe this kind of coded-OFDM loading.

The (MIMO) Vector OFDM systems later-to-come in Subsection 4.7.3 are another embedded-equivalent-channel example where each of $J = L$ groups has N dimensions each over the set of LN total dimensions. Different numbers of total bits pass over each of the L spatial dimensions, so that adaptive loading occurs in space and fixed-redundancy codes to mitigate frequency-selective fading. Each of the L equivalent channels has its own geometric SNR and equivalent g_n that can be used for loading.

4.3.6 Chow's Dynamic Loading

In some applications, the gains, 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". Dynamic loading was first studied by Dr. Peter Chow in his 1992 Stanford thesis [13].

It is conceivable that an entire loading algorithm might execute every time there is a change in a channel's bit and/or energy distribution. However, if the channel variation is small, then "reloading" might produce a very similar bit distribution. 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 reverse channel's relative data rate. 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 unless the feedback channel is sufficiently rapid.

However, space-time wireless MIMO channels may change relatively (over space) more slowly than SISO channels, basically because L_x times more channel measurements assist tracking with respect to single-input-single-output wireless systems. Thus loading information's feedback (or as we'll see later, also partitioning information) may occur in some wireless MIMO transmission systems.

LC algorithms lend themselves most readily to incremental variation. Specifically, for continuous MA solution, the previous bit distribution $\boldsymbol{b}$ can be input to the EF algorithm as an initial condition for the new incremental-energy table of $e_n(b_n)$, 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 channel variation, then the EF algorithm will continue to converge to the optimum finite-granularity solution. The incremental information consists of the two swapped dimensions' indices and which to decrement. 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 energy added/deleted from swapping subchannels can be computed) with use of the EF algorithm.

Bit-swapping systems in practice also allows the addition of a bit or the deletion of a bit without a corresponding action on a second subchannel, 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 can occur, rather than leave it to the channel). These DRA 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 can often be undesirable.

However, such true DRA can occur through the execution of the ET algorithm theoretically. That algorithm will change the suchannels' total number of bits to conform to the best use of limited transmit energy. The ET algorithm maintains resultant bit distribution's efficiency. The incremental information passes 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 systems implement a dynamic fixed-margin adaption where constant data rate is maintained, but power reduces/increases 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.

4.3.6.1 Synchronization of dynamic loading

With dynamic loading, both transmitter and receiver must know on which symbol a bit/energy distribution change occurs. Thus the system maintains two synchronized symbol counters, 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 might otherwise take more time).

4.3.6.2 The multichannel normalizer and computing the new subchannel SNR's

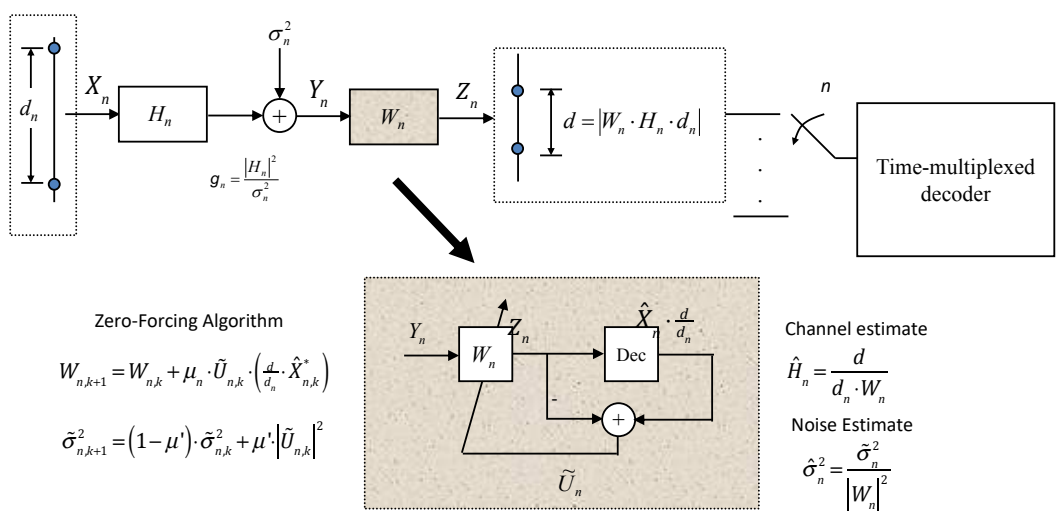


Figure 4.11: A multichannel normalizer for one subchannel, n.

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. With MT, W_n is complex and often has name **Frequency EQualizer (FEQ)**. The multi-channel normalizer may allow a single decision device to be time-shared for decoding all the subchannel outputs. The normalization of distance satisfies:

$$d = |W_n \cdot H_n \cdot d_n| \quad , \tag{4.109}$$

where d_n is the minimum distance between points on the nth 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,

$$W_n = \frac{d}{H_n \cdot d_n} \quad , \tag{4.110}$$

will maintain (4.109)'s magnitudes. However, since the subchannel may change its gain or SNR, the value for W_n is adapts periodically (as often as every symbol is possible). The error, $\tilde{U}_n$ between the input and output of decision device shown in Figure 4.11's enlarged normalizer box, estimates the normalized noise (as long as the decision is correct). The normalizer may update W_n in a number of ways that average several successive normalized errors, of which the two most popular are the **zero-forcing algorithm**

$$W_{n,k+1} = W_{n,k} + \mu_n \cdot \tilde{U}_n \cdot \frac{d}{d_n} \cdot \hat{X}_n^* \quad , \tag{4.111}$$

which produces an unbiased estimate of the signal at the normalizer output and converges[20] when (and allowing different step size to be set on different dimensions)

$$0 \leq \mu_n < \frac{|H_n|^2 \cdot \mathcal{E}_n}{2} \quad , \tag{4.112}$$

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

$$W_{n,k+1} = W_{n,k} + \mu_n \cdot \tilde{U}_n \cdot Y_n^* \quad , \tag{4.113}$$

which instead produces a biased estimate and an optimistic MMSE-SNR estimate (See Appendix D.2 where this SNR is $\text{SNR}_n + 1$). Multichannel normalization usually uses the ZF algorithm because there is it corresponds trivially to bias removal issue with a single scalar estimate. The factor μ is a positive gain constant for the adaptive algorithm selected to balance time-variation versus noise averaging.

The receiver estimates the channel gain according to

$$\hat{H}_n = \frac{d}{d_n \cdot W_n} \quad . \tag{4.114}$$

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

$$\tilde{\sigma}_{n,k+1}^2 = (1 - \mu') \cdot \tilde{\sigma}_{n,k}^2 + \mu' \cdot |\tilde{U}_n|^2 \quad , \tag{4.115}$$

where μ' is another averaging constant and

$$0 \leq \mu' < 1 \quad . \tag{4.116}$$

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

$$\hat{g}_n = \frac{|H_n|^2}{\frac{\tilde{\sigma_n}^2}{|W_n|^2}} = \frac{d^2}{d_n^2} \cdot \frac{1}{\tilde{\sigma_n}^2} \quad . \tag{4.117}$$

[20] A convergence proof notes that with stationarity $E[W_n] = E[W_n] + \mu_n E\left[\left(\frac{d}{d_n} \cdot X_n - W_{n,k} Y_{n,k}\right) \cdot \frac{d}{d_n} \cdot X_n^*\right] = \left(1 - \mu_n \frac{d}{d_n} H_n \mathcal{E}_n\right) E[W_n] + \mu_n \frac{d^2}{d_n^2} \mathcal{E}_n$, A simple first-order recursion in average FEQ value. This will converge to $E[W_n] = d/(d_n \cdot H_n)$ when the geometric ratio is less than 1. The approximation that d/d_n is roughly the channel gain $H_n|$ is used in Equation (4.112).

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

$$\frac{\frac{\Gamma}{\hat{g}_{n,new}}}{\frac{\Gamma}{\hat{g}_{n,old}}} = \frac{\hat{g}_{n,old}}{\hat{g}_{n,new}} = \frac{\tilde{\sigma}^2_{n,new}}{\tilde{\sigma}^2_{n,old}} \quad . \tag{4.118}$$

Thus the system estimates the new incremental-energy entries by scaling the old values by the relative increase/decrease in Equation (4.118)'s normalizer noise. Often the "old" value is an initial value stored at system initialization, rather than the most recent previous-noise-variance estimate. This reference to initial value then avoids a cascade of "finite-precision" errors and precisely coordinates the transmitter and the receiver.

4.3.6.3 Chow's real-time "bit-swapping"

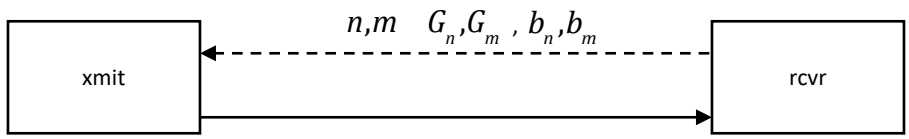


Figure 4.12: Illustration of reverse channel in bit-swapping.

If the loading algorithm moves or "swaps" a bit (or more generally information unit) from one subchannel to another, then both the multi-channel transmitter and receiver should implement this "bit-swap" on the same symbol, as in the last subsection. The receiver initiates a bit-swap by sending a pair of subchannel indices n (subchannel where bit is deleted) and m (subchannel where bit is added) through some reverse channel to the transmitter, as per the dashed line in Figure 4.12. Receivers often also send the number of bits, b_n, through the reverse channel. The individual quantities b_n rarely take more than 4 bits ($0 \le b_n \le 15$) to convey how many bits to be swapped, added, or deleted in each such command for this minimal additional reverse-channel overhead. There is often also an associated small change in transmitted energy on each subchennel:

$$G_i = \frac{\mathcal{E}_i(new)}{\mathcal{E}_i(old)} \quad \text{for } i = m, n \quad . \tag{4.119}$$

The energy for the bit-swap subchannelss $i = m$ or $i = n$ can be recomputed by summing the new increments in the receiver's incremental-energy table,

$$\mathcal{E}_i(new) = \frac{\hat{g}_{i,old}}{\hat{g}_{i,new}} \cdot \sum_{j=1}^{b_i} e_{i,old}(j) \quad . \tag{4.120}$$

Typically this energy-scaling factor is near unit value and usually less than a 3 dB/bit change for 2-dimensional subchannels like those of MT (and less than 6 dB/bit change for one-dimensional subchannels). The near-unity-value occurs because subchannel 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 a single bit 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 subchannel n. However, it could be that the gain is not exactly 2. Thus, the gain factor G_i in Equation (4.119) should also be conveyed through the reverse channel to the transmitter. Again, 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-to-last gain could lead to a chain of cascading finite-precision errors.

For RA or MA operation, the system selects the two subchannel's gain factors to produce a total transmit energy that is exactly equal to the allowed limit. The receiver also nominally reduces the gain normalizer coefficients by the inverse square-root of this energy factor as

$$W_{n,new} \rightarrow \frac{1}{\sqrt{G_n}} \cdot W_{n,old} \quad . \tag{4.121}$$

However, this may not include distance adjustment: Equation (4.114) reorganizes to

$$W_n = \frac{d}{d_n \cdot \hat{H}_n} \quad . \tag{4.122}$$

The distance d is constant in the receiver, but the transmitter's d_n can change. ($\hat{H}_n$ may change, but (4.111)'s adaptation of W_n already addresses this channel transfer change that that lead to the swap, so (4.122) already includes W_n's change. The as-yet-uncompensated change in distance d_n is usually roughly a factor of 2, but this value is exactly known to the loading-algorithm control as

$$\Delta d_n = \frac{d_{n,new}}{d_{n,old}} \quad , \tag{4.123}$$

as is the gain factor $G_{n,new}$ relative to the old (initial) transmit gain $G_{n,inital}$, usually 1. When a swap is executed, the values for W_m and W_n and the corresponding noise estimates need one-time scaling by an additional factor: Thus new normalizer values for $W_{n,k}$ and $\tilde{\sigma}^2_{n,k}$ that will enter the updates (Equations (4.122) and (4.123)) as the current normalizer value are

$$W_n(\text{new}) \rightarrow W_n(\text{old}) \cdot \frac{1}{\Delta d_n \cdot \sqrt{G_n}} \quad , \quad \tilde{\sigma}^2_n(\text{new}) \rightarrow \tilde{\sigma}^2_n(\text{old}) \cdot \frac{1}{(\Delta d_n)^2 \cdot G_n} \quad , \tag{4.124}$$

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

The accuracy of channel estimation is important and discussed in Chapter 7. This section assumes all normalizer quantities have no estimation error.

4.3.6.4 Gain Swapping

In addition to the previous subsection's gain changes consequent to a bit-swap, gain may also change in the absence of a bit-swap. This 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 an information unit β. On a two-dimensional subchannel with $\beta = 1$, this cost could be 3 dB of energy for a particular subchannel. This means that a particular subchannel's error probability of 10^{-6} could degrade by as much as roughly 3 orders of magnitude before the swap occurs. A particularly problematic situation would be a 2.9 dB loss that never grew larger and so no swap occurs. 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 significantly effect the overall error probability.

There are two solutions to this potential problem: The first reduces β, but this may be difficult. The second increases transmit energy on one subchannel while reducing the same amount from another subchannel. Again, the factor G_n conveys the level of the "gain swap," relative to $G_n(old) = 1$. Furthermore, the presence of a dramatically increased noise (for instance a noise caused by another system turning on or off) might lead to a completely different energy distribution of energy when loading. Thus, a dimension's transmitted energy (or equivalently the relative energy-change factor applied to such a subchannel) may need to change significantly. 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[21] instead of the nominal level, so $G_n(new) = 0$. If a gain-swap occurs, then the two columns of the incremental energy table needed to be scaled by the respective factors – and indeed this scaling may subsequently cause a bit-swap to occur. One channel change may produce many gain swaps.

[21] A small "keep-alive" amount allows monitoring of when the the subchannel returns to potential good use.

4.3.7 Use of wider bands than waterfill

A common misunderstanding in communication design confuses the optimality of mutual information for any given input spectrum with the optimal water-filling spectrum choice. This mistake occurs so often that this subsection provides a simple specific example that hopefully elucidates the misconception. Figure 4.13

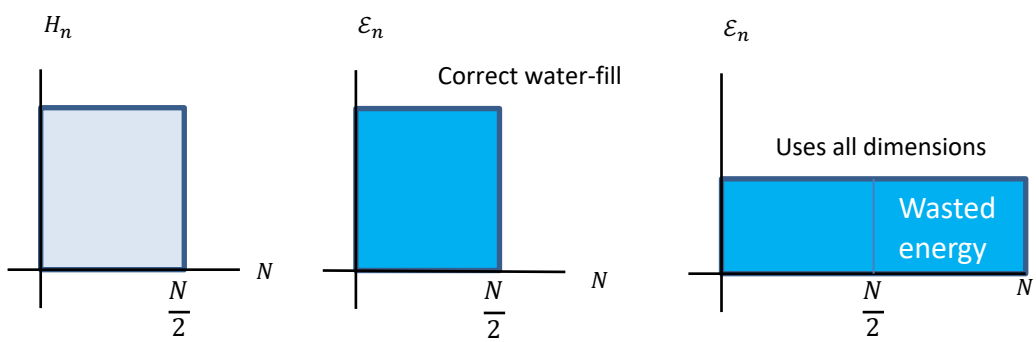


Figure 4.13: The half-band water-fill spectrum.

Figure 4.13 shows a "flat" channel that trivially has water-fill band equal to the channel's nonzero passband. Energy placement outside this band wastes energy as it does not pass to the channel output and could better apply to the useful band. While the use of a "brickwall" transmit filter to approximate the correct transmit bandwidth might find utility, Section 3.12 deals with the various realization difficulties and their implications for a single-carrier system of this mistake with respect to simply energizing only the water-fill dimensions. Here Figure 4.14 illustrates the impact of non-ideal codes with a gap or with a margin (or both).

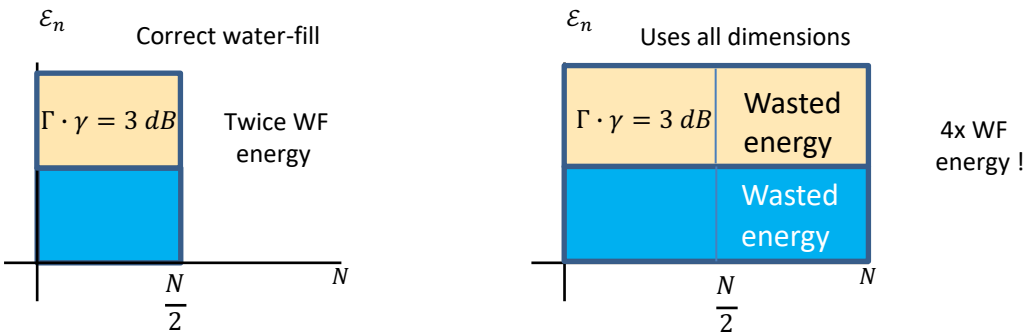


Figure 4.14: The half-band water-fill spectrum.

Margins and non-ideal codes occur in all field systems' use. The margin's application to the energy already wasted on non-water-fill dimensions simply amplifies the poor choice. Additional energy on the useless dimensions simply does not help the design. While clear in this simple context, the mistake appears often. Indeed, many wireless systems today, especially those with wider bandwidths or channel aggregation of multiple bands may apply equal energy to almost the entire temporal-frequency band allocated, even if fading or channel effects produce a different water-fill spectrum. The non-ideality of

the codes (and any outage or system margins) amplifies on the bands that are not water-filling. However, as the next section shows, flat energy on the correct water-fill dimensions (even if the channel is not flat) and zero energy on those water-filling zeros is essentially very close to optimum. This does not mean, as many wireless designs assume, "flat energy everywhere with a good code is optimum." Many wireless designs assume the water-fill bandwidth is close to what is fixed and used. When that is the case, there is little loss. Section 3.12 provides a plot of some of the massive losses when this decision is incorrect.

The same analysis also applies to spatial dimensions. Application of flat energy on all spatial streams/dimensions is only optimum and best MIMO design if water-fill would also use those same dimensions. Fortunately, wireless systems often do allow zeroing of some spatial streams to approximate this water-fill-on-spatial-dimensions effect, although this is sometimes confused with "attaining" diversity. Application of energy to a band that is non-water-fill does not help in any way or form. If the location of zeroed water-fill is unknown, spreading of energy everywhere can ensure that some passes to channel output no matter what, but it does not mitigate the loss otherwise.

4.4 Coded Loading with Constant Constellation Size

The per-dimensional mutual-information data rates, $\overline{\mathcal{I}}_n$, achieved by Section 4.3's optimized loading depend only on the product of the calculated energy distribution values $\mathcal{E}_n$ and the channel gains, g_n, according to

$$\overline{\mathcal{I}}_n = \frac{1}{2} \cdot \log_2 \left(1 + \bar{\mathcal{E}}_n \cdot g_n\right) \quad . \tag{4.125}$$

This section has $\widetilde{N} = 1$ (so then $\overline{N} = N$) for code application to any, or across all, dimensions a multi-dimensional parallel-channel set. Any energy distribution $\{\mathcal{E}_{n=1,\ldots,N}\}$ (optimized or not) for a given gain set $\{g_{n=1,\ldots,N}\}$ produces a corresponding set of mutual-information data rates according to (4.125). These mutual-information data rates correspond to use of a capacity-achieving Gaussian-input code on each AWGN subchannel independently, so $\Gamma = 0$ dB. When $\Gamma > 0$ dB, then the product $\mathcal{E}_n \cdot g_n$ in (4.125) reduces by Γ to approximate the data rates achievable with the applied constant-gap codes (like PAM/QAM with gap $\Gamma = 8.8$ dB at $P_e = 10^{-6}$). At capacity ($\Gamma = 0$ dB), the maximizing Gaussian input distribution for each independent dimension has a possibly large number of per-dimensional constellation-point choices from which on average the encoder selects one of $2^{\overline{\mathcal{I}}_n}$ corresponding to $\overline{\mathcal{I}}_n$ input bits/dimension for that dimension, as per Sections 2.1-2.3. From the on-average $2^{\overline{\mathcal{I}}_n}$ values, these selected codeword subysymbol constellation values need not be the same in each dimension, and a good code's encoder selects them from some larger single-dimension constellation[22] of size $|C|$ (which have selected-value probabilities that follow a Gaussian distribution as in Chapter 2, which means larger-energy values are less likely to be selected). Indeed, an encoder can accept a constant $\overline{\mathcal{I}} = (1/N) \cdot \sum_n \overline{\mathcal{I}}_n$ bits for each dimension to be transmitted and select one of $2^{\overline{\mathcal{I}}}$ values for each and every dimension, completely removing the dependency on n or g_n from the bit-to-subsymbol encoder $C(\boldsymbol{v})$ in Figure 4.15. Such a constant-constellation design may retain the per-dimensional energy $\bar{\mathcal{E}}_n$, which often is constant at $\mathcal{E}_n = \mathcal{E}_x/N \; \forall n$, but not necessarily so. Figure 4.15 shows this energy allocation as the average number of bits that corresponds to the overall geometric SNR in (4.24) for the set of of MT dimensions. The encoder output has $\log_2 |C|$ values for each dimension, as in Figure 4.15, where

$$|C| \geq 2^{\overline{\mathcal{I}}} \quad . \tag{4.126}$$

A more specific form of this system is known as **Coded OFDM or C-OFDM** as in Section 4.7. From Section 2.1, $|C| = 2^{\bar{b}+\bar{\rho}}$ where $\bar{b} \leq \overline{\mathcal{I}}$ and $\overline{\mathscr{H}}_x < 2^{\bar{b}+\bar{\rho}}$.

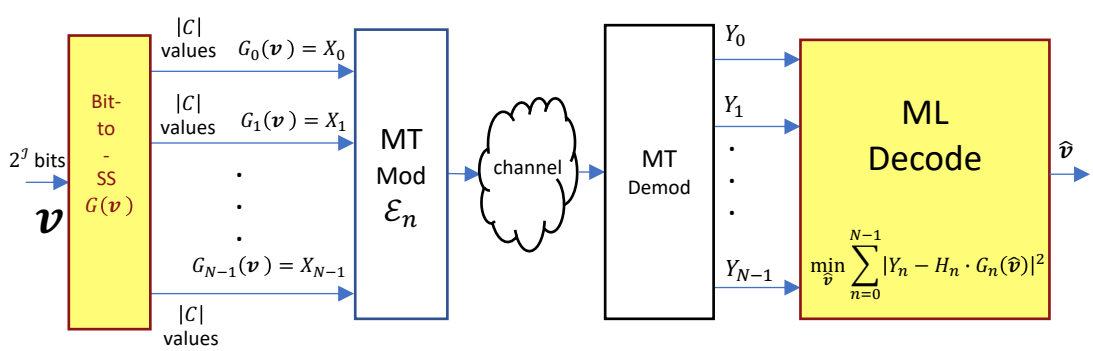


Figure 4.15: Coded Mutlitone. (A specific form is Coded-OFDM in Section 4.7)

When the index n counts real dimensions, as in Figure 4.15, there are often two identical-gain real dimensions for two-connected n values in a complex-baseband system.

[22] Precisely $|C|^{1/\widetilde{N}}$, but $\widetilde{N} = 1$ in this section.

The constant-constellation coded-loading alternative: Figure 4.15's encoder $C(\boldsymbol{v})$ can introduce a dependency between the MT dimensions through the channel-input codewords. That is - the MT modulator/demodulator and channel still partition the channel into independent dimensions, but a specific encoder may introduce inter-dimensional dependency so the set of parallel channels may no longer be fully independent.[23] Thus, an independent decoder for each dimension's output may no longer be maximum likelihood - Sections 4.1 - 4.3's decoder-implementational independence is a simplification that occurs when an independent encoder is used on each otherwise-independent subchannel as in Section 4.1's MT. Instead in this Coded-MT case, the receiver's ML decoder must consider all the $\overline{N}$ dimensions together. However, as long as the encoder uses a 1-to-1 mapping from input bits to symbol values without change of spectra (or equivalently $\{\mathcal{E}_n\}_{n=0,\ldots,\overline{N}-1}$), the corresponding ML decoder has no loss in information, data rate, nor increase in $\bar{P}_e$ (see Chapter 1's reversibility theorem). The equivalent geometric SNR and data rate $\overline{\mathcal{I}}$ remain the same for the system, as long as the decoder implements a maximum-likelihood dectector (now possibly more complex).

Subsection 2.3.6's AWGN-Channel Convergence Theorem states that the solution (recalling $\overline{N} = N/2$ here) to

$$\text{ML: choose: } \hat{\boldsymbol{X}} \ni \min_{\{\hat{\boldsymbol{X}}\}} \sum_{k=-\infty}^{\infty} \sum_{n=1}^{N^*} \left| Y_{n,k} - H_n \cdot \hat{X}_{n,k} \right|^2 \quad , \tag{4.127}$$

is MAP/ML/MMSE on average over good codes randomly selected from a Gaussian distribution with the input autocorrelation $R_{\boldsymbol{X}\boldsymbol{X}}$. or more precisely stated, this decoder minimizes the approximate mean square error between channel output and noise-free filtered codeword (exact MMSE as $N \to \infty$) over all the possible codeword choices $C_{n,k}(\hat{\boldsymbol{v}})$:

$$MMSE \cong \min_{\hat{\boldsymbol{v}}} \frac{1}{N} \sum_{k=-\infty}^{\infty} \sum_{n=1}^{N^*} |Y_{n,k} - H_n \cdot C_n(\hat{\boldsymbol{v}}_k)|^2 \quad . \tag{4.128}$$

Critical Step: The critical step recognizes that a MMSE estimate of each dimension (within a MT symbol or as repeated over many instances of such) accounts for the dimensionally dependent channel weighting of H_n. While a specific code, from the set of good codes that random coding produces, will still need a decoder as in Figure 4.15, the MMSE estimate preserves the mutual information with all the codes (again see Subsection 2.3.6's discussion, or also Appendix D.2). While the average number of bits encoded per dimension remains $\overline{\mathcal{I}}$, the constellation need simply be large enough to accommodate the specific good code – that is, this good code has sufficient redundancy with respect to the average $\mathcal{I}$ that corresponds of course to an equivalent repeated use of a single channel with $SNR_{geo} = SNR_{mmse}$. Thus, with a good code, and any $R_{\boldsymbol{x}\boldsymbol{x}}$ (so any spectrum), Coded-MT will achieve the best possible data rate for that input spectrum. Of course, a waterfilling spectrum choice further improves the capacity to its maximum value, at which level the result still holds.

Intuitive Explanation: C-MT codewords $C(\boldsymbol{v})$ in a good code result, on average, in constellation subsymbol values with roughly equal minimum separation/dimension between different $\{\boldsymbol{X}_{n=0,\ldots,N-1}\}$ codeword values at the dimension's inputs. These average separations (each dimension's inter codeword distances) undergo a gain H_n that will amplify (or attenuate) that dimension. The corresponding value for Y_n will also typically be larger in (4.128) for large-H_n dimensions and smaller for small H_n dimensions because the AWGN has the same (usually smaller) variance in all dimensions. When the ML/MMSE

[23] This concept may be somewhat confusing because most encoders do indeed produce outputs that are uncorrelated from dimension to dimension even though there are clearly dependencies used to increase the distance between possible codewords. If the process has a Gaussian distribution however, uncorrelated samples means independent. This apparent contradiction is addressed by recalling that, for a specific code, all codewords consist of discrete samples from this distribution and thus are not truly Gaussian. Only the ensemble average over all random such codes is fully Gaussian. The corresponding ML receiver, for any given sample codeword sequences that could be transmitted, exploits dependencies between a codeword's dimensions that occur even though their sample cross-correlations may average to zero over all codewords that could have been selected. Thus, the ML decoder then needs to consider all dimensions.

decoder compares a possible incorrect scaled codeword, $H_n \cdot C_n(\hat{v})$, to the channel output Y_n on a large-H_n dimension, that dimension's contribution to the sum in (4.128) is correspondingly large and thus more greatly impacts the full-squared MMSE in (4.128), and thus has greater impact on the selected $\hat{\boldsymbol{X}}$.

AWGN Log-Likelihood Interpretation: A code sequence's log-lilkelihood[24] is

$$LL(\boldsymbol{y}) \propto \sum_n \underbrace{|y_n - H_n \cdot \hat{x}_n|^2}\, LL_n \tag{4.129}$$

individual subsymbol log likelihoods are the terms LL_n, which will be larger for large values of H_n where the channel is stronger and so any decoder will exploit this log-likelihood information (sometimes called "soft information" as in Chapter 7). Thus while the individual dimensional constellations may be the same and carry the same constant $\tilde{b}_n = \tilde{b}$, their decoder impact is larger.

Mathematically, a very good code that evenly distributes squared codeword separation ϵ to all dimensions, thus includes terms like

$$|H_n|^2 \cdot |C_n(\hat{\boldsymbol{v}}_1) - C_n(\hat{\boldsymbol{v}}_2)|^2 = |H_n|^2 \cdot \epsilon \quad , \tag{4.130}$$

where ϵ is just the size of the per-dimensional squared difference between closest codewords. These terms have greater impact in the ML decoding, whci sum over all (large and small H_n) terms.

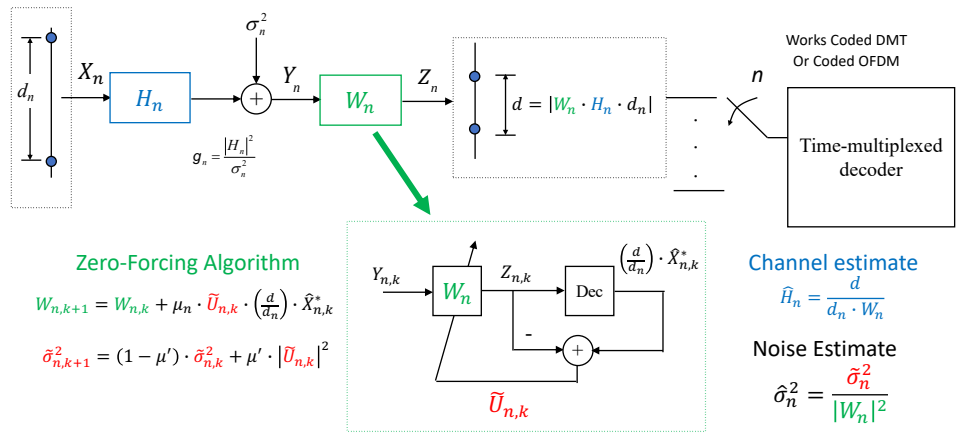


Figure 4.16: The Multichannel Normalizer for Coded MT.

Normalizer Use: Figure 4.16 revisits the multichannel normalizer for Coded MT. As also per Subsection 2.3.6, for a single independent channel dimension (complex or real), the MMSE and ZF ($W_n = H_n^{-1}$ shown in Figure 4.16) are the same with bias removal, so the figure shows ZF. The receiver's normalizer imposes a constant d in all dimensions, which corresponds to the distance in the dimensionally constant transmitter constellation $|C_n| = |C|$. The normalizer output error component $\tilde{U}_n$ reduces relatively for large $|H_n|$ and increases for small $|H_n|$; this error component determines the relative size of contributions within the ML sum. In effect, the ML receiver with the normalizer prior to the (otherwise unscaled decoder) does the loading instead of the transmitter, and subsequently the receiver finishes with the usual decoder for the known good code.

[24] Presuming noise whitening occurs individually on each dimension and absorbed into H_n.

Theorem 4.4.1 **[Separation of Coding and Modulation]** *Given a set of independent partitioned AWGN-channel dimensions with energies/dimension $\bar{\mathcal{E}}_n$ and gains g_n with equivalent*

$$SNR_{geo} = \left[\prod_{n=1}^{N} (1 + \bar{\mathcal{E}}_n \cdot g_n) \right]^{1/N} - 1 \quad ,$$

N repeated uses of a single good code's subsymbols with $\Gamma \to 0$ dB and

$$\bar{b} \leq \overline{\mathcal{I}} = \frac{1}{2} \cdot \log_2(1 + SNR_{geo})$$

and corresponding constant constellation $|C| = 2^{\bar{b}+\bar{\rho}}$ achieves the same performance as using N instances of a same-Γ good code's subsymbols with $\Gamma \to 0$ dB each with variable constellation $|C_n|$ and bits per tone

$$\bar{b}_n \leq \overline{\mathcal{I}}_n = \frac{1}{2} \cdot \log_2(1 + \mathcal{E}_n \cdot g_n) \quad .$$

Proof: *See the 3 preceding paragraphs.* **QED.**

For example, good "Coded-MT designs" that use a constant constellation size $|C| = 2^{\bar{b}_{max}}$ that is sufficiently large and a code rate $0 \leq r < 1$ such that

$$r \cdot |C| \leq \overline{\mathcal{I}} \quad , \tag{4.131}$$

reliably achieve the same performance as MT systems with variable bit loading and constellations. If the energy distribution is waterfilling, then $\overline{\mathcal{I}} \to \overline{\mathcal{C}}$.

Significance of Theorem 4.4.1: The constant constellation and single-code system requires less feedback of loading information to the transmitter, particularly if some fixed known energy distribution is à priori agreed. Essentially only the constellation size and some index to a particular code choice from some set of available codes returns through the feedback channel instead of MT's full set of bits/dimension quantities. The choice of $[r, |C|]$ has the name **Modulation Coding Scheme (MCS)**.

This section proceeds by investigating the decoder and loading problem that reduces essentially to choice of a code rate r and a constellation size $|C|$ that is consistently used on all (energized) dimensions and produces the desired average data rate. This problem is first addressed for a deterministic channel in Subection 4.4.1 and then for a statistically characterized channel in Subsection 4.4.2.

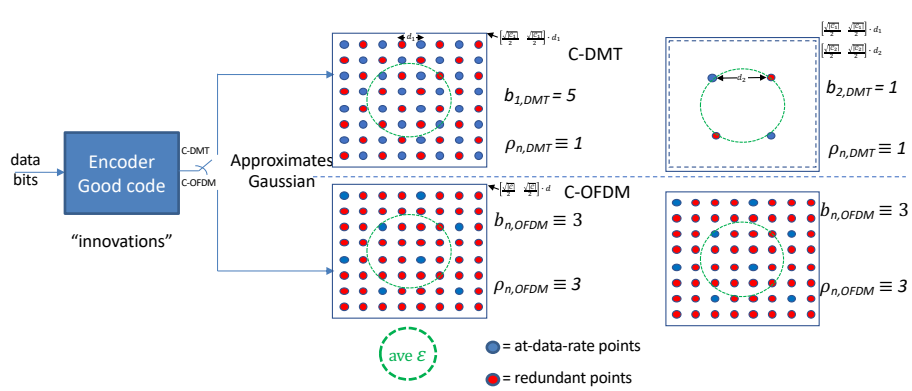


Figure 4.17: Illustration of equivalent constellations for Example 4.4.1.

EXAMPLE 4.4.1 *[Illustration of equivalent constellations for geometric SNR]* Figure 4.17 illustrates two different constellations. Section 4.7 further details the labels DMT and C-OFDM that here correspond to variable and fixed constellations. Both systems have the same overall data rate and geometric SNR. With very good codes, both will have the same performance on any channel that uses these two complex dimensions with the same (non-zero) energy on each dimension.

4.4.1 Group Loading with Coded MT

A unbiased geometric SNR, $\text{SNR}_{geo,u}$, characterizes all forms of loading and all energy distributions as per (4.24). An overall equivalent channel gain has definition:

Definition 4.4.1 [Geometric Channel Gain] *The* **geometric channel gain** *is*

$$g_{geo} \triangleq \frac{SNR_{geo,u}}{\bar{\mathcal{E}}_{\boldsymbol{x}}} \quad . \tag{4.132}$$

Theorem 4.4.1 relates that the aggregate group-loaded AWGN channel is equivalent to successive dimensional uses of a single AWGN channel, all with the same gain g_{geo}, and with the appropriate decoder weightings necessary for ML decoding. Use of Chapter 2's binary codes with $r < 1$ leads to a (soft[25]-decoded) free distance that will be a function of this r, call it $d_{free}(r)$. Coded MT loading presumes a known relationship between r and d_{free} that has monotonically non-decreasing inverse so that lower r means at least as large a value of $d_{free}(r)$. These values can be tabulated and stored for a good code (or encoder) choice. Usually higher r values occur when bits are "punctured" from output codewords (often some of the parity bits when the code is systematic, but not necessarily), which can cause a decrease in the value of d_{free}. When $r \to 1$, then $d_{free} = 1$. When $r \to 1$, the constellation shrinks and will be less likely to approach Gaussian. Consequently, Theorem 4.4.1 no longer holds and performance reduces, and the dimensions will become independent (both average and individual-code senses). Thus, only reasonably good codes with $r < 1$ will profit from the ML detector via Theorem 4.4.1. With lower performing codes, it is advisable to load each dimensions bits and constellation size individually.

With good codes at the reduced rate of $r < 1$, the probability of individual dimensional errors is

$$\overline{P}_e = \underbrace{2}_{\tilde{N}_e} \cdot Q\left[\sqrt{\frac{3 \cdot \text{SNR}_{geo,u} \cdot d_{free}(r)}{|C|^2 - 1}} \quad .\right] \tag{4.133}$$

Loading algorithms then compute Equation (4.133) for each $[|C| \; r]$ pair. The choice then has largest corresponding product $r \cdot \log_2 |C|$ that meets the target error probability. Clearly if the energy distribution is waterfilling, then all the possible computed values are as large as possible for the given amount of energy. More generally, the factor of 2 for nearest neighbors per dimension can be inserted for the code, which for good codes often exceeds 2 even if the constellation used is a square QAM constellation or per real dimension. When the system uses SQ QAM constellations and $\tilde{N} = 2$, then all applies with $|C|^2 \to |C|$. Similarly, the factor $\frac{3}{|C|^2-1}$ inside the Q-function argument generalizes for more sophisticated (closer to circle, sphere, or hypershpere) to κ if non-square constellations are used.

4.4.1.1 Coded-MT Loading

The Coded-MT loading problem for a fixed channel with $\text{SNR}_{geo,u}$ is the solution of

[25] soft decoding means the squared distance is used between channel output and potential filtered input sequence.

Definition 4.4.2 [Rate-Adaptive Coded-MT Loading] Rate-adaptive Coded-MT loading *with* $\widetilde{N} = 2$ *solves*

$$\textbf{objective:} \quad \max_{r,|C|} \quad r \cdot \log_2 |C| \tag{4.134}$$

$$\tag{4.135}$$

$$\textbf{subject to:} \quad \bar{P}_e = \quad \bar{N}_e \cdot Q\left[\sqrt{\frac{3 \cdot SNR_{geo,u} \cdot d_{free}(r)}{|C|-1}}\right] , \tag{4.136}$$

where $0 < r \leq 1$ *is selected from among the allowed codes, and where* $|C|$ *is the number of subsymbol values in the square QAM constellation used on all tones. The fraction* $\frac{3}{|C|-1}$ *can be adjusted to* κ *if non-square constellations are used, but the concept is the same; but there may be a larger number of* $|C|$ *choices.*

Coded MT loading's optimization can exhaustively search all combinations. (Recursions are possible, but typically the calculations are not particularly complex.) Typically the allowed constellations are BPSK, 4QAM, 16QAM, 64QAM, 256QAM$\cdots$ for a complex baseband-equivalent channel, and the square constellations could be viewed as 2PAM, 4PAM, 8 PAM, $\cdots$ in one dimension. The gap has essentially been replaced by d_{free} in this C-MT loading, which assumes that free distance is evenly distributed across all dimensions (which is not quite true, but roughly true especially for well-designed codes with $\Gamma \to 0$ dB).

EXAMPLE 4.4.2 [loading with a 64-state punctured rate 1/2 code] A nominally rate $r = 1/2$ convolutional code accepts inputs in 6-bit blocks and produces consequent 12-bit output blocks. The free distance of the code is $d_{free}(r = 1/2) = 10$. The encoder output's 12 bits can be punctured by discarding 3 bits to leave 9 output bits and a rate 2/3 code where $d_{free}(r = 6/9 = 2/3) = 7$; or by puncturing 4 bits to leave only 8 output bits and a rate $r = 6/8 = 3/4$ code with $d_{free}(r = 2/3) = 6$. This example channel's geometric SNR for a group of 48 tones is 14.5 dB. The possible constellations are 4-QAM, 16-QAM, and 64-QAM. The target bit error rate is 10^{-7}.

Clearly, $r = 1$ and $d_{free} = 1$ would allow 4-QAM to be transmitted at $\bar{P}_e = 10^{-7}$ if all the subchannels had the same gains. However, when $r = 1$ the code is not good, and the performance would be poor – that is the Separation Theorem does not apply. The ML/MMSE fails and large noise on a few dimensions dominates performance with this poor MT design. The data rate would be 1 bit/dimension. Instead, a rate 3/4 code would enable 16-QAM to be transmitted with data rate $3/4 \cdot 2 = 1.5$ bits/dimension (the code has enough gain at roughly 6dB to offset the constellation distance reduction). The rate 2/3 code could also be used with 16-QAM and would achieve $2/3 \cdot 2 = 1.33$ bits/dim. The distance gain of 10 for $r = 1/2$ is not sufficient to allow 64-QAM to be transmitted (which would have only at best also produced the 1.5 bits/dim anyway), so the best Coded-MT loading choice of $r = 3/4$ and C=16-QAM leads to 1.5 bits/dimension. See Problem 4.13 for more on this example.

Example 4.4.2 illustrates C-MT's somewhat gross granularity, which is used in IEEE 802.11 Wi-Fi systems (the most basic of which do indeed use 48 tones and SNR_{geo}-based modulation-coding-scheme loading). Clearly the more finely tuned adaptive bit-loading of Levin-Campello would lead to somewhat higher data rate, but LC needs to know all the subchannels' g_n values (and accurately). Often, C-MT loading uses some subchannels that probably would have otherwise been zeroed by waterfilling. The restriction to a fixed set of used channels simplifies system implementation, but thus likely reduces performance. Example 4.4.2's process limits feedback to the two simple parameters[26] $[|C|,\ r]$. Extension to an MIMO system with $\widetilde{N} = 2 \cdot L_x$ follows the same that expands to include the $L_y \times L_x$ matrix H_n's different singular values on each of its constituent spatial dimensions.

[26] Wi-Fi achieves this by selecting one of 128 (7 bits of feedback) possible modulation-coding-scheme choices.

4.4.2 Coded-MT Loading for Statistically Characterized Channels

Wireless systems often model the geometric SNR or equivalently a single representative average value of g as per Equation (4.132) for all tones by a probability distribution, p_g where $\sum_{g\in\mathcal{G}} p_g = 1$ where $\mathcal{G}$ is the discrete set of allowed g values (as in Section 1.6). $\mathcal{E}_{x,g}$ will denote the energy transmitted for a specific known g value. For the upcoming ergodic loading, the gains' (g_n) probability distribution is the same for all dimensions (or effectively across the dimensions). A receiver can model or measure this distribution.

With a stationary probability distribution (or more generally a p_g that is the same on all dimensions), it is possible to implement a waterfilling or optimized input spectra as a function of g that is presumably measured and known to both receiver and transmitter (sometimes called **channel-state information or CSI**). At first glance, this problem may appear to be hopelessly non-causal, especially with time dimensions. Nonetheless, it is possible to implement an **"ergodic" waterfilling**:

4.4.2.1 Margin-Adaptive or Energy-Minimizing Ergodic Waterfilling

MA adaptive loading is more pertinent to systems where the constellations and thus bit rates are often held constant. For MA waterfilling, an ergodic waterfilling constant derives from an average bit rate $< b >$ over some used optimum subdomain of $\mathcal{G}$ that is called $\mathcal{G}^*$, so $g \in \{\mathcal{G}^*\}$:

$$< b > = \sum_{g\in\mathcal{G}^*} p_g \cdot \log_2\left(1 + \frac{\mathcal{E}_{x,g}\cdot g}{\Gamma}\right) \tag{4.137}$$

$$= \log_2 \prod_{g\in\mathcal{G}^*}\left(1 + \frac{\mathcal{E}_{x,g}\cdot g}{\Gamma}\right)^{p_g} \tag{4.138}$$

$$= \log_2 \prod_{g\in\mathcal{G}^*}\left(\frac{K_{ma}\cdot g}{\Gamma}\right)^{p_g} \tag{4.139}$$

$$2^{<b>} = \left(\frac{K_{ma}}{\Gamma}\right)^{\sum_{g\in\mathcal{G}^*} p_g} \cdot \prod_{g\in\mathcal{G}^*} g^{p_g} \tag{4.140}$$

Thus

$$K_{ma} = \Gamma \cdot \left(\frac{2^{<b>}}{\prod_{g\in\mathcal{G}^*} g^{p_g}}\right)^{\frac{1}{\sum_{g\in\mathcal{G}^*} p_g}} \tag{4.141}$$

$$= \Gamma \cdot \frac{\left(2^{<b>}\right)^{\left[\frac{1}{\sum_{g\in\mathcal{G}^*} p_g}\right]}}{\gamma^*_{geo}} \quad . \tag{4.142}$$

Definition 4.4.3 [Average Geometric Channel ratio] *The* **average geometric channel gain** *is*

$$\gamma^*_{geo} \triangleq \prod_{g\in\mathcal{G}^*} (g)^{\left[\frac{p_g}{\sum_{g\in\mathcal{G}^*} p_g}\right]} \quad . \tag{4.143}$$

The quantity g_{geo} in (4.132) is slightly different than γ_{geo} in (4.143), but the two converge together as the used subchannels' SNRs all significantly exceed unity[27] . Ergodic waterfilling algorithm proceeds by sorting the g values from largest to smallest (if not already naturally sorted) and then computing an energy value using all $g \in \mathcal{G}$ for K_{ma}. These delete the smallest for a negative-energy result along with

[27] This is because $1 + \frac{SNR}{\Gamma} \to \frac{SNR}{\Gamma}$ as $SNR \to \infty$.

the corresponding g value for which the energy is negative. Then waterfilling continues by recursively recomputing K_{ma}. The remaining final largest set of all positive-energy values will be denoted $\mathcal{G}^*$. Thus,

$$\mathcal{E}_{x,g} = \begin{cases} K_{ma} - \frac{\Gamma}{g} & \gamma^*_{geo} > \frac{\Gamma}{K_{ma}} \\ 0 & \gamma^*_{geo} \leq \frac{\Gamma}{K_{ma}} \end{cases} \quad . \tag{4.144}$$

The remaining terms' minimized energy is

$$\mathcal{E}_{\boldsymbol{x}} = \sum_{g \in \mathcal{G}^*} p_g \cdot \left(K_{ma} - \frac{\Gamma}{g} \right) \quad . \tag{4.145}$$

Energy increase by (4.145)'s ratio to the allowed $\mathcal{E}_{\boldsymbol{x}}$ limit maximizes margin (or the energy can instead stay at minimum level to conserve energy). The solution basically relates that for any time period where the current measured g value (communicated to the transmitter as CSI) is less than Γ/K_{ma}, then no transmission occurs (so then the transmitter sends zero energy). This is nearly causal if channel change is relatively slow and thus can be implemented, often called a **quasi-static channel**. When $g > \Gamma/K_{ma}$ transmission occurs with $< b >$ and energy $K_{ma} - \Gamma/g$, possibly scaled to the allowed maximum energy. An infinite-dimensional form of this loading algorithm was found by Andrea Goldsmith[28] [16].

The MA case requires no "Levin-Campello discrete-loading" because presumably the value for $< b >$ corresponds to a constellation size that already is an integer. The transmit energy varies however to ensure the code gap is met. Ergodic waterfill still requires knowledge of the value of γ^*_{geo} at the transmitter to either silence (unused dimensions from waterfill) or to adjust the energies. However, a feedback improvement over MT occurs because only the γ^*_{geo} value need return to the transmitter. These could be for all dimensions individually or better instead be g_{geo} values with corresponding group $p_{g_{geo}}$ used to compute K_{ma}.

4.4.2.2 Rate Adaptive Ergodic Waterfilling

Similarly, Rate Adaptive Ergodic waterfilling has the solution

$$\mathcal{E}_{x,g} = K_{ra} - \frac{\Gamma}{g} \quad , \tag{4.146}$$

where

$$\mathcal{E}_{\boldsymbol{x}} = \sum_{g \in \mathcal{G}^*} p_g \cdot \left(K_{ra} - \frac{\Gamma}{g} \right) \tag{4.147}$$

$$= K_{ra} \cdot \sum_{g \in \mathcal{G}^*} p_g - \sum_{g \in \mathcal{G}^*} p_g \cdot \frac{\Gamma}{g} \tag{4.148}$$

Thus,

$$K_{ra} = \frac{\mathcal{E}_{\boldsymbol{x}} + \Gamma \cdot \sum_{g \in \mathcal{G}^*} \frac{p_g}{g}}{\sum_{g \in \mathcal{G}^*} p_g} \tag{4.149}$$

where the set $\mathcal{G}^*$ is the largest set in waterfilling that has only non-negative energies. The bit rate will be a function of the current value of g as

$$b(g) = \log_2 \left(\frac{K_{ra}}{\Gamma} g \right) \quad . \tag{4.150}$$

The average bit rate $< b >$ (over time) will be maximized. Again, no transmission occurs when $g \leq \Gamma/K_{ra}$. When $\Gamma = 0$, then $< b >$ is the averaged-over-g mutual information $\langle \mathcal{I} \rangle$.

[28] Prof Andrea Goldsmith (07/04-1964 -), now Dean of Engineering at Princeton University and well-known for her enthusiastic promotion of wireless communication.

Finite granularity in rate-adapted variable $< b >$ can be accommodated in the discrete probability distribution p_g. Basically the "binning" (See Subsection 4.4.2.5) of g values to construct the p_g distribution should use the lowest value in the g range corresponding to the bin and otherwise separates ranges by sufficient energies that each allow an additional unit of information at the desired P_e. Incremental energy comparison has less meaning in ergodic loading because there is only one energy that can change at any given time in power control, namely the current transmitted energy. The solution needs to use the largest constellation (or really data rate when a code is involved) less than the RA solution. Prob 4.14 addresses both MA and RA ergodic loading. For RA Ergodic waterfilling, again the transmitter sends no energy when the reported γ^*_{geo} corresponds to the zero energy level.

4.4.2.3 Aversion of gain-feedback with a statistically modeled channel

In many systems, the knowledge of time-varying g values at the transmitter may be difficult, and the transmitter always instead energizes a pre-agreed dimension set. There is no waterfilling (ergodic or otherwise). Instead, the quasi-static design fixes the average error probability as

$$\langle \bar{P}_e \rangle = \sum_{g \in \mathcal{G}} p_g \cdot \underbrace{\bar{N}_e \cdot Q\left[\sqrt{\frac{3 \cdot \bar{\mathcal{E}}_{\boldsymbol{x}} \cdot g \cdot d_{free}(r)}{|C| - 1}}\right]}_{\bar{P}_e(g)} \quad , \tag{4.151}$$

where a code-choice general characterization might have monotonic $\bar{P}_e(g)$ in g as well as any code parameters. When $p_g = \delta(g)$, then the expression reduces to the usual AWGN $\bar{P}_e$ expression. The average error probability further decomposes into a random-error probability and an outage probability according to the following:

Definition 4.4.4 [AWGN Outage Probability] *The* **outage probability** *differs from the* **random-error probability** *according to (they differ in the sum's value range)*

$$\langle \bar{P}_e \rangle = \sum_{g > g_{out}} p_g \cdot \underbrace{\bar{N}_e \cdot Q\left[\sqrt{\frac{3 \cdot \bar{\mathcal{E}}_{\boldsymbol{x}} \cdot g \cdot d_{free}(r)}{|C| - 1}}\right]}_{\bar{P}_e(g)} \tag{4.152}$$

$$\bar{P}_{out} = \sum_{g \leq g_{out}} p_g \quad , \textit{ respectively,} \tag{4.153}$$

where g_{out} is a threshold channel SNR to be determined so that (4.152) holds, while the "outage" corresponding to lower gains (meaning very poor performance with high error probability) must be accommodated by the receiver's erasure marking, or more generally through à prior soft information, in decoding[29]*. The fraction $\frac{3}{|C|-1}$ can be adjusted to κ if the design uses non-square constellations, but the concept is the same.*

In practice with good codes and coded-MT and Figure 4.16's **erasure marking**, the choice[30] of r must also satisfy

$$r \leq 1 - P_{out} \quad , \tag{4.154}$$

which means that symbols in outage must be marked or erased as in Chapter 2. More generally, a monotonic increasing function $f(P_{out})$ can correspond to other codes' use. The expression

$$\overline{\mathcal{C}}_{out} = (1 - P_{out}) \langle \overline{\mathcal{C}} \rangle \tag{4.155}$$

is the outage capacity of the binary erasure channel as per Chapter 2 and thus represents the bound on what good codes can achieve with a certain outage probability. A $p = 1/2$ binary symmetric channel

[30] This would occur for instance with the MDS Reed Solomon Codes of Chapter **??** and 8.

has zero capacity, but a binary erasure channel has instead capacity $1/2 > 0$, even with $p = 1/2$. Erasures will occur with the code when the measured multichannel gain (which is a function of each tone's estimated noise energies) are too low (below the g_{out}), as Figure 4.18 generally depicts.

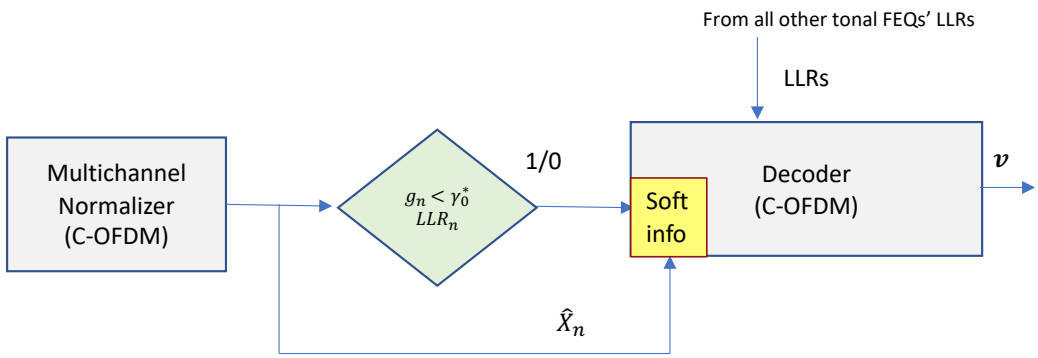


Figure 4.18: Coded-MT Multichannel Receiver's Normalizer and Erasures.

The corresponding receiver uses Figure 4.16's altered Coded MT Multichannel Normalizer

Definition 4.4.5 [Ergodic Rate-Adaptive Coded MT Loading with constant energy] *Ergodic Rate-adaptive coded MT loading* **with constant energy** *solves*

$$\textbf{objective:} \quad \max_{r,|C|,g_{out}} \quad r \cdot \log_2 |C| \tag{4.156}$$

$$\tag{4.157}$$

$$\textbf{subject to:} \quad \langle \bar{P}_e \rangle = \sum_{g > g_{out}} p_g \cdot \underbrace{\bar{N}_e \cdot Q\left[\sqrt{\frac{3 \cdot \bar{\mathcal{E}}_{\boldsymbol{x}} \cdot g \cdot d_{free}(r)}{|C| - 1}}\right]}_{\bar{P}_e(g)} \leq \bar{P}_e \tag{4.158}$$

$$r \quad \leq f\left(1 - \sum_{g \leq g_{out}} p_g\right) , \tag{4.159}$$

where the algorithm selects the code rate $0 < r \leq 1$ from among the allowed code rates, and $|C|$ is the selected (usually square) QAM constellation size in (4.156) - (4.159). The outage threshold g_{out} characterizes the two sums that are computed for each candidate ordered pair of $[r, |C|]$. The fraction $\frac{3}{|C|-1}$ can be adjusted to κ with non-square constellations, but the concept is the same. Even more generally as in the underbrace, any monotonic function of error-probability $\bar{P}_e(g)$ in the first constraint 4.158 can be combine with a function $f(P_{out})$ to correspond to code choice.

Summarizing, this Coded-MT Loading approaches MT loading and thus optimum performance IF

1. N^* does not change much (in wireless, this means faded dimensions are small in number relative to N), so that the waterfill energy optimization offers only small improvement over using a constant equal-energy excited set.

2. A maximum likelihood detector processes all dimensions (and not just one for each dimension). This includes the decoder's weighting subsymbol LL_n terms appropriately.

3. The constellation size is sufficiently larger than the size needed for transmission but satisfies $r \cdot \log_2(|C|) < \langle \mathcal{I} \rangle$.

4. The code used is good (close to capacity achieving, so $\Gamma \to 0$ dB), which in practice means it distributes distance well over all dimensions.

5. Interleaving ensures that any correlated adjacent-dimension channel outages are distributed across many BICM codewords in decoding (see Chapter 8, sections 1 and 6).

However, gain feedback is then not necessary with the Coded-MT loading, just r and $|C|$ are returned to the transmitter. Only the receiver needs to know g_{out}.

EXAMPLE 4.4.3 [Wireless Loading] Wireless systems typically allow code rates of $r = 7/8, 5/6, 4/5, 2/3, 1/2, 1/3, 1/4$, which this example extends to rates .9 and 1/5 with corresponding free distances:

```
>> r =    [  0.9    0.8    0.75    0.67    0.5    0.25    0.2];
>> dfree = [ 2       4       6       7      10     20     25];
```

For this channel example, the nearest neighbor count has been fixed to 2 per dimension. A given wireless channel has various SNR's depending on other traffic from other (overlapping spectra) wireless systems. The observed gain distribution is:

```
>> g = [ 3   30   300  600   1200   2400   4800    10000];
>> pg=[.11   .1   .03  .05   .35     .2     .1      .06];
```

The loading algorithm selects the optimum values for g_{out}, r, and constellation size from 4QAM, 16QAM, 64QAM, 256QAM, and 1024QAM. The following matlab commands first create a table of Q-function arguments for evaluation of different d_{free} and g values from the discrete distribution, and then also evaluates the probability distribution in a table of code choices (vertical) versus possible g_{out} (horizontal) selections as the g values. $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$ in this example. The $d_{free} \cdot g$ values for the numerators in (4.158) and (4.159) are

```
>> SNR=kron(dfree',g) =
  6        60        600       1200       2400       4800       9600      20000
12         120       1200      2400       4800       9600      19200      40000
15         150       1500      3000       6000      12000      24000      50000
18         180       1800      3600       7200      14400      28800      60000
30         300       3000      6000      12000      24000      48000     100000
60         600       6000     12000      24000      48000      96000     200000
75         750       7500      5000      30000      60000     120000     250000

>> 10*log10(SNR) =  (in dB)
    7.7815   17.7815   27.7815   30.7918   33.8021   36.8124   39.8227   43.0103
   10.7918   20.7918   30.7918   33.8021   36.8124   39.8227   42.8330   46.0206
   11.7609   21.7609   31.7609   34.7712   37.7815   40.7918   43.8021   46.9897
   12.5527   22.5527   32.5527   35.5630   38.5733   41.5836   44.5939   47.7815
   14.7712   24.7712   34.7712   37.7815   40.7918   43.8021   46.8124   50.0000
   17.7815   27.7815   37.7815   40.7918   43.8021   46.8124   49.8227   53.0103
   18.7506   28.7506   38.7506   41.7609   44.7712   47.7815   50.7918   53.9794
```

The matrix below called "prob2" simply repeats the probability distribution in all rows to facilitate later calculations. Thus,"Pout" is the outage probability for the given gain distribution"pg" and sums from the left across a row's g values below/equal to the possible g_{out} possibilities. That is done for row 1 below, but is the same for all rows. Analysis compares $1 - P_{out}$ to $f(r) = r$ to find when $r < 1 - P_{out}$.

```
>> prob2=kron(ones(7,1),pg) =
    0.1100    0.1000    0.0300    0.0500    0.3500    0.2000    0.1000    0.0600
    0.1100    0.1000    0.0300    0.0500    0.3500    0.2000    0.1000    0.0600
    0.1100    0.1000    0.0300    0.0500    0.3500    0.2000    0.1000    0.0600
    0.1100    0.1000    0.0300    0.0500    0.3500    0.2000    0.1000    0.0600
    0.1100    0.1000    0.0300    0.0500    0.3500    0.2000    0.1000    0.0600
    0.1100    0.1000    0.0300    0.0500    0.3500    0.2000    0.1000    0.0600
    0.1100    0.1000    0.0300    0.0500    0.3500    0.2000    0.1000    0.0600
>> Pout=cumsum(prob2(1,1:8)) % =
    0.1100    0.2100    0.2400    0.2900    0.6400    0.8400    0.9400    1.0000
>> ones(1,8)-Pout =
    0.8900    0.7900    0.7600    0.7100    0.3600    0.1600    0.0600         0

>> r =
    0.9000    0.8000    0.7500    0.6700    0.5000    0.2500    0.2000
```

Only the 3rd and 4th entries for the possible P_{out} satisfy $r < 1 - P_{out}$ corresponding to rates $r = 3/4$ or $r = 2/3$ as the only viable choices. Further analysis looks at only those entries' $\langle P_e \rangle$ possibilities further. Correspondingly, analysis deletes rows 1,2, and 5:8 from further consideratio. The quantity below called "prob1" contains the values in the sum of probabilities comprising $\langle P_e \rangle$, which will be a function of constellation (qfunc is the Q-function):

4-QAM:

```
>> prob1=qfunc(sqrt(SNR(3:4,:)))
prob1 =   1.0e-04 *

    0.1105    0.0000         0         0         0         0         0         0
    0.0230    0.0000         0         0         0         0         0         0
>> avePe=cumsum((prob2(3:4,1:8).*prob1)','reverse')'
avePe =   1.0e-05 *
    0.1215    0.0000         0         0         0         0         0         0
    0.0253    0.0000         0         0         0         0         0         0
```

Each successive row of "avePe" corresponds to a smaller value of r and thus larger value of d_{free} as given, and there are only 2 such rows that remain. The cumulative sum (reversed) sums the values in prob1 times the p_g values that have index $g > g_{out}$. The preceding P_{out} analysis further suggests that values beginning at entries 4 (=3+1) and 5 = (4+1) are of interest because only those satisfy the outage requirements, and $\langle P_e \rangle$ sums only values $g > g_{out}$. Clearly the cut-off is $g_{out} = 300$ because column 4 meets the $\langle P_e \rangle < 10^{-6}$ requirement and corresponds to $r = 3/4$, and thus $3/4 \times 1 = 0.75$ bits/dimension is a possibility with 4QAM. The other cutoff of $g_{out} = 3000$ also ensures $\langle P_e \rangle < 10^{-6}$, but would have an unnecessarily high outage probability.

Varying the constellation to **16-QAM**:

```
>> prob1=qfunc((3/15)*sqrt(SNR(3:4,:))) % =
    0.1981    0.0036    0.0000    0.0000    0.0000    0.0000    0.0000         0
    0.1797    0.0019    0.0000    0.0000    0.0000    0.0000    0.0000         0

>> 100*avePe=cumsum((prob2(3:4,1:8).*prob1)','reverse')' % =
    2.2152    0.0365    0.0000    0.0000    0.0000    0.0000    0.0000         0
    1.9954    0.0188    0.0000    0.0000    0.0000    0.0000    0.0000         0
```

(The multiplication by 100 simply allows us to see with matlab's 4-digit reporting values that are less than 10^{-6}.) For 16QAM, the outage value is again $g_{out} = 300$ (column 2, row 2) and meet the $\langle P_e \rangle < 10^{-6}$ in all rows with best data rate of $r \cdot \bar{b} = 0.75(2) = 1.5$ bits/dimension.

64-QAM:

```
>> prob1=qfunc((3/63)*sqrt(SNR(3:4,:))) % =

    0.4199    0.2615    0.0217    0.0021    0.0000    0.0000    0.0000    0.0000
    0.4136    0.2451    0.0145    0.0010    0.0000    0.0000    0.0000    0.0000

>> avePe=100*cumsum((prob2(3:4,1:8).*prob1)','reverse')' % =
    7.3106    2.6912    0.0766    0.0116    0.0009    0.0000    0.0000    0.0000
    7.0496    2.4997    0.0489    0.0053    0.0002    0.0000    0.0000    0.0000
```

64QAM is very marginal. The 2×10^{-6} might be considered "close enough" in column 5, in which case the code rate selected would be $r = 2/3$ and the bits per dimension would be $\bar{b} = 2/3 \cdot 4 = 2..67$ and the highest data rate. Smaller r values cannot be chosen because of the outage constraint. Close enough means that slight improvements in other aspects of the jump from 16QAM to 64QAM might allow targets to be met.

Higher $\bar{b} > 3$ clearly will not meet both outage and average error probability constraints.

4.4.2.4 Outage Capacity

The **outage capacity** leverages outage probability. First:

Definition 4.4.6 [Average (Ergodic) Capacity] *The* **ergodic capacity** *is*

$$< \overline{\mathcal{C}} >= \sum_{g \in \mathcal{G}^*} p_g \cdot \frac{1}{2} \cdot \log_2 \left(1 + \bar{\mathcal{E}}_{x,g} \cdot g\right) \quad . \tag{4.160}$$

Second:

Definition 4.4.7 [Outage Capacity] *There is no outage (the outage is presumed to be an erasure) at rates such that*

$$(1 - P_{out}) \cdot < \mathcal{C} > \quad . \tag{4.161}$$

When designing with the outage approach, codes must introduce sufficient redundancy to recover the data lost during the outage as well as recover from random errors occurring when there is not an outage. If there were no outages (meaning most likely the channel is not time-variant), this code could achieve data rate $< \overline{\mathcal{C}} >$. Outage capacity represents the highest achievable data rate by Coded-MT with the associated code of rate r, when the outage-bits are marked by erasures or other iterative-decoding use of soft information (See Chapter 7.

Use of flat-energy distributions: From an implementation perspective, a use of a flat energy $\mathcal{E}_x/N^*$ over the same subchannels/dimensions as waterfilling may be convenient. This flat energy corresponds to power-spectral-density masks that would prevent large disparity among the used (non-zero energy) dimensions. Good loading algorithms like waterfilling or Levin-Campello for discrete bit information quanta will tend to have nearly equal energy on the used dimensions. One reason for the nearly flat (or on/off) optimum input-energy distribution is that many channels simply look flat in a passband and decay rapidly outside that passband, making waterfilling's bowl look more like a deep box. The amount of information in the steep band edges is relatively small anyway, especially if N is large. Thus the total number of bits per symbol (or effectively the average number of bits for group loading) is less sensitive to the exact energy in these trough boundaries. Further, on channels with highly variable channel passbands, waterfilling's N^* (or set of N^*'s, one for each used band) will vary itself to cause only the best frequency spectra to be used - water filling uses large g_n dimensions first and often [17]. When the cost of an additional bit loaded to these better dimensions becomes sufficiently large that water filling expands to use additional dimensions, the range of energies can be larger but still are limited by the total energy constraint. Thus, waterfilling naturally produces nearly flat energy over an optimum subset of N^* dimensions on most channels of practical interest.

The average number of bits/subsymbol (subsymbol = complex dimension or two real dimensions) for a waterfill system and a flat on/off spectrum, using the same dimensions, at capacity is

$$\delta c = \frac{1}{N^*} \cdot \left\{ \sum_{n=1}^{N^*} \log_2 \left(1 + \mathcal{E}_n \cdot g_n\right) - \sum_{n=1}^{N^*} \log_2 \left(1 + \frac{\mathcal{E}}{N^*} \cdot g_n\right) \right\} \tag{4.162}$$

$$\approx \frac{1}{N^*} \cdot \sum_{n=1}^{N^*} \log_2 \left[\frac{\mathcal{E}_n}{\mathcal{E}/N^*}\right] \tag{4.163}$$

$$= \log_2 \left[\frac{\mathcal{E}_{geo}}{\mathcal{E}_{ave}}\right] \quad . \tag{4.164}$$

The ratio inside the log in (4.164) is sometimes called the "spectral flatness" and also finds use in audio compression. In this transmission case, the fixed energy constraint (which is the average energy in the denominator) further limits the ratio δc. Large δc corresponds to nearly all energy loaded to one or a few dimensions, thus changin N^*. The consequent spectrum is flat again on the corresponding enlarged optimum band. The logarithm further compresses δc. While it is hard to put an exact bound on the difference, practical situations with large (4.164) are rare. This text's examples also have very small δc. Thus, little is lost by using flat power on dimensions that loading suggests should carry energy. This means with ergodic loading, essentially power control decides to transmit or not. For other loading, simple on/off decision on which frequency bands or spatial paths to use or not use is nearly optimal as well. This does NOT mean that flat energy everywhere is close to optimum, as was also true with Chapter 3's CDEF results. Flat energy on the best band is close to optimum, but flat energy everywhere is usually not close to optimum.

4.4.2.5 Binning Estimation of the Gain Distribution:

Figure 4.16's normalizer estimates the channel gains g_n. The probability distribution p_g then follows from the relative sizes of the noise estimates $\tilde{\sigma}^2_{n,k}$.

Gain-distribution granularity typically corresponds to 3 dB (factor of 2) increases. One approach searches first for the smallest g_n value g_{min}. Then via sorting the g_n values in increasing order, those with values $g_n < 2 \cdot g_{min}$ determine $\mathcal{G}_1 = \{g_n \mid g_n < g_{min}\}$. Those with values $2 \cdot g_{min} \leq g_n < 4 \cdot g_{min}$ form set $\mathcal{G}_2$, and so on until all sets have been found. Then p_g is approximated by p_i for the sets as

$$p_g \rightarrow p_i = \frac{|\mathcal{G}_i|}{|\mathcal{G}|} \quad , \tag{4.165}$$

where $|\mathcal{G}|$ is the number of measured g_n values. (4.165)'s distribution can be used to complete Coded-MT loading. Binning creates an error-probability histogram that estimates the discrete distribution.

The binning-histogram thresholds can be chosen through inversion of the nominal error-probability equation for a target P_e according to

$$\bar{P}_e = 2 \cdot Q\left[\sqrt{\frac{3 \cdot \bar{\mathcal{E}}_{\boldsymbol{x}} \cdot g \cdot d_{free}}{|C| - 1}}\right] \tag{4.166}$$

to

$$g = \left[Q^{-1}\left(\frac{\bar{P}_e}{2}\right)\right]^2 \cdot \left(\frac{|C| - 1}{3 \cdot \bar{\mathcal{E}}_{\boldsymbol{x}} \cdot d_{free}}\right) \quad . \tag{4.167}$$

(4.167)'s evaluation with the allowed constellation sizes $|C||$ determines the corresponding g thresholds that delineate the $\mathcal{G}_i$. These static g values are then the bins' indices (the lower and upper g values). If there are some overlaps for some code-rate choices in g values, then the choice with highest rate might be selected (or with lowest r if outage is the dominant concern). There will be a lowest bin (or bins) for which no code works. These bins' probabilities add to the estimated outage probability. The selected code rate and erasure mechanism should support recovery from this outage probability. If the code rate does not support this outage, the particular code choices are not sufficient to ensure reliable transmission on this channel. Problem 4.16 further investigates an example of binning.

4.4.3 Spatial Separation Theorem

The separation theorem applies to any set of parallel independent channels, so indeed also applies to spatial channels as might occur with Section 2.3.5's simple vector coding (which appears in more detail in Sections 4.6 and 4.7. Certain combinations of parallel channels in space and time also can exploit the separation theorem, and this appears more in Section 4.7.

4.5 Channel Partitioning

Channel Partitioning methods divide a transmission channel into a set of parallel, and ideally independent, subchannels to which the loading algorithms of Sections 4.3 and 4.4 can be applied. This section re-introduces MIMO explicitly with the channel allowed to be $L_y \times L_x$, in addition to the multiple dimensions that may arise from time-frequency decompositions such as in this chapter's earlier multi-tone examples where the frequency-dimension index is n. Subsection 4.5.1 provides basics and generalizes Chapter 3's $q(t)$ to a matrix $Q(t)$ and a Generalized Nyquist Criterion, while Subsection 4.5.2 develops optimum basis-function sets that satisfy this generalized criterion. Subsection 4.5.3 revisits Section 2.5's asymptotic results and relates them to periodic channels. Subsection 4.5.4 introduces the important guard periods (and their dual guard bands) that approximate asymptotic effects' fundamentals from Subsection 4.5.3.

4.5.1 Matrix Channel Characterization

MT assigns each dimension (possibly and usually complex) a frequency index $n = 0, ..., \overline{N} - 1$, and all subchannels are independent[31], 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 $\{h_c(t) * \varphi_n(t)\}$ remains orthogonal for any $h_c(t)$. Reference to Chapter 2 finds that the MT basis functions, when combined with continuous-time waterfilling, 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 $h_c(t)$ in each MT frequency band is not quite flat in general. Thus, some of Chapter 3's "equalization loss" can occur on each tone. Furthermore, the MT basis functions exist over an infinite time interval, whereas practical implementation restricts of attention to a finite observation interval. Thus, MT introduces multi-channel modulation, but is not practical for channel partitioning.

This subsection introduces optimum basis functions for modulation based on a finite time interval T. Subsections 4.5.1.1 and 4.5.1.2 show that the optimum basis functions become channel-dependent for SISO and MIMO systems respectively, leading to Subsection 4.5.1.3's Generalized Nyquist Theorem 4.5.1. These optimum basis functions are the eigenfunctions of a certain channel covariance function (a matrix function in the MIMO case).

4.5.1.1 The SISO $Q(t)$ Channel Characterization

Before generalizing to MIMO, a SISO ($L_y = L_x = 1$) channel's analysis provides insight with less notational burden. A one-shot **single-input-single-output (SISO)** multi-channel modulated signal $x(t)$ satisfies

$$x(t) = \sum_{n=1}^{\overline{N}} x_n \cdot \varphi_n(t) \quad , \tag{4.168}$$

while a time succession of such transmissions is

$$x(t) = \sum_k \sum_{n=1}^{\overline{N}} x_{n,k} \cdot \varphi_n(t - kT) \quad . \tag{4.169}$$

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

$$h_c(t) * x(t) = \sum_k \sum_{n=1}^{\overline{N}} x_{n,k} \cdot \varphi_n(t - kT) * h_c(t) = \sum_k \sum_{n=1}^{\overline{N}} x_{n,k} \cdot h_n(t - kT) \quad , \tag{4.170}$$

which has N pulse responses $h_n(t)$, $n = 1, ..., \overline{N}$. Following Section 3.1's development, the set of sampled outputs of the N matched filters $h^*(-t)$ at all symbol instants kT forms a sufficient statistic or complete

[31] Convenient analysis often views DC $n = 0$ and Nyquist $n = \overline{N}$ as one complex dimension for real baseband $h_c(t)$ cases; if the $h_c(t)$ is complex baseband then there are exactly $\overline{N}$ complex dimensional gains and the anomaly of the DC/Nyquist situation does not occur.

representation of the transmitted signal as shown in Figure 4.19. There are $\overline{N}^2$ channel-ISI-describing functions[32]

$$q_{n,m}(t) \triangleq \frac{h_n^*(-t) * h_m(t)}{\|h_m\| \cdot \|h_n\|} \tag{4.171}$$

that characterize an $\overline{N} \times \overline{N}$ square matrix $Q(t)$, given by

$$Q(t) = \begin{bmatrix} q_{\overline{N},\overline{N}}(t) & ...q_{\overline{N},1}(t) & \\ \vdots & \ddots & \vdots \\ q_{1,\overline{N}}(t) & ...q_{1,1}(t) & \end{bmatrix} \quad . \tag{4.172}$$

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 (MIMO) although not here as physically separated wires nor antennas as spatial dimensions. $Q(t)$ has sampled-time representation

$$Q_k = Q(kT) \quad . \tag{4.173}$$

The next subsection then generalizes this to essentially a "doubly MIMO" system, with $L_x \geq 1$ and $L_y \geq 1$.

4.5.1.2 The MIMO $Q(t)$ Channel Characterization

From Subsection 3.10 in Chapter 3, the transmission model generalizes to L_x dimensional inputs as

$$\boldsymbol{x}(t) = \sum_k \sum_{n=1}^{\overline{N}} \underbrace{\boldsymbol{\Phi}_n(t-kT)}_{L_x \times L_x} \cdot \underbrace{\boldsymbol{x}_k}_{L_x \times 1} \quad , \tag{4.174}$$

where the $L_x \times L_x$ basis-function matrix now has dimensional index n as $\boldsymbol{\Phi}_n(t)$. Chapter 3 has no such index n because the pulse response was one-dimensional (one real dimension for baseband PAM and one complex dimension for quadrature QAM). The MIMO channel $H_c(t)$ is $L_y \times L_x$. Thus,

$$H_c(t) * \boldsymbol{x}(t) = \sum_k \sum_{n=1}^{\overline{N}} \underbrace{H_c(t)}_{L_y \times L_x} * \underbrace{\boldsymbol{\Phi}_n(t-kT)}_{L_x \times L_x} \underbrace{\boldsymbol{x}_k}_{L_x \times 1} \tag{4.175}$$

$$= \sum_k \sum_{n=1}^{\overline{N}} \underbrace{\boldsymbol{H}_n(t-kT)}_{L_y \times L_x} \cdot \boldsymbol{x}_k \quad , \tag{4.176}$$

where $\boldsymbol{H}_n(t) \triangleq H_c(t) * \boldsymbol{\Phi}_n(t)$ is the nth $L_y \times L_x$ **pulse-response matrix**. Section 3.10.1 defines a norm squared for a pulse-response-matrix as an $L_x \times L_x$ diagonal matrix (with diagonal elements each equal to the squared norm of each of the corresponding L_x column-vector functions of $\boldsymbol{H}(t)$ (which are the sums of the individual squared norms of each vector functions' scalar element functions, each of those in turn integrals of the scalar-element function's magnitude squared over time or frequency). For the present $L_y \times L_x$ MIMO case, each $\|\boldsymbol{H}_n\|^2$ is now an $L_x \times L_x$ diagonal matrix and there are $\overline{N}$ of these that can be stacked into a larger $\|\boldsymbol{H}\|^2$ matrix of size $\overline{N}L_x \times \overline{N}L_x$. A normalized $L_x \times L_x$matrix of the convolution of the nth pulse response matrix with the mth pulse-response matrix is

$$Q_{n,m}(t) \triangleq \|\boldsymbol{H}_n\|^{-1} \cdot \boldsymbol{H}_n^*(-t) * \boldsymbol{H}_m(t) \cdot \|\boldsymbol{H}_m\|^{-1} \quad m,n = 1,...,\overline{N} \quad . \tag{4.177}$$

[32]The pulse response $h(t)$'s distinction from the channel impulse response $h_c(t)$ reappears here, as in Chapter 3 to emphasize the separation of channel transmit filter from the channel. In other parts of this text, and in general practice, the channel response absorbes the analog transmit filter $\varphi(t)$ into the channel so that the H then combines that response with the channel itself. Any true variable or optimum basis-function processing, then becomes part of digital signal processing in practice.

Aggregation of these $\overline{N}^2$ different $L_x \times L_x$ matrices produces

$$\boldsymbol{Q}(t) = \begin{bmatrix} Q_{\overline{N},\overline{N}}(t) & ...Q_{\overline{N},1}(t) & \\ \vdots & \ddots & \vdots \\ Q_{1,\overline{N}}(t) & ...Q_{1,1}(t) & \end{bmatrix} \quad . \tag{4.178}$$

Sampling this larger matrix at the symbol rate produces

$$\boldsymbol{Q}_k = \boldsymbol{Q}(kT) \quad . \tag{4.179}$$

When $L_x = 1$, then $\boldsymbol{Q}(t) = Q(t)$ of the SISO case.

4.5.1.3 Generalization of the Nyquist Criterion

Theorem 4.5.1 (Generalized Nyquist Criterion (GNC)) *A generalization of Chapter 3's Nyquist Criterion for no ISI (time-dimension index k) nor intra-symbol interference (frequency-dimension index n) nor spatial crosstalk (spatial-dimension index ℓ) in the MIMO case is*

$$\boldsymbol{Q}_k = I \cdot \delta_k \quad , \tag{4.180}$$

that is the sampled (doubly) MIMO channel matrix is the identity matrix at time 0, and zero elsewhere.

proof: Since $\boldsymbol{Q}_k = I \cdot \delta_k = 0$ 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. Since each of the block matrices on the diagonal of Equation (4.178) is an identity, there is also no crosstalk. The unit value follows from the normalization of each pulse-response-matrix basis function, $Q_{n,n}(0) = I$. **QED**

Chapter 3's earlier (SISO) Nyquist Theorem is a special case that occurs when $Q_k = q_k = \delta_k$ (and $\overline{N} = L_x = 1$). Thus, to avoid all inter-dimensional interference, basis-function designs satisfy (4.180). 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, Section 3.10's vector equalization as well as Chapter 5's Generalized Decision Feedback Equalizers act to minimize the inter-dimensional interference in an MMSE sense. This section addresses only with those that do satisfy the GNC or its equivalents.

4.5.2 Optimal Choice of Transmit Basis

This subsection investigates a theoretical method for describing best basis-function sets. Subsection 4.5.4 instead investigates the elimination of the interference between subchannels with the often-encountered use of "guard periods" or "guard intervals."

The SISO ($L_x = 1$) channel impulse response forms a **channel autocorrelation function** according to

$$r(t) = h_c(t) * h_c^*(-t) \quad , \tag{4.181}$$

which is presumed to be nonzero only over the finite interval $(-T/2, T/2)$. This channel autocorrelation function defines a possibly countably infinite-size set of orthonormal **eigenfunctions**, $\{\varphi_n(t)\}$, nonzero only over the time interval $(-T/2, T/2)$, that satisfy the relation[33]

$$\lambda_n \cdot \varphi_n(t) = \int_{-T/2}^{T/2} r(t-\tau) \cdot \varphi_n(\tau) \cdot d\tau \quad n = 1, ..., \infty \quad , \quad \forall t \in (-T/2, T/2) \; . \tag{4.182}$$

That is, the function $\varphi_n(t)$ that when convolved with the channel autocorrelation function over the interval $[-T/2, T/2)$ reproduces itself, scaled by a constant λ_n, called the **eigenvalue**. The set of

[33] 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.

eigenvalues is unique to the channel autocorrelation $r(t)$, while the eigenfunctions are not necessarily unique.[34] Each eigenfunction represents a channel transmission "mode" (a dimension) through which data passes independently of the other modes/dimensions with gain λ_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 eigenfunctions at the symbol rate lead to a Q_k that satisfies the generalized Nyquist criterion. The eigenvalue problem is very difficult to solve when $r(kT) \neq \delta_k$. Fortunately, 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. Section 4.6 (and also Subsection 4.5.4.2 briefly) discusses this concept, but Subsection 4.5.4 provides a simpler solution known as a guard period or cyclic extension that circumvents the difficulty with negligible loss.

Space-Time MIMO Generalization of Eigenfunctions: The scalar case's generalization to the $L_x \times L_x$ MIMO channel autocorrelation matrix, with attention again on the interval $(-T/2, T/2)$, provides

$$R(t) = H_c^*(-t) * H_c(t) \quad , \tag{4.183}$$

and which has **eigenfunction vector** decomposition defined by

$$\boldsymbol{\Phi}_n(t) \cdot \boldsymbol{\lambda}_n = \int_{-\frac{T}{2}}^{\frac{T}{2}} R(t-\tau) \cdot \boldsymbol{\Phi}_n(\tau) \cdot d\tau \;\; n = 1, ..., \infty \; \forall t \in (-T/2, T/2) \quad . \tag{4.184}$$

In this case $\boldsymbol{\lambda}_n$ will be an $L_x \times L_x$ (possibly complex) diagonal matrix of eigenvalues for the nth MIMO eigenmode with singular value decomposition $\boldsymbol{\lambda}_n = F_n \cdot \Lambda_n \cdot M_n^*$. When SISO ($L_x = L_y = 1$), $R(t)$ simplifies to $r(t)$.

Ordering and Optimality: There may be a countably infinite number of eigenvalues and corresponding eigenfunction matrices, so the best designs select the $\overline{N}$ largest (largest determinants in the MIMO case). 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 matrix functions $\boldsymbol{\Phi}_n(t)$ for $n = 1, 2, ...\overline{N}$. These functions would have a linear orthonormal representation (think Gram Schmidt) on the $\overline{N}$ largest eigenfunction matrices and a component on the remaining eigenfunctions (since the eigenfunctions can be the basis of a Hilbert space that represents all non-zero probability signals that have the given autocorrelation function). 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 eigenvalue's eigenfunctions. Thus, the largest set of $\overline{N}$, or any set in 1-to-1 correspondence with them, are the best to use for transmission.

Figure 4.19's **Modal Modulation (MM)** uses the channel-dependent orthonormal set of eigenfunctions (the $\overline{N}$ with largest eigenvalues) as a basis for modulation,

$$\boldsymbol{x}(t) = \sum_{n=1}^{\overline{N}} \boldsymbol{\Phi}_n(t) \cdot \boldsymbol{x}_n \quad . \tag{4.185}$$

The $L_x \times 1$ post-matched-filter-matrix receiver[35] signal $\boldsymbol{y}(t)$ in Figure 4.19 is then

$$\boldsymbol{y}(t) = \sum_{n=1}^{N} [R(t) * \boldsymbol{\Phi}_n(t)] \cdot \boldsymbol{x}_n + \tilde{\boldsymbol{n}}(t) \tag{4.186}$$

$$= \sum_{n=1}^{N} [\boldsymbol{\Phi}_n(t) \cdot \boldsymbol{\lambda}_n] \cdot \boldsymbol{x}_n + \tilde{\boldsymbol{n}}(t) \tag{4.187}$$

[34] 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 called "computing the eigenfunctions."

[35] The dimensions are **not** $L_y \times 1$ even if the channel output has $L_y \neq L_x$ because of the matched filter matrix.

where $\tilde{\boldsymbol{n}}(t)$ is the $L_x \times 1$ matched-filtered noise with autocorrelation function $\frac{\mathcal{N}_0}{2} \cdot R(t)$.

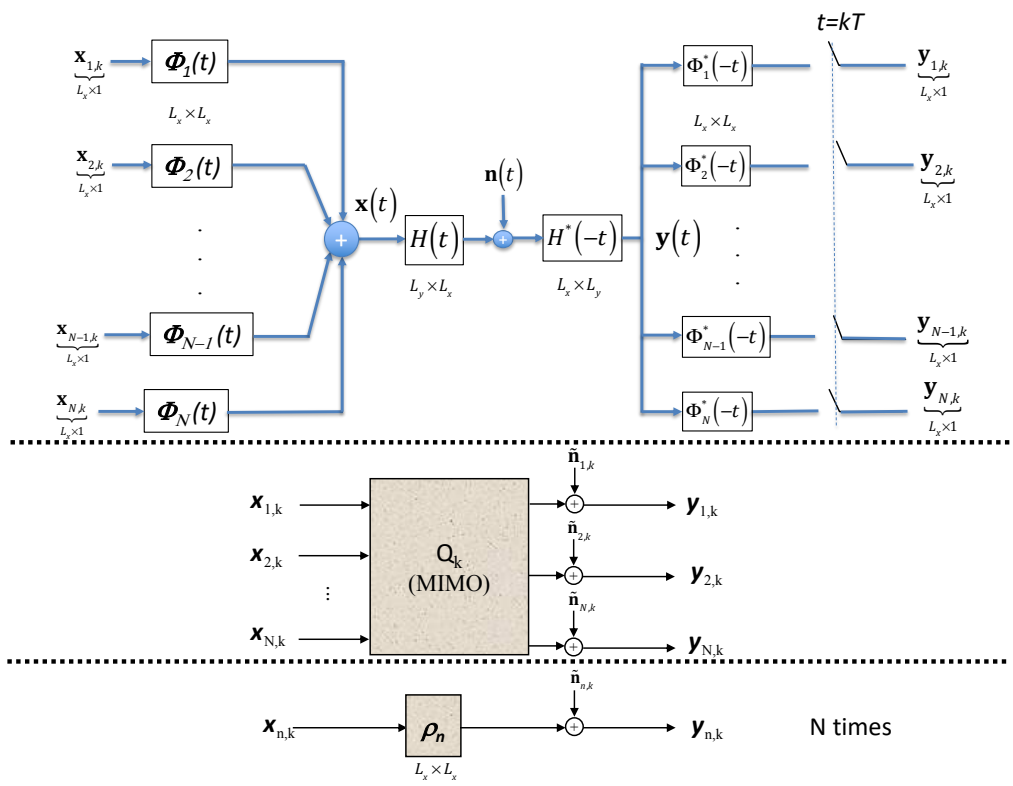


Figure 4.19: Modal Modulation: optimum channel partitioning.

There is no information loss in this signal because it is the output of the matched filters for each of the transmit bases. The matched-filter-matrix output vector the contains weighted channel input samples, $\boldsymbol{\lambda}_n \cdot x_n$. This summation has the same form as the modulated waveforms studied in Chapter 1. Each symbol element $\boldsymbol{x}_n$ inow generalizes to $\boldsymbol{\lambda}_n \cdot \boldsymbol{x}_n$.

The MAP/ML receiver is thus also the same as in Chapter 1 and recovers $\boldsymbol{\lambda}_n \cdot \boldsymbol{x}_n$ by processing each filter output by a matched-filter/sampler for each of the basis functions, as shown in Figure 4.180. These matched-filter matrices' noise-sample outputs are independent for different n because the basis functions are orthogonal, or equivalently,

$$\mathbb{E}[\tilde{\boldsymbol{n}}_i \tilde{\boldsymbol{n}}_j^*] = \frac{\mathcal{N}_0}{2} \int_{-T/2}^{T/2} \int_{-T/2}^{T/2} \boldsymbol{\Phi}_j^*(s) \cdot R(t-s) \cdot \boldsymbol{\Phi}_i(t) \cdot dt \cdot ds \tag{4.188}$$

$$= \frac{\mathcal{N}_0}{2} \int_{-T/2}^{T/2} \boldsymbol{\Phi}_i^*(s) \cdot \boldsymbol{\Phi}_j(t) \cdot \boldsymbol{\lambda}_i \cdot dt \tag{4.189}$$

$$= \frac{\mathcal{N}_0}{2} \delta_{ij} \cdot I \cdot \boldsymbol{\lambda}_i \quad . \tag{4.190}$$

Thus, the joint-probability density function $\mathrm{p}_{\boldsymbol{y}/\boldsymbol{x}}$ factors into $\overline{N} L_x$ 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 $L_x \times L_x$ matrix SNR's are

$$\mathrm{SNR}_n = |\boldsymbol{\lambda}_n| \cdot |R_{\boldsymbol{x}\boldsymbol{x},n}| \quad . \tag{4.191}$$

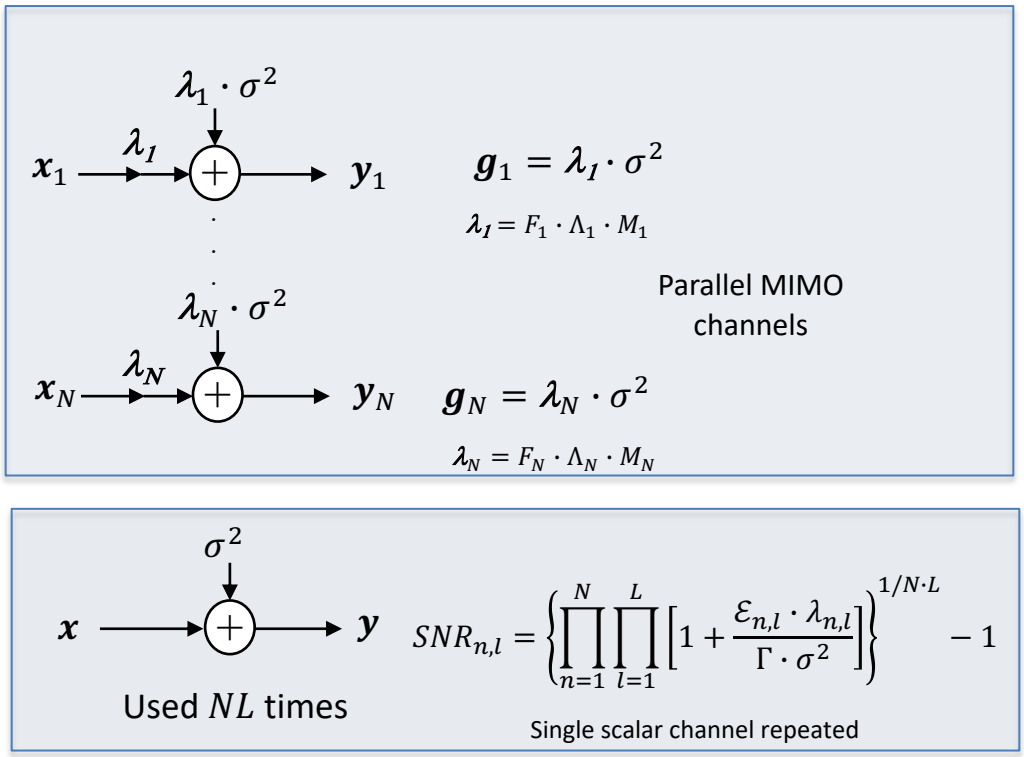


Figure 4.20: **Parallel Subchannels of Modal Modulation.**

The overall product SNR has $\overline{N}L_x$ scalar terms $\text{SNR}_{n,\ell}$ for $n = 1, ..., \overline{N}$ and $\ell = 1, ..., L_x$.

The functions $\mathbf{\Phi}_n(t)$ defined by modal modulation clearly satisfy the generalized Nyquist criterion because of the independence of the subchannels created (and any diagonal scaling removed without loss) and because they are zero outside the interval $[-T/2, T/2)$, thus averting intersymbol interference.

Theorem 4.5.2 **[Optimality of Modal Modulation]** *Given a linear additive Gaussian noise channel with ISI, modal modulation produces the largest $SNR_{m,u}$ of any $\overline{N}$-dimensional set of orthogonal basis functions that exist only on the interval $t \in (-T/2, T/2)$.*

Proof: The MM transmission system of Figure 4.19 can be optimally detected by observing each of the independent subchannel outputs individually. The optimal detector's 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 $\overline{N}$ largest eigenvalues are unique and MM selects the eigenfunctions corresponding to the $\overline{N}$ largest. A well-known property of the eigenvalues[36] is that no other set of N has a larger product. Then, defining $\boldsymbol{g}_n = \frac{2}{\mathcal{N}_0} \cdot \boldsymbol{\lambda}_n$ with $|\boldsymbol{g}_n| = \prod_{l=1}^{L_x} \boldsymbol{g}_{n,\ell} = \sigma^{-2 \cdot L_x} \cdot \prod_{l=1}^{L_x} \boldsymbol{\lambda}_{n,\ell}$ has

$$\prod_{n=1}^{N} |\boldsymbol{g}_n| \tag{4.192}$$

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

$$\prod_{l=1}^{L_x}\prod_{n=1}^{N}\left(1+\frac{\mathcal{E}_{n,\ell}\cdot \boldsymbol{g}_{n,\ell}}{\Gamma}\right) \tag{4.193}$$

is also a maximum because the function $1+\frac{\mathcal{E}_{n,\ell}\cdot \boldsymbol{g}_{n,\ell}}{\Gamma}$ is linear (convex) in g_n. Then since this function is maximized for any set of energies, it is certainly the maximum for the waterfilling set of energies that are known to maximize for any set of $\{\boldsymbol{g}_n\}$. 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 $(-T/2, T2)$. As $T\to\infty$ for fixed T_H, 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 $-(T/2, T/2)$. 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 $\overline{N}$ to go to infinity while also allowing the time interval to span $(-\infty,\infty)$:

Theorem 4.5.3 [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: (SISO case only): 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 $\varphi_n(t) = e^{-j\frac{2\pi n}{T}t}$ 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 $\omega = \frac{2\pi n}{T}$. As $\overline{N}\to\infty$, 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 4.5.2. The multitone system is thus optimum for infinite-length symbol period and an infinite number of tones. **QED.**

The extension to MIMO is relatively easy after understanding the vector coding and singular value decomposition as yet to come.

Theorem 4.5.3's implication is that for sufficiently large $\overline{N}$ and $T = \overline{N}T'$ with $\frac{1}{T'}$ being fixed, 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. The optimum detector is a simple set of threshold detectors in the uncoded case when $\Gamma = 0$ dB.

Similar to the scalar case, using the eigenfunction vectors $\boldsymbol{\Phi}_n(t)$, and all their L_xN constituent element functions $q_{n,i}(t)$, as the basis vectors is again Modal Modulation for the MIMO case. The product SNR maximization (for any given energy distribution with bounded total energy over all dimensions , but also for the overall maximizing waterfill energy distribution) is largest and this is an optimum transmission method with highest possible SNR for the MIMO case. The detectors on each of the individual subchannel outputs can be independent of one another and still create an ML detector.

If there is inter-**symbol** interference, then the MIMO method becomes optimum if the number of tones (on each and every dimension) goes to infinity, while the use of the eigenvalues (at each of these tone frequencies) across the spatial dimensions is also optimum for crosstalk or **intra**-symbol interference handling. Systems that today use this vector-DMT or vector-OFDM approach include ITU xDSL standards G.vector (G.993.5) and G.fast (G.9902) and wireless systems IEEE standards IEEE 802.11 (n), (ac), (ax), (d), and (ay) as well as 3GPP LTE Rev 14 and beyond (4G and 5G wireless).

4.5.3 Limiting Results (for time-frequency) - SISO case

This subsection elaborates on Section 2.5's asymptotic capacity analysis for MT. MM's used spectrum thus converges also to MT's used-tone set as $\overline{N} \to \infty$. 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 used bandwidth

$$|\Omega| = \int_{\Omega} d\omega \quad . \tag{4.194}$$

An optimum bandwidth Ω^* then corresponds to waterfilling's used subchannels $\overline{N} \to \infty$. If MT uses $\overline{N}^*$ such subchannels, then

$$\lim_{\overline{N}\to\infty} \frac{\overline{N}^*}{\overline{N}} \cdot \frac{2\pi}{T} = |\Omega^*| \quad . \tag{4.195}$$

The data rate is

$$\lim_{T\to\infty} \frac{b}{T} = \lim_{N\to\infty} \frac{b}{NT'} \quad . \tag{4.196}$$

Continuous frequency can then replace the frequency index n according to

$$\omega = 2\pi \cdot \lim_{N\to\infty} \frac{n}{NT'} \ , \quad n = -N/2, ..., N/2 \quad , \tag{4.197}$$

and the width of a tone becomes

$$d\omega = \lim_{N\to\infty} \frac{2\pi}{NT'} \tag{4.198}$$

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

$$S_x(\omega) = \lim_{N\to\infty} \bar{\mathcal{E}}_n \quad . \tag{4.199}$$

For infinite positive and negative 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

$$P_x = \lim_{N\to\infty} \frac{1}{N \cdot T'} \cdot \sum_{n=-N/2}^{N/2} \bar{\mathcal{E}}_n = \frac{1}{2\pi} \cdot \int_{\Omega} S_x(\omega) d\omega \quad . \tag{4.200}$$

The waterfilling equation then has continuous equivalent

$$\bar{\mathcal{E}}_n + \frac{\Gamma}{g_n} \to S_x(\omega) + \frac{\Gamma}{g(\omega)} = \lambda \text{ (a constant)} \tag{4.201}$$

subject to the total power constraint in (4.200) , 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

$$R = \frac{1}{2\pi} \int_{\Omega^*} \frac{1}{2} \log_2 \left(1 + \frac{S_x(\omega) \cdot g(\omega)}{\Gamma}\right) d\omega \tag{4.202}$$

$$= \frac{1}{2\pi} \int_{\Omega^*} \frac{1}{2} \log_2 \left(\frac{\lambda \cdot g(\omega)}{\Gamma}\right) d\omega \tag{4.203}$$

These results do extend to MIMO, but require some notational effort that will be facilitated by the development of the next section (4.6).

EXAMPLE 4.5.1 *[1 + .9D channel]* The channel with impulse response $h_c(t) = \text{sinc}(t) + .9 \cdot \text{sinc}(t-1)$ has the same performance as the $1 + .9 \cdot D^{-1}$ channel studied throughout this book, if the transmit filter is $\frac{1}{\sqrt{T}} \cdot \text{sinc}(t/T)$. An system with optimized MT basis functions of infinite length (as $\overline{N} \to \infty$ with $T' = T/\overline{N}$) would have an optimum bandwidth

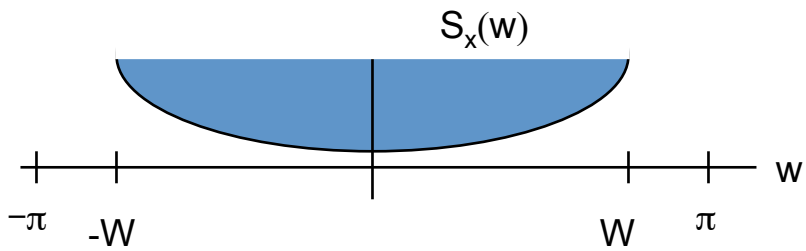


Figure 4.21: Capacity frequency interval for $1 + .9D^{-1}$ channel.

$\Omega^* = [-W, W]$ for some $W \leq \pi/T'$ as in Figure 4.21 Then, continuous waterfilling with $\Gamma = 1$ produces

$$P_x = \int_{-W}^{W} \left(\lambda' - \frac{.181}{1.81 + 1.8\cos(\omega)} \right) \frac{d\omega}{2\pi} \tag{4.204}$$

where W is implicitly in radians/second for this example. If $H_x = 1$ with $\frac{\mathcal{N}_0}{2} = .181$, the integral in (4.204) simplifies to

$$\pi = \int_{0}^{W} \left(\lambda' - \frac{.181}{1.81 + 1.8\cos(\omega)} \right) d\omega \tag{4.205}$$

$$= \lambda' W - .181 \left\{ \frac{2}{\sqrt{1.81^2 - 1.8^2}} \arctan \left[\frac{\sqrt{1.81^2 - 1.8^2}}{1.81 + 1.8} \tan\left(\frac{W}{2}\right) \right] \right\} \tag{4.206}$$

At the bandedge W,

$$\lambda' = \frac{.181}{1.81 + 1.8\cos(W)} \quad . \tag{4.207}$$

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

$$\pi = \frac{.181W}{1.81 + 1.8\cos(W)} - 1.9053 \arctan\left(.0526 \tan(W/2)\right) \tag{4.208}$$

$W = .88\pi$ approximately solves (4.208), and the corresponding value of λ is $\lambda = 1.33$.

The highest data rate with $1/T' = 1$ is then

$$\mathcal{C} = \frac{2}{2\pi} \int_{0}^{.88\pi} \frac{1}{2} \log_2 \left(\frac{1.33}{.181} (1.81 + 1.8\cos\omega) \right) d\omega \tag{4.209}$$

$$= \frac{1}{2\pi} \int_{0}^{.88\pi} \log_2 7.35 d\omega + \frac{1}{2\pi} \int_{0}^{.88\pi} \log_2 \left(1.81 + 1.8\cos\omega\right) d\omega \tag{4.210}$$

$$= 1.266 + .284 \tag{4.211}$$

$$\approx 1.55 \text{bits/second} \quad . \tag{4.212}$$

This exceeds the 1 bit/second transmitted on this channel in Chapter 3's examples where $T = T' = 1$. This chapter's later sections show that this channel's $\text{SNR}_{m,u}$ becomes 8.8 dB for large $\overline{N}$, .4 dB better than Chapter 3's 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 waterfilling with $\Gamma > 0$ dB, the designer needs to remember that the gap concept is only accurate when $\bar{b} \geq 1$. Often waterfilling 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 exact at all $\bar{b}$ when the gap is 0 dB, meaning capacity results are all exact.

4.5.3.1 Periodic Channels - SISO only

A channel with periodic $R(t) = R(t+T)$ will always have eigenfunctions $\mathbf{\Phi}_n(t) = e^{\jmath \frac{2\pi}{T} nt} \cdot I$ for the interval $[-T/2, T/2)$. This is easily evident from the Fourier Series for a periodic function on that interval, which is

$$R(t) = \sum_{n=-\infty}^{\infty} R_n \cdot e^{\jmath \frac{2\pi}{T} nt} \quad , \tag{4.213}$$

where

$$R_n = \frac{1}{T} \int_{-T/2}^{T/2} R(t) \cdot e^{-\jmath \frac{2\pi}{T} nt} dt \quad . \tag{4.214}$$

For instance, a useful Fourier series is that of the periodic impulse train

$$\sum_{m=-\infty}^{\infty} \delta(t - mT) = \frac{1}{T} \cdot \sum_{m=-\infty}^{\infty} e^{\jmath \frac{2\pi}{T} mt} \quad . \tag{4.215}$$

Substitution of the Fourier Series into the eigenfunction equation yields

$$\begin{aligned}
\mathbf{\Phi}_n(t) \cdot \boldsymbol{\lambda}_n &= \int_{-T/2}^{T/2} \sum_{m=-\infty}^{\infty} \mathbf{\Phi}_n(\tau) \cdot R_m \cdot e^{\jmath \frac{2\pi}{T} m(t-\tau)} d\tau && (4.216)\\
&= \int_{\tau=-T/2}^{T/2} \sum_{m=-\infty}^{\infty} \mathbf{\Phi}_n(\tau) \cdot \left[\frac{1}{T} \int_{u=-T/2}^{T/2} R(u) \cdot e^{-\jmath \frac{2\pi}{T} mu} du \right] \cdot e^{\jmath \frac{2\pi}{T} m(t-\tau)} d\tau && (4.217)\\
&= \int_{u=-T/2}^{T/2} \int_{\tau=-T/2}^{T/2} \left[\frac{1}{T} \cdot \sum_{m=-\infty}^{\infty} e^{-\jmath \frac{2\pi}{T} m(t-u-\tau)} \right] \cdot \mathbf{\Phi}_n(\tau) \cdot R(u) \cdot d\tau du && (4.218)\\
&= \int_{u=-T/2}^{T/2} \int_{\tau=-T/2}^{T/2} \left[\sum_{m=-\infty}^{\infty} \delta(t-u-mT) \right] \cdot \varphi_n(\tau) \cdot \mathbf{\Phi}_n(\tau) \cdot R(u) \cdot d\tau du && (4.219)\\
&= \int_{u=-T/2}^{T/2} \cdot \sum_{m=-\infty}^{\infty} \mathbf{\Phi}_n(t-u-mT) \cdot R(u) \cdot du && (4.220)
\end{aligned}$$

Continuing with $\mathbf{\Phi}_n(t) = e^{\jmath \frac{2\pi}{T} nt} \cdot I$ yields

$$\begin{aligned}
\boldsymbol{\lambda}_n \cdot e^{\jmath \frac{2\pi}{T} nt} &= \int_{-T/2}^{T/2} R(u) \cdot \sum_{m=-\infty}^{\infty} e^{\jmath 2\pi mn} \cdot e^{\jmath \frac{2\pi}{T} n(t-u)} du && (4.221)\\
&= \int_{-T/2}^{T/2} R(u) \cdot e^{\jmath \frac{2\pi}{T} n(t-u)} du && (4.222)\\
&= R_n \cdot e^{\jmath \frac{2\pi}{T} nt} \quad , && (4.223)
\end{aligned}$$

so $\mathbf{\Phi}_n(t) = e^{\jmath \frac{2\pi}{T} nt} \cdot I$ is an eigenfunction and the eigenvalue is $\boldsymbol{\lambda}_n = R_n$, the Fourier Series coefficient. 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 $-T/2 = -LT'$ to $T/2 = LT'$ to yield

$$\begin{bmatrix} \mathbf{\Phi}_n(-LT') \\ \mathbf{\Phi}_n(-LT'+T') \\ \vdots \\ \mathbf{\Phi}_n(LT') \end{bmatrix} \cdot \boldsymbol{\lambda}_n = T' \cdot \begin{bmatrix} R(0) & \dots & R(-2LT') \\ R(T') & \dots & R(-2LT'+T') \\ \vdots & \ddots & \vdots \\ R(2LT') & \dots & R(0) \end{bmatrix} \begin{bmatrix} \mathbf{\Phi}_n(-LT') \\ \mathbf{\Phi}_n(-LT'+T') \\ \vdots \\ \mathbf{\Phi}_n(LT') \end{bmatrix} \tag{4.224}$$

$$\mathbf{\Phi} \cdot \boldsymbol{\lambda} = R \cdot \boldsymbol{\varphi} \quad . \tag{4.225}$$

This equation is equivalent to finding the eigenvectors and eigenvalues of the expanded sampled matrix R, which for instance is accomplished for the designer with the matlab function "eig." The largest $\overline{N}$ eigenvalue magnitudes $|\lambda_n|$ select the corresponding eigenvectors as the sampled basis functions. These basis functions may be interpolated to create eigenfunction approximations. Clearly, the smaller the T' and thus larger L, the more accurate the approximation. In implementation, Section 4.6 illustrates a yet better way to approximate eigenfunctions in discrete time, which is known as vector coding.

4.5.4 Time-Domain Packets and Other Frequency-Domain Processing

This subsection re-investigates intersymbol interference and leads to Section 4.6's "guard-period" solution.

4.5.4.1 Overlap, Excess Bandwidth, and the Guard Period

A realistic transmission channel has most or all its nonzero impulse response within 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[37] T, as in Figure 4.22.

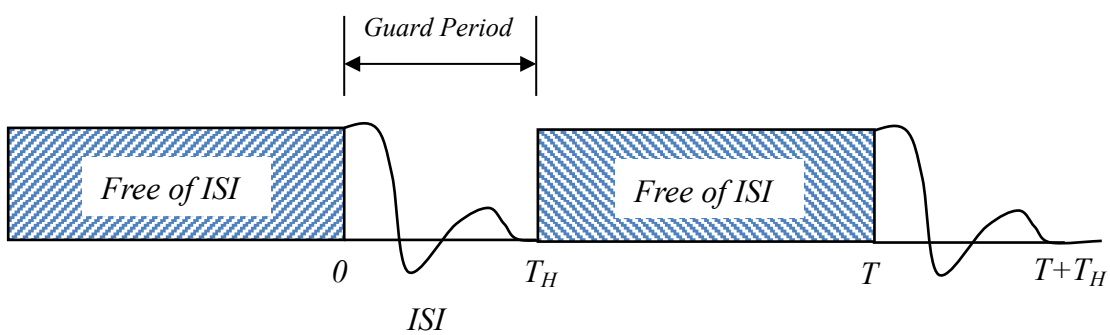


Figure 4.22: Illustration of guard period.

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 $(T - T_H, T)$. The receiver then ignores every symbol's first T_H seconds. If $T_H << T$, then this overhead penalty, or "excess bandwidth" penalty, is small. 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** or **guard interval** in multichannel modulation. For finite symbol period, the multichannel system then

[37] 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.

has an excess bandwidth (or equivalently excess dimensionality) given by the factor $\alpha = T_H/(T - T_H)$. There is no need for, nor equivalent of, a spatial guard period in space-time MIMO systems.

4.5.4.2 Filters with Multitone

SISO MT partitioning almost satisfies the Generalized Nyquist criterion with any impulse response $h_c(t)$, but requires an infinite-length basis function $\varphi(t) = \frac{1}{\sqrt{T}} \cdot \text{sinc}\left(\frac{t}{T}\right)$ and is not physically realizable even when $\overline{N}$ is finite. Dimensionality orthogonality occurs because the basis function $\varphi(t)$'s Fourier Transform translates by the offsets $f = \frac{n}{T}$ do not overlap and thus are trivially orthogonal (trivially through Parseval's Theorem) at the linear time-invariant channel output. ISI is nearly zero because these bands that are nearly flat at the channel output. This suggests the possibility of approximating the MT functions with physically realizable basis functions, as generally depicted in Figure 4.23. There is no guard band in the time domain, but instead an excess bandwidth in the frequency domain.

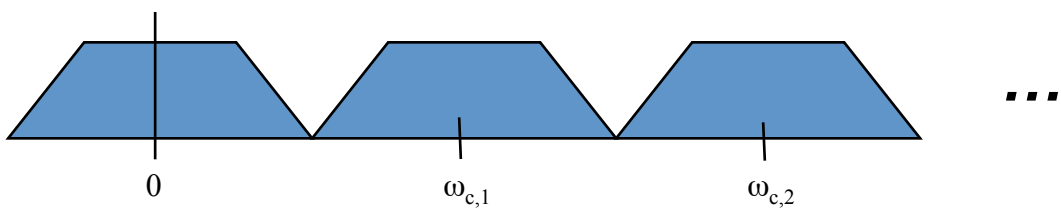


Figure 4.23: Filters with multitone.

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 $\varphi(t) = \frac{1}{\sqrt{T}} \cdot \text{sinc}\left(\frac{t}{T}\right)$. One method might be to simply truncate this basis function to say $\pm$ 100 symbol periods, thus introducing a delay of approximately $101T$ delay in realization (and introducing a high complexity), and of course necessarily another $101T$ processing delay in the receiver. Contrast of this with a delay of just T with the use of the guard period and the method appears unattractive. However, shorter length approximations may suffice. One method introduces again Chapter 3's raised cosine functions 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. Other functions may also achieve the same type of "low side bands" for the Fourier transform of the basis function. Any such realization requires in reality some type of over-sampled digital signal processing. Section 4.7, Subsection **??** studies discrete realizations of a few methods that have been proposed. However, the most heavily used method appears next in Section 4.6.

4.6 Discrete-time channel partitioning

While MM restricts nonzero sign to a finite time interval T for transmit basis-function construction, 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 subsymbols. 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 channel's exact interdimensional interference. Further, the time-frequency or space-time descriptions have the same matrix-multiply description, as will become evident in Subsection 4.6.2, but Subsection 4.6.1 first develops the SISO case with $L_x = L_y = 1$ to provide insight between discrete channel partitioning and discrete-time convolution with matrix multiplication. $N = 2\overline{N}$ remains for complex baseband systems. Figure 4.24 illustrates the SISO $\overline{N}$-dimensional discrete-time channel partitioning.

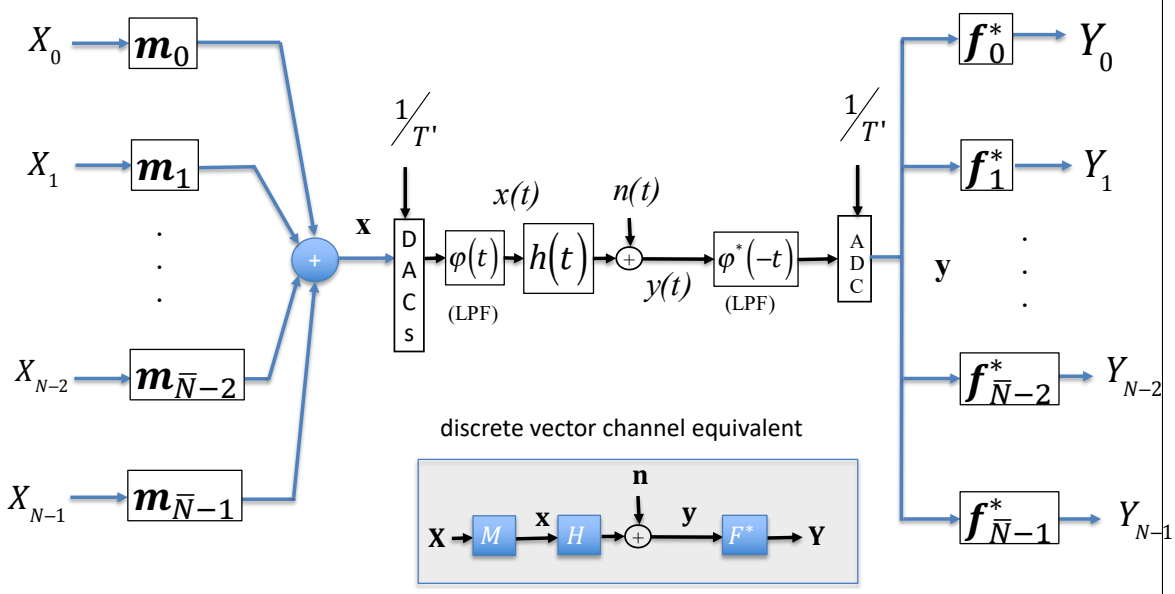


Figure 4.24: Digital realization of channel partitioning.

Discrete-time **basis vectors**, $\boldsymbol{m}_n$ replace the continuous basis functions of MM or MT. The dimensional index has been subtly changed to run from $n = 0, ..., \overline{N}-1$ instead of $1, ..., \overline{N}$ so that designers can denote the DC subchannel by index 0. Also, the transmitter inputs (and receiver filters' outputs) use capital letters to distinguish them from the same-letter-but-not-capitalized inputs to those same filters. These basis vectors have a finite length of $\overline{N} + \nu$ samples[38] over a duration of time T. The sampled pulse response, $h(kT')$, of the channel[39] is spans a time interval of less than $\nu + 1$ sample periods, where $(\nu + 1)T' = T_H$ for time-frequency. Each basis vector, $\boldsymbol{m}_n$, multiplies a sub-symbol element X_n before

[38] $\nu = 0$ will correspond to the space-time channel without ISI can still be non-trivial (non-diagonal, corresponding to crosstalk). $\nu > 0$ helps mitigate non-zero ISI.

[39] The transmit filter is shown as a basis function just after the DAC in Figure 4.24. Thus, it is appropriate to call the convolution of $h_c(t)$ with this filter $\varphi(t)$ the pulse response $h(t) = h_c(t) * \varphi(t)$. In many situations, and indeed later in this text, the channel response including any transmit and receive anti-alias/filtering are all part of $h(t)$. At this point, this text avoids that temptation while the student is still expanding from Chapter 3 into Chapter 4 and beyond. In practice, the real basis functions are implemented in digital signal processing.

being summed to form the transmit symbol vector $\boldsymbol{x}$.[40] Then, $\boldsymbol{x}$ passes through a digital-to-analog converter (or set of them for the MIMO case) into the transmit filter and continuous-time channel. The combined post-DAC filtering, channel impulse response, and pre-ADC filtering in the receiver form a pulse response $h(t)$. 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 matrix channel's dispersive effect, just as orthogonality was maintained with MM.

Discrete Channel: The receiver low-pass filters with gain $1/\sqrt{T'}$ (that $h(t)$ also includes) to maintain noise per dimension at $\frac{\mathcal{N}_0}{2}$), and then samples (ADC) at rate $1/T'$ the received modulated signal. The receiver filters are discrete-time and also finite length. The receiver filters are only complex if the entire set of modulated signals had been passband modulated by an external passband modulator that Figure 4.24 does not show, and in this case $h(t)$ corresponds to a complex baseband-equivalent channel. There are $\overline{N}+\nu$ input samples that lead to $\overline{N}$ matched-filter output samples; the receiver discards ν samples in the guard period to avoid intersymbol interference.

4.6.1 Optimal Partitioning Vectors - SISO Vector Coding

The last $\overline{N}$ ADC channel-output symbol samples in Figure 4.24 have vector representation, with time indexed within a symbol from $k=-\nu$ to $\overline{N}-1$,

$$\begin{bmatrix} y_{\overline{N}-1} \\ y_{\overline{N}-2} \\ \vdots \\ y_0 \end{bmatrix} = \begin{bmatrix} h_0 & h_1 & ... & h_\nu & 0 & ... & 0 & 0 & 0 & 0 \\ 0 & h_0 & \ddots & h_{\nu-1} & h_\nu & \ddots & 0 & 0 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & ... & 0 & 0 & 0 & 0 & h_0 & h_1 & ... & h_\nu \end{bmatrix} \begin{bmatrix} x_{\overline{N}-1} \\ \vdots \\ x_0 \\ x_{-1} \\ \vdots \\ x_{-\nu} \end{bmatrix} + \begin{bmatrix} n_{\overline{N}-1} \\ n_{\overline{N}-2} \\ \vdots \\ n_0 \end{bmatrix} \quad , \tag{4.226}$$

or more compactly,

$$\boldsymbol{y} = H \cdot \boldsymbol{x} + \boldsymbol{n} \quad . \tag{4.227}$$

The matrix H need not be Toeplitz in the MIMO case and has dimensionality equal to the number of output dimensions by the number of input dimensions in general. The $\overline{N} \times (\overline{N}+\nu)$ matrix H has a **singular value decomposition**

$$H = F \begin{bmatrix} \Lambda \vdots \underbrace{\mathbf{0}_{-1,-\nu}}_{\overline{N}\times\nu} \end{bmatrix} M^* \quad , \tag{4.228}$$

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

In this time-indexed case[41], **Vector Coding** creates a set of $\overline{N}$ parallel independent time-indexed channels by using the transmit basis vectors, $\boldsymbol{m}$, that are the first $\overline{N}$ columns of M - in other words,

[40] 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.

[41] More generally, Vector Coding simply creates a set of parallel independent channels, but this section focuses on use with guard periods.

the transmit vector $\boldsymbol{x}$ forms from the $\overline{N}$ vector components X_n, $n = 1, ..., \overline{N}$ according to

$$\boldsymbol{x} = M \begin{bmatrix} \boldsymbol{X} \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \left[\boldsymbol{m}_{\overline{N}-1}\ \boldsymbol{m}_{\overline{N}-2} \ldots \boldsymbol{m}_1 \boldsymbol{m}_0 \ldots \boldsymbol{m}_{-\nu}\right] \begin{bmatrix} X_{\overline{N}-1} \\ X_{\overline{N}-2} \\ \vdots \\ X_0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \sum_{n=0}^{\overline{N}-1} X_n \cdot \boldsymbol{m}_n \quad . \tag{4.229}$$

4.6.1.1 Creating the set of Parallel Channels for Vector Coding

The last ν columns of M that do not affect H because they are multiplied by zero in (4.229), 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

$$\boldsymbol{Y} = F^* \boldsymbol{y} = \begin{bmatrix} \boldsymbol{f}^*_{\overline{N}-1} \boldsymbol{y} \\ \ldots \\ \boldsymbol{f}^*_0 \boldsymbol{y} \end{bmatrix} \quad . \tag{4.230}$$

The mathematical description of the resultant $\overline{N}$ parallel channels is

$$\boldsymbol{Y} = \Lambda \cdot \boldsymbol{X} + \boldsymbol{N} \quad , \tag{4.231}$$

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

$$Y_n = \lambda_n \cdot X_n + N_n \quad . \tag{4.232}$$

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 4.6.1 [Optimality of Vector Coding] *Vector Coding has the maximum $SNR_{m,u}$ for any discrete channel partitioning.*

Proof: M is invertible, F is invertible, and the channel dimensions are independent. Thus mutual information remains the same according the the preservation of information under invertible transformation, as per the Lemma in Section 2.3. **QED.**

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 $\overline{N} >> \nu$, 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 Sections 3.10-3.12 for a MIMO linear and decision-feedback equalizers that would limit the intersymbol interference between successive blocks without use of a guard period).

4.6.1.2 Comment on correlated noise:

Figure 4.24's discrete-time transmission system has noise that is unlikely to be white. Section 1.3.7 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 1.3.7 factors the noise autocorrelation to

$$\mathbb{E}\left[\boldsymbol{n}\boldsymbol{n}^*\right] = R_{\boldsymbol{nn}} \cdot \sigma^2 = R_{\boldsymbol{nn}}^{1/2} \cdot R_{\boldsymbol{nn}}^{*/2} \cdot \sigma^2 \quad , \tag{4.233}$$

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

$$\boldsymbol{y} \leftarrow R_{\boldsymbol{nn}}^{-1/2} \cdot \boldsymbol{y} = \left(R_{\boldsymbol{nn}}^{-1/2} \cdot H\right) \cdot \boldsymbol{x} + \tilde{\boldsymbol{n}} \quad . \tag{4.234}$$

where $R_{\boldsymbol{nn}}^{-1/2}$ is any matrix square root[42]. Equation (4.228)'s SVD applies to the noise-equivalent channel $\widetilde{H} = R_{\boldsymbol{nn}}^{-1/2} \cdot H$ rather than just H. In general, VC does not require $\widetilde{H}$ to be Toeplitz, so any choice of matrix square root produces the same overall performance (the product of singular values does not change with square-root choice[43]).

4.6.1.3 Vector-Coding SNR

VC's number of bits/symbol for constant-gap coding (regardless of b_n) on all dimensions is

$$b = (1/k) \cdot \sum_{n=0}^{\overline{N}-1} \log_2\left(1 + \frac{\mathrm{SNR}_n}{\Gamma}\right) \quad , \tag{4.235}$$

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 equals Chapter 2's "mutual information." With the water-fill energy distribution, b is the capacity $b = \mathcal{C}$ 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:

$$\mathcal{E}_n + \frac{\frac{\mathcal{N}_0}{2} \cdot \Gamma}{\lambda_n^2} = K \tag{4.236}$$

$$\sum_{n=0}^{\overline{N}^*-1} \mathcal{E}_n = (\overline{N} + \nu) \cdot \bar{\mathcal{E}}_{\boldsymbol{x}} \tag{4.237}$$

The SNR for vector coding is then

$$\mathrm{SNR}_{vc} = \Gamma \cdot \left[\prod_{n=1}^{\overline{N}^*}\left(1 + \frac{\mathrm{SNR}_n}{\Gamma}\right)\right]^{1/(\overline{N}+\nu)} - \Gamma \quad . \tag{4.238}$$

The quantity $(\overline{N} + \nu) \cdot \bar{\mathcal{E}}_{\boldsymbol{x}}$ is now the total energy. The exponent depending on $\overline{N} + \nu$ reflects vector coding's use of $\overline{N} + \nu$ channel input dimensions[44]. For complex passband systems,

$$\bar{b} = \frac{\tilde{b}}{2} = \frac{b}{2(\overline{N} + \nu)} = \frac{1}{2} \cdot \log_2\left(1 + \frac{\mathrm{SNR}_{vc}}{\Gamma}\right) \quad , \tag{4.239}$$

and for real (PAM) baseband systems

$$\bar{b} = \tilde{b} = \frac{b}{(\overline{N} + \nu)} = \frac{1}{2} \cdot \log_2\left(1 + \frac{\mathrm{SNR}_{vc}}{\Gamma}\right) \quad . \tag{4.240}$$

[42] Matlab's sqrt command provides such a square root, which is not unique.

[43] $R_{\boldsymbol{nn}}^{-1/2} \cdot H = Q \cdot \Lambda_{\boldsymbol{n}} \cdot Q^* \cdot F \cdot \Lambda \cdot M^*$, which has $\left|R_{\boldsymbol{nn}}^{-1/2} \cdot H\right| = |\Lambda_{\boldsymbol{n}} \cdot \Lambda|$ for an and all unitary M, F, and Q.

[44] The number of samples is $\overline{N} + \nu$ dimensions for real channels, and also $\overline{N} + \nu$ for complex dimensions oncomplex channels. This means the designer must consider ν in context as the length of the guard period in real or complex samples as applies in each case. In the complex case, each SNR factor would be squared and then (4.238) holds in either the complex or real case without further modification

So, in either case, $\bar{b}$ is the same.

Thus, the designer can compare vector coding's geometric SNR to an AWGN's detection SNR; Chapter 3's various MMSE-LEs or MMSE-DFEs because SNR_{vc} has the same relation to achievable data rate with the same class of codes with constant gap Γ. Further, with $\bar{\mathcal{I}}$ as the sum of the capacities for any input energy distribution choice,

$$\text{SNR}_{vc} = 2^{2\cdot\bar{\mathcal{I}}} - 1 \quad , \tag{4.241}$$

and

$$\text{SNR}_{vc} = 2^{2\cdot\bar{\mathcal{C}}} - 1 \tag{4.242}$$

when water-filling is used with $\Gamma = 1$. Thus the SNR is maximum when the input distribution is water-filling. This SNR_{vc} could then replace SNR_{MFB} as the highest possible SNR on an ISI channel, and SNR_{vc} is always achievable. SNR_{vc} needs to be viewed with respect to the used bandwidth – Chapter 3's SNR_{MFB} is usally a function of a different transmit-filter choice than VC and is often not achievable. Also, SNR_{vc} is highest when capacity-achieving codes are used and is a function of the gap, while SNR_{MFB} is independent of Γ. Nonetheless, SNR_{vc} is a better, and attainable, indicator of maximum performance.

EXAMPLE 4.6.1 [Vector Coding for $1 + .9 \cdot D^{-1}$ 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 $\overline{N} = 8$ 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

$$H = \begin{bmatrix} .9 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & .9 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & .9 & 1 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & .9 & 1 \end{bmatrix} \quad . \tag{4.243}$$

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

$$[1.87\ 1.78\ 1.64\ 1.45\ 1.22\ .95\ .66\ .34] \quad , \tag{4.244}$$

and corresponding subchannel SNR's

$$g_n = \frac{\lambda_n^2}{\sigma^2} = [19.3\ 17.6\ 15.0\ 11.7\ 8.3\ 5.0\ 2.4\ .66] \quad . \tag{4.245}$$

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

$$K = \frac{1}{7}\left(9 + \sum_{n=0}^{6} \frac{1}{g_n}\right) = 1.43 \quad . \tag{4.246}$$

The corresponding subchannel energies are:

$$[1.38\ 1.37\ 1.36\ 1.34\ 1.30\ 1.23\ 1.01\ 0] \quad , \tag{4.247}$$

and the SNR's are

$$[26.6\ 24.2\ 20.4\ 15.8\ 10.8\ 6.2\ 2.4\ 0] \quad , \tag{4.248}$$

resulting in an SNR of

$$\text{SNR}_{VC} = \left[\prod_{n=0}^{6} \text{SNR}_n + 1\right]^{1/9} - 1 = 6.46 = 8.1 \text{ dB} \quad . \tag{4.249}$$

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

$$\bar{c} = \frac{1}{9}\sum_{n=0}^{6}\frac{1}{2}\log_2(1+\text{SNR}_n) = 1.45 \text{ bits/dim.} \tag{4.250}$$

The capacity is close to the true capacity of 1.55 bits/dimension, which would be determined by executing this example as $\overline{N} \to \infty$. These subchannels are exactly independent, and the SNR and capacity are exact - no approximations of no-ISI on a subchannel were made. The SNR as $\overline{N} \to \infty$ is then 8.8 dB, consistent with Chapter 3's results. However, no error propagation and the detector is optimum. The overhead of $\nu = 1$ becomes negligible as $\overline{N}$ increases. It should be noted that Chapter 8's Viterbi or sequence-detector for this channel might appear to have a higher SNR since it achieves the MFB performance level; however such a system effectively has a code with a different gap so cannot be compared fairly here using gap analysis. Indeed that system uses a non-waterfilling spectra and so would perform worse with a $\Gamma = 0$ dB code.

4.6.1.4 More general H matrices and vector coding

The structure of the convolution (Toeplitz) matrix H in Equation (4.226) reflects a linear-time-invariant channel with successive transmission. However, SVD and vector coding may be applied to any matrix H – that is, even without convolutional structure. Channels may be derived in communication in many ways. The space-time MIMO channel for instance does not usually have a Toeplitz structure across the spatial dimensions. Other channels may process multi-dimensional images (photos, videos) and be described by a simple matrix H that could have inter-dimensional interference. As long as that channel is linear, it can be described by H multiplying some input $\boldsymbol{x}$. Thus, vector coding is an important tool for any Matrix AWGN channel.

4.6.2 Expanding to Spatial MIMO

A general form of H occurs when spatial MIMO adds dimensionality to the channel with either or both $L_x > 1$ and/or $L_y > 1$, but $\nu \geq 0$ because there is also a convolution in time ongoing on each of the possibly spatial paths. In this case, each of the transmit basis-vectors m_n generalizes to an $(\overline{N}+\nu)\times L_x$ set of vectors $\boldsymbol{m}_n$. Similarly, each of the receiver matched-vectors f_n^* becomes an $\overline{N}\times L_y$ set of vectors $\boldsymbol{f}_n^*$. Each input sample can be viewed as an $L_x \times 1$ vector so that the overall input becomes the $(\overline{N}+\nu)L_x \times 1$ input vector:

$$\boldsymbol{x} = \sum_{n=0}^{\overline{N}-1}\boldsymbol{m}_n \cdot \boldsymbol{x}_n = M \cdot \boldsymbol{X} \quad . \tag{4.251}$$

The outputs after matched filtering will be the $\overline{N}$ sets of $L_y \times 1$ vectors

$$\boldsymbol{Y}_n = F_n^* \cdot \boldsymbol{y} \tag{4.252}$$

The channel is an $\overline{N} \cdot L_y \times (\overline{N}+\nu)L_x$ matrix $\boldsymbol{H}$. A singular value decomposition can be calculated for this new larger matrix and vector coding applied in the same way as previously in this section with all the same optimality.

There are many ways of organizing or re-indexing the inputs, but one in particular that will have $\overline{N}^2$ separate matrices, each of size $L_y \times L_x$ will be convenient in what is called Vector DMT or Vector OFDM, a method heavily used in highest performing communication designs, as in Section 4.7.

Some applications individually limit the energy for each of the L_x transmitters (instead of just the overall sum of energies). This will be called an **link energy constraint**. However, loading must check this constraint for bit addition on any dimension (clearly deleting a bit will not cause the energy constraint to be violated). The link index corresponding to next least energy to be added is called i, where i could be any of 1, 2, ..., L_x. For LC algorithms, the sum of energies for this i^{th} link

$$\mathcal{E}_i = \sum_{n=1}^{\overline{N}}\mathcal{E}_{i,n} \leq \mathcal{E}_{max,i} \tag{4.253}$$

is the link-energy-constraint check. If this constraint is violated, then LC must check the incremental energy table on all remaining indices $l \neq i$, $l = 1, ..., L_x$ for the next-to-next least energy increment. Similarly in turn, individual link-energy constraints are checked until the bit can be added (or the addition of a bit is not possible). Clearly this algorithm simplifies to normal LC when $L_x = 1$.

For waterfilling, the algorithm adjustment is similar to the shaped-lid water-filling that occurred earlier for PSD constraints. In this case though, the dimensional-specific constraint is not a PSD but the link-energy sum constraint. Thus, for each K in water-filling a second check (after the negative energy check) needs to be executed on the sum of energies for each "link" (or more generally set of dimensions). Energy is filled only to the constraint, and the remaining energy increases the value of K with this link removed from further consideration. This may cause other links to violate their link-energy constraints - if so they are removed similarly. The process continues until all energy is used (or all constraints have been met). In the MA case, the process continues until the desired bit-rate is attained.

4.6.2.1 Space-Time Block Coding (STBC)

Space-Time coding aggregates several successive time-indexed symbols of a matrix AWGN's input into a larger symbol called a **block**. AWGN codes, as per the separation theorem or with variable bit-loading water-fill, continue to apply to the successive blocks' dimensions as if the entire system were an AWGN (or in the water-filling case, several AWGNs with different bit distributions and energies). While VC remains optimum, it depends on the channel-dependent SVD $H = F \cdot \Lambda \cdot M^*$ to determine best transmit and receive filters. Subject to the guard-period dimensionality loss,

$$\gamma_{guard} = \frac{\overline{N}}{\overline{N} + \nu} \quad , \tag{4.254}$$

VC is optimum for any energy distribution.[45] STBCs ignore this time-frequency dimensionality loss development, addressing only spatial crosstalk between dimensions and no intersymbol interference[46].

Space-Time Block Codes use an expanded channel-independent transmit-modulation-vector set that "orthogonalizes" the channel, with a channel-dependent spatial-MMSE-LE receiver matrix multiply. STBCs expand dimensionality, beyond the minimally necessary spatial dimensionality, through time-repetition of channel-input symbols (often with some reordering of dimensions within those symbols). The expanded dimensionality requires that the bits/dimension increase if fair comparison with a VC system for the same channel H is to occur. For instance, a system that repeats a channel input symbol

$$\boldsymbol{x} \rightarrow \begin{bmatrix} \boldsymbol{x} \\ \boldsymbol{x} \end{bmatrix} \tag{4.255}$$

would need to use $\bar{b} \rightarrow 2 \cdot \bar{b}$ to retain an equal number of bits per used input dimension with a system that does not redundantly repeat $\boldsymbol{x}$. STBCs must have $\nu_{ST} \geq L_x$. This creates a dimensional-loss factor, which multiplies any guard-period loss factor that already represents the penalty for intersymbol-interference compensation, namely γ_{guard}, so total dimensional loss is

$$\gamma_{STBC} = \underbrace{\frac{L_x}{L_x + \nu_{ST}}}_{\gamma_{repeat}} \cdot \underbrace{\frac{\overline{N}}{\overline{N} + \nu}}_{\gamma_{guard}} \quad . \tag{4.256}$$

All STBCs always have $\gamma_{repeat} \leq 0.5$. As long as the γ_{guard} is the same for compared systems, and it is occurring external to the space-time designs, STBC design need not explicitly consider it.

Most treatments of STBC's ignore the γ_{repeat} dimensionality increase in defining a "rate" that is between 0 and 1 because of one special case that has the unity rate ONLY for the 1×2 channel matrix $H = [h \;\; h]$. This special case blocks two successive one-dimensional channel output sysmbols to

[45] The highest data rate, or highest margin, occurs for the water-filling distribution, but VC is the best simple AWGN ML-detector system for any input $\boldsymbol{R_{xx}}$.

[46] The Vector DMT systems of Section **??** will indeed produce a square (and cyclic) H matrix and that matrix often appears as the H matrix here with only crosstalk and the γ_{guard} will apply but otherwise not appear in STBC.

correspond to a block of two successive two-dimensional input symbols, as in the Alamouti Code below. Good design properly understands that $\gamma_{STBC} \leq \frac{1}{2} \cdot \gamma_{guard}$ in fair comparisons with spatial vector coding, for any channel that is not this trivial 1×2 repetition channel. The extra dimensionality-loss factor γ_{repeat} provides the benefit of allowing a channel-independent modulator that also has certain diversity benefit. Specifically, channel outputs (after a simple block MMSE-LE matrix multiply) have no crosstalk from other spatial dimensions and so outer AWGN decoders then directly apply. The repeated input leads to a dimensionally constant increased SNR, creating a "diversity" effect that assures designers that no matter what the actual channel in practice, the system will provide a robust symmetry in performance for inputs of equal energy/dimension. However, this higher SNR is almost always more than negated by the losses caused through the lost dimensions when data rate is consistently constant, with the sole exception being the trivial 1×2 case.

Alamouti Code: The most common STBC is an **Alamouti Code**[47][21] where $L_x = 2$ and $\overline{N} = 2$, resulting in the least possible dimensionality loss of $\gamma_{STBC} = 0.5$ (when $\gamma_{guard} = 1$), except for the 1×2 trivial channel that has $L_y = 1$. Figure 4.25 illustrates this trivial case (and the reader can initially ignore all complex conjugates there and divide all real dimensional counts by 2 since real-baseband uses 1/2 the number of dimensions of complex baseband). Other STBC's have larger dimensionality losses but may apply to larger spatial dimensionality, $L_x > 2$. **Spatial vector coding has no such STBC-like dimensional loss factor** and will always peform at least as well as STBCs and often significantly better.

The Alamouti code follows from the trival channel's SVD (vector coding)

$$H = [h \;\; h] = \underbrace{1}_{F} \cdot \underbrace{\left[\sqrt{2} \cdot h \;\vdots\; 0\right]}_{\Lambda} \cdot \underbrace{\begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix}}_{M^*} \; . \tag{4.257}$$

The F and M matrices do not depend on h for this channel. The columns of M, if applied to two successive scalar time inputs v_2 and v_1 blocked into a 2×1 input vector $\boldsymbol{x} = M \cdot \boldsymbol{v}$, look like

$$x_2 = \frac{1}{\sqrt{2}} \cdot (v_2 + v_1) \tag{4.258}$$

$$x_1 = \frac{1}{\sqrt{2}} \cdot (v_2 - v_1) \; . \tag{4.259}$$

The zero-noise channel output block after matched-filter matrix H^* is

$$H^* \cdot H \cdot \boldsymbol{x} = 2 \cdot h^2 \cdot \begin{bmatrix} v_2 \\ v_1 \end{bmatrix} \; . \tag{4.260}$$

This one special case has vector coding and Alamouti code the same. For $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$ and $\sigma^2 = 1$, these systems also have the same $SNR = 2 \cdot h^2$ as two repeated transmitted messages x_i if each simply transmits at its own time through both the transmit antennas. In this later case, the channel H effectively adds them for the scalar receiver input, which repeats for a block. Each sample would also have the same SNR. When the channel is complex baseband, the Alamouti Code extents to

$$x_2 = \frac{1}{\sqrt{2}} \cdot (v_2 + v_1^*) \tag{4.261}$$

$$x_1 = \frac{1}{\sqrt{2}} \cdot (v_2 - v_1) \; . \tag{4.262}$$

with $SNR = 2\,|h|^2$ and again is the same performance as vector coding with channel-independent inputs. The receiver is again channel dependent at H^*. The Alamouti code has essentially half of each input v_n

[47] After Iranian born Siavash Alamouti, 1962 -, also known as a great humanitarian for his aid to refugees escaping middle eastern war zones.

on each of the two 2-dimensional successive inputs to the channel, suggesting it is more diverse in time in this trivial case of vector coding and Alamouti being the same.

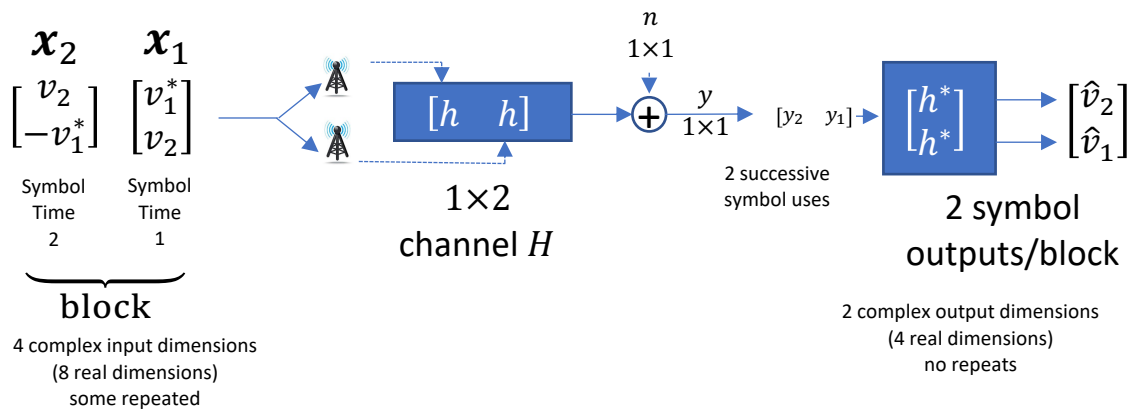


Figure 4.25: Alamouti Code for 1×2 "repeat" H where it is trivially vector coding. (*The Alamouti Code is suboptimal in any other configuration.*)

Alamouti STBC Channel Orthogonalization: Alamouti's Figure 4.26 extension to a 2×2 channel retains the transmit modulator's independence to the channel, albeit at a loss with respect to spatial vector coding.

First, Real 2×2 Baseband Case: It is helpful to start with a case where the channel is real baseband and so $\widetilde{N} = 1$. Two unitary symmetric real matrices, which are almost permutation matrices but for a sign change, are useful in this case:

$$J_2 \triangleq \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad ; \quad J_1 \triangleq \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \quad . \tag{4.263}$$

For a real symmetric matrix

$$R \triangleq \begin{bmatrix} a & c \\ c & b \end{bmatrix} \quad , \tag{4.264}$$

the following diagonalization identity holds by simple matrix multiplication

$$J_2 \cdot R \cdot J_2 + J_1 \cdot R \cdot J_1 = \begin{bmatrix} a+b & 0 \\ 0 & a+b \end{bmatrix} \quad . \tag{4.265}$$

The (2×2)-channel-size real-baseband Alamouti code has a 4-(real-)dimensional modulation: There are two spatial dimensions for each of two symbol periods per block, even though some authors call it two-dimensional because the input $\boldsymbol{v}$ remains two-real-dimensional like the 1×2 channel.

$$\underbrace{\boldsymbol{x}}_{4\times 1} = \underbrace{\begin{bmatrix} J_2 \\ J_1 \end{bmatrix}}_{4\times 2} \cdot \underbrace{\begin{bmatrix} v_2 \\ v_1 \end{bmatrix}}_{\boldsymbol{v},2\times 1} \quad . \tag{4.266}$$

Often, STBC work will abbreviate this code with the 2×2 matrix notation

$$\begin{bmatrix} x_2 & x_1 \\ -x_1 & x_2 \end{bmatrix} \quad , \tag{4.267}$$

where the x_n really would be v_n in this text's context (as well as others' contexts) but that is the abbreviation in use. While $\boldsymbol{x}$ truly has 4 dimensions, there are only two independent channel-input entries. A twice-used matrix AWGN, with 2×2 matrix

$$H = \begin{bmatrix} H_{22} & H_{21} \\ H_{12} & H_{11} \end{bmatrix} , \tag{4.268}$$

on each use, processes this channel as two independent symbols[48].

$$\boldsymbol{y} = \underbrace{\begin{bmatrix} H & 0 \\ 0 & H \end{bmatrix}}_{\boldsymbol{H}} \cdot \boldsymbol{x} + \boldsymbol{n} = \begin{bmatrix} H \cdot J_2 \\ H \cdot J_1 \end{bmatrix} \cdot \boldsymbol{v} + \boldsymbol{n} \ , \tag{4.269}$$

Section 2.7 introduced a (2×2 in this case) channel autocorrelation matrix $R_f = \boldsymbol{H}^* \cdot \boldsymbol{H}$ when inputs were considered different users, but applies here also. When that matrix (as well as the inverse backward matrix $R_b^{-1} = R_f + I$) is diagonal, the canonical receiver becomes maximum likelihood and attains optimal performance without need for processing beyond the a linear (MMSE-LE matrix) filter for the block. The optimal estimate of $\boldsymbol{v}$ follows from (4.265) with some algebra where $R_{\boldsymbol{nn}} = I$ (or the equivalent white-noise channel $\widetilde{H}$ replaces H everywhere). Then, since each 2×2 block-diagonal element of R_f is

$$\begin{aligned} R_{f,2\times2} &= J_2^* \cdot H^* \cdot H \cdot J_2 + J_1 \cdot H^* \cdot H \cdot J_1 && (4.270) \\ &= \begin{bmatrix} H_{22}^2 + H_{21}^2 + H_{12}^2 + H_{11}^2 & 0 \\ 0 & H_{22}^2 + H_{21}^2 + H_{12}^2 + H_{11}^2 \end{bmatrix} && (4.271) \end{aligned}$$

is diagonal. The SNRs for each of the two input dimensions estimation are

$$SNR_2 = SNR_1 = \|\boldsymbol{H}\|_F^2 \ . \tag{4.272}$$

This is a larger SNR than VC, **but is for a reduced data rate and is suboptimal to VC**. Thus, the MMSE-LE Matrix (maximum likelihood) receiver is

$$W = \begin{bmatrix} \frac{1}{\|\boldsymbol{H}\|_F^2+1} & 0 \\ 0 & \frac{1}{\|\boldsymbol{H}\|_F^2+1} \end{bmatrix} \cdot \begin{bmatrix} J_2^* \cdot H^* & \vdots & J_1^* \cdot H^* \end{bmatrix} . \tag{4.273}$$

The unbiased receiver removes the denominator +1 terms.

Second, Complex 2×2 Alamouti STBC Code Extension: The complex case is most easily understood with a slight abuse of matrix notation to redefine J_2 and J_1 as

$$J_2 \triangleq \begin{bmatrix} 1 & 0 \\ 0 & -\text{conj}(\cdot) \end{bmatrix} \quad ; \quad J_1 \triangleq \begin{bmatrix} 0 & \text{conj}(\cdot) \\ 1 & 0 \end{bmatrix} \ , \tag{4.274}$$

where conj$(\cdot)$ means multiply by 1 but take the complex conjugate of the scalar quantity being multiplied. The complex Alamouti code thus has 8 real dimensions. Both Figures 4.25 and 4.26 illustrate this complex case, respectively for the 2×1 trivial channel and the 2×2 channel.

For a complex Hermitian symmetric matrix

$$R \triangleq \begin{bmatrix} a & c^* \\ c & b \end{bmatrix} \ , \tag{4.275}$$

all of the above equations (4.265) - (4.273) apply and the system remains ML and orthogonal, but again with $\gamma_{repeat} = 0.5$ so 1/2 the complex dimensions are spent to orthogonalize the channel with channel-independent transmit matrix modulator. Complex vector coding again instead has $\gamma_{repeat} = 1$, but also

[48] Guard periods are assumed but not shown; and the corresponding 2×2 resultant square H would contain only rows of M^* that correspond to nonzero singular values in reforming this square $H = F \cdot \Lambda_{\text{square nonsingular}} \cdot M^*_{\text{nonsingular components}}$.

has a channel-dependent matrix modulator and may have smaller SNRs, but more of them in estimating twice as many independent channel inputs, and will always perform at least as well and usually much better as in this next example.

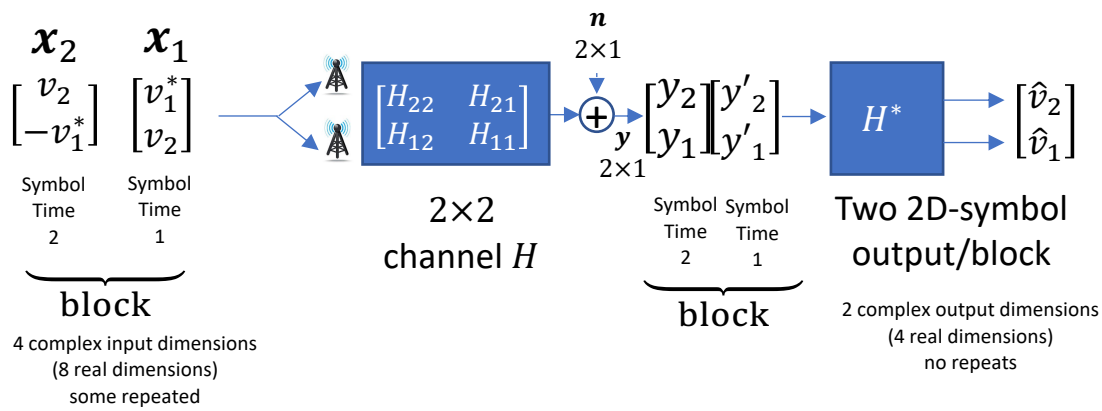


Figure 4.26: Alamouti Code for 2×2 general H where it is **NOT** optimal.

EXAMPLE 4.6.2 *[Vector Coding performance advantage over STBC]* A 2×2 complex channel has

```
>> H
H =
    1.0000    0.9000
    0.8000    1.0000
>> [F,L,M]=svd(H)

F =
   -0.7245   -0.6892
   -0.6892    0.7245
L =
    1.8512         0
         0    0.1512
M =
   -0.6892   -0.7245
   -0.7245    0.6892
```

With $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$ and $\sigma^2 = 1$, the number of bits per (two complex dimensions) for VC and STBC are respectively then

```
bvc=log2(det(L^2+eye(2))) =
    2.1790
bstbc=log2(sqrt(norm(H,'fro')^2+1)) =
    1.0769
```

The square root in the bstbc calculation accounts for the fact that the two-dimensional input is repeated while vector coding's two-dimensional input is for one vector-symbol use (not two repeated successive uses like the Alamouti Code in this example). Equivalently if both systems use the same data rate of 2.1790, the vector coding system is

```
>> 10*log10((2^bvc-1)/(2^bstbc - 1))  =    5.0245 dB
```

better in its use of energy. At higher energy or SNR levels, consequently with larger values of bvc and bstbc, the vector coding advantage increases.

The simple example shows a 5 dB advantage of vector coding over Alamouti, with the same diversity essentially by using either the M matrix for a modulator or the Alamouti code's simpler channel-independent modulator. The receiver in both cases is channel dependent and the same complexity, although calculation of F could be slightly higher before it is used (however, recall F is the unbiased MMSE-LE so could be determined if M is given by simple gradient search). The advantage of vector coding typically is larger in more complicated channels and would be larger with spatial water-filling energy. Additionally, with time-varying channels like Section 1.7's Raleigh fading, an average (over channels) SNR determines performance. The error probability relationship[49] to average SNR decays only in proportion $\propto (SNR)^{-d_{free}}$. This is less than the stationary channel's exponential decay with SNR. The STBC can increase d_{free} through repetition although if both systems are using good outer codes, STBC's symbol repetition has a bits/symbol cost that still is disadvantageous to STBC relative to vector coding (which could lower its rate and thus effectively use a larger d_{free} outer code). Some good examples of VC's gain over STBC for the IEEE 802.11 Wi-Fi series of standards appear a book written by the standards leaders for Wi-Fi (Chapter 6) [19] that has several examples illustrating almost vector coding methods[50] and performance averaged over SNRs for various in-home standards model. The advantages there over STBC are yet larger, albeit it at higher SNRs and bit rates than Example 4.6.2. For these reasons, many practical designs (such as the Wi-Fi and Cellular examples later in this subsection) attempt to approximate or quantize different pre-stored M matrices rather than use STBCs. Field systems often index and exchange said index for the appropriate M choices, while the F matrix is learned adaptively through a simple MMSE learning algorithm or calculation. This averts SVD's complexity. At very low SNRs, the vector coding advantage becomes minimal, and essentially STBC's robust redundant repeating input symbols and channel independence can become an appreciated advantage, essentially allowing connectivity at very low data rates without needing to alter the exterior code to do so. Thus, use of an STBC in a strained situation with low SNR may be a simple solution and so it often also appears as an option for field systems.

Larger-Dimensionality STBC Codes with real dimensions: The channel model in (4.269) expands to a block-diagonal of $\overline{N}$, $L_x \times L_x$, repeats of H multiplying an $(L_x \cdot \overline{N}) \times \overline{N}$ matrix with $L_x \times \overline{N}$ matrix $J_{\overline{N}}$ at the top and similarly J_1 at the bottom, see [20]. There are only two more real-dimensional STBC codes that attain the upper bound of $\gamma_{repeat} = \frac{\#\text{ input symbols}}{L_x \cdot \overline{N}} = \frac{\overline{N}}{L_x \cdot \overline{N}} = \frac{1}{L_x}$ because $\overline{N} = L_x$ for these codes. For 4 antennas:

$$J_4 \triangleq \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \quad J_3 \triangleq \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \tag{4.276}$$

$$J_2 \triangleq \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad J_1 \triangleq \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} . \tag{4.277}$$

The dimensionality loss is $\gamma_{repeat} = 0.25$ because there are 4 real inputs to 4 antennas over 4 successive time symbols in the block, so 3/4 of the dimensions are lost. Vector Coding instead still has $\gamma_{repeat} = 1$, but STBC's $\|\boldsymbol{H}_F\|^2$ is larger yet for 4 dimensions, but again almost never able to recover the value of

[49] This relationship follows directly from Example 1.6.1 in Chapter 1 where the $\mathcal{E}_g$ is simply replaced by $d_{free} \cdot \mathcal{E}_g$ for the derviation with coding gain.

[50] There called MRC, maximum ratio combining, which is really just MMSE-LE with a transmit matrix that tries to approximate M.

the lost dimensions that VC uses. The corresponding diagonal R_f matrix is

$$R_f = \sum_{\ell=1}^{L_x} J_\ell^* \cdot H^* \cdot H \cdot J_\ell \ . \tag{4.278}$$

The last 8-real-dimensional system, with $\gamma_{repeat} = 1/8$ has

$$J_8 \triangleq \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \quad J_7 \triangleq \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \tag{4.279}$$

$$J_6 \triangleq \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad J_5 \triangleq \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.280}$$

$$J_4 \triangleq \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \quad J_3 \triangleq \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & - & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \tag{4.281}$$

$$J_2 \triangleq \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad J_1 \triangleq \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.282}$$

The expression of STBC often abbreviates the J matrices by essentially doing the J_n multiplication to create the corresponding $\boldsymbol{x}$ column notation output, so thus the $L_x = \overline{N} = 8$-dimensional code is compactly written

$$\begin{bmatrix} x_8 & -x_7 & -x_6 & -x_5 & -x_4 & -x_3 & -x_2 & -x_1 \\ x_7 & x_8 & -x_5 & x_6 & -x_3 & x_4 & x_1 & -x_2 \\ x_6 & x_5 & x_8 & -x_7 & -x_2 & -x_1 & x_4 & x_3 \\ x_5 & -x_6 & x_7 & x_8 & -x_1 & x_2 & -x_3 & x_4 \\ x_4 & x_3 & x_2 & x_1 & x_8 & -x_7 & -x_6 & -x_5 \\ x_3 & -x_4 & x_1 & -x_1 & x_7 & x_8 & x_5 & -x_6 \\ x_2 & x_1 & -x_4 & -x_3 & x_6 & -x_5 & x_8 & x_7 \\ x_1 & x_2 & -x_3 & -x_4 & x_5 & x_6 & -x_7 & x_8 \end{bmatrix} \tag{4.283}$$

There are real-dimensional STBC codes that have $\overline{N} > L_x$ but maintain the best $\gamma_{repeat} = \frac{1}{L_x}$ with

$L_x = 3$ and $\overline{N} = 4$ for $\gamma_{repeat} = 1/3$:

$$\begin{bmatrix} x_4 & -x_3 & -x_2 & -x_1 \\ x_3 & x_4 & x_1 & -x_1 \\ x_2 & -x_1 & x_4 & x_3 \end{bmatrix} \tag{4.284}$$

$L_x = 5$ and $\overline{N} = 8$ for $\gamma_{repeat} = 1/5$:

$$\begin{bmatrix} x_8 & -x_7 & -x_6 & -x_5 & -x_4 & -x_3 & -x_2 & -x_1 \\ x_7 & x_8 & -x_5 & x_6 & -x_3 & x_4 & x_1 & -x_2 \\ x_6 & x_5 & x_8 & -x_7 & -x_2 & -x_1 & x_4 & x_3 \\ x_5 & -x_6 & x_7 & x_8 & -x_1 & x_2 & -x_3 & x_4 \\ x_4 & x_3 & x_2 & x_1 & x_8 & -x_7 & -x_6 & -x_5 \end{bmatrix} \tag{4.285}$$

$L_x = 6$ and $\overline{N} = 8$ for $\gamma_{repeat} = 1/6$:

$$\begin{bmatrix} x_8 & -x_7 & -x_6 & -x_5 & -x_4 & -x_3 & -x_2 & -x_1 \\ x_7 & x_8 & -x_5 & x_6 & -x_3 & x_4 & x_1 & -x_2 \\ x_6 & x_5 & x_8 & -x_7 & -x_2 & -x_1 & x_4 & x_3 \\ x_5 & -x_6 & x_7 & x_8 & -x_1 & x_2 & -x_3 & x_4 \\ x_4 & x_3 & x_2 & x_1 & x_8 & -x_7 & -x_6 & -x_5 \\ x_3 & -x_4 & x_1 & -x_1 & x_7 & x_8 & x_5 & -x_6 \\ x_2 & x_1 & -x_4 & -x_3 & x_6 & -x_5 & x_8 & x_7 \\ x_1 & x_2 & -x_3 & -x_4 & x_5 & x_6 & -x_7 & x_8 \\ x_3 & -x_4 & x_1 & -x_2 & x_7 & x_8 & x_5 & -x_6 \end{bmatrix} \tag{4.286}$$

$L_x = 7$ and $\overline{N} = 8$ for $\gamma_{repeat} = 1/7$:

$$\begin{bmatrix} x_8 & -x_7 & -x_6 & -x_5 & -x_4 & -x_3 & -x_2 & -x_1 \\ x_7 & x_8 & -x_5 & x_6 & -x_3 & x_4 & x_1 & -x_2 \\ x_6 & x_5 & x_8 & -x_7 & -x_2 & -x_1 & x_4 & x_3 \\ x_5 & -x_6 & x_7 & x_8 & -x_1 & x_2 & -x_3 & x_4 \\ x_4 & x_3 & x_2 & x_1 & x_8 & -x_7 & -x_6 & -x_5 \\ x_3 & -x_4 & x_1 & -x_1 & x_7 & x_8 & x_5 & -x_6 \\ x_2 & x_1 & -x_4 & -x_3 & x_6 & -x_5 & x_8 & x_7 \\ x_1 & x_2 & -x_3 & -x_4 & x_5 & x_6 & -x_7 & x_8 \\ x_3 & -x_4 & x_1 & -x_2 & x_7 & x_8 & x_5 & -x_6 \\ x_2 & -x-1 & -x_4 & x_3 & x_6 & -x_5 & x_8 & x_7 \end{bmatrix} \tag{4.287}$$

Larger-Dimensionality STBC Codes with complex dimensions: For the complex case, the conj$(\cdot)$ operation is indicated by a conjugate on the matrix entry. Further these codes may have fewer yet independent input symbols because a deterministic input symbol value and its conjugate are not independent. Two codes of higher dimension are: $L_x = 3$ and $\overline{N} = 8$ but the number of independent input symbols is 4 for $\gamma_{repeat} = 1/6$:

$$\begin{bmatrix} x_4 & -x_3 & -x_2 & -x_1 & x_4^* & -x_3^* & -x_2^* & -x_1^* \\ x_3 & x_4 & x_1 & -x_2 & x_3^* & x_4^* & x_1^* & -x_2^* \\ x_2 & -x_1 & x_4 & x_3 & x_2^* & -x_1^* & x_4^* & x_3^* \end{bmatrix} \tag{4.288}$$

$L_x = 4$ and $\overline{N} = 8$ but the number of independent input symbols is 4 for $\gamma_{repeat} = 1/8$:

$$\begin{bmatrix} x_4 & -x_3 & -x_2 & -x_1 & x_4^* & -x_3^* & -x_2^* & -x_1^* \\ x_3 & x_4 & x_1 & -x_2 & x_3^* & x_4^* & x_1^* & -x_2^* \\ x_2 & -x_1 & x_4 & x_3 & x_2^* & -x_1^* & x_4^* & x_3^* \\ x_1 & x_2 & -x_3 & x_4 & x_1^* & x_2^* & -x_3^* & x_4^* \end{bmatrix} \tag{4.289}$$

Fixed Transmit Spatial Modulators and Receiver Linear Equalization; Transmission systems may better use a fixed transmit modulation matrix A (or choose one from a set of possibilities) and maintain $\gamma_{repeat} = 1$. This is often called a "precoder." More accurately, it is a fixed modulation matrix. The matrix A may approximate the optimum SVD $A \approx M$ matrix, but of course this depends on the channel. A MMSE-LE matrix W is used in the receiver for each block. When the transmit modulator is $A = M$, the unbiased MMSE-LE is $W_{unbiased} = F^*$ so coinciding with vector coding. In practice the approximation may vary. Some systems may agree to select a fixed A matrix from a set of such channel-independent choices and communicate (through a feedback overhead/control channel) the index of the selected A. Choice of a single A, or of a "spatial-precoder codebook" of choices, is essentially an adaptive problem addressed in Chapter 7. Not surprisingly perhaps, good choices of the fixed modulator matrices tend to look like DFT matrices that spread the dimension of any input element v_ℓ equally across all modulator output dimensions $x_n, \ell = 1, ..., L_x$. This also can create a diversity over the spatial dimensionality of the channel. It is not usually water-filling, but can approximate such when water-filling is nearly "on/off" over a best set of input dimensions.

Wi-Fi spatial coding: Wi-Fi systems share the same frequency channel downlink and uplink for a given pair of communicating devices, with one being a controller or "access point" able to communicate on the same frequency channel with many devices and the other being a user and only able to communicate to the access point. In these systems, each transmission consists of possibly many successive codewords transmitted, preceded by a training period where the receiver can learn the channel and possibly communicate an M matrix to the other side. This training involves some exchange at the beginning of the then subsequent data unidirectional transmission. The training repeats on the next use. The M matrix can be approximated by a series of 2×2 elemental unitary transformations, each characterized by an angle (or two angles in complex baseband case). The exchange will send these angles and the elemental indices to which they apply. Often just a few are sufficient to approximate M.

Wi-Fi also optionally allows STBC use, which can be helpful in cases with very low SNR (long distances so the received signal will operate at low data rate). Because the Wi-Fi convolutional (or LDPC) codes are not capacity achieving and a fixed set, the STBC effectively increases SNR appearance through additional redundancy (and consequent data-rate reduction). Wi-Fi then uses the complex Alamouti Code on a 1×2 or 2×2 channel. For a 2×4 channel (2 receive antennas and 4 transmit antennas, likely downlink), the Alamouti Code is independently applied to two spatial streams each on 2 of the input antennas. The receiver will implement a MMSE-LE over all 4 dimensions. For a 2×3 channel, there is one Alamouti use and one transmit antenna uses no spatial code. For a 3×4 channel, the 2×2 double-Alamouti appears on 1 spatial stream, and two other streams are uncoded.

Cellular spatial coding: Cellular spatial coding is more complex than Wi-Fi. Often the number of base-station antennas can be large and arranged in a rectangular array. The transmit matrixes are called precoders. These precoders are learned from embedded repeated training patterns in continuous transmission. Cellular does not usually time-multiplex the downlink and uplink onto a common channel, although some versions do[51]. The precoding matrices construct from the multiplication of factors

$$M = M_{slow} \cdot M_{fast}$$

that correspond to numerous indices corresponding to the number of transmit antennas in the horizontal direction of an antenna panel and the number of antennas in the vertical direction. Often analog inphase-quadrature phase adjustment, called "oversampling" of angle, allow the selection of one antenna phasing from a set of possibilities called a subarray. The first precoder matrix factor M_{slow} arises from a two-dimensional DFT structure with the two dimensions being spatial distance (not time) along the two antenna-panel directions. This factor is unitary and all precoder matrix elements have constant amplitude. The second unitary matrix factor M_{fast} allows different magnitudes and phases on the elements. The first factor is updated less often and corresponds to indices sent from the receiver to the

[51] These systems usually assume reciprocity of the channel, ignoring noise, and assume the other direction's H is the transpose of that just learned locally. In this case, the singular value decomposition is assumed the same in both directions so no exchange for spatial modulation is necessary.

transmitter in a reverse control channel. The second factor also corresponds to indices transmitted for the possible magnitudes (which correspond to simple bit shifts, so multiplication by $1/\sqrt{2}$, $1/\sqrt{4}$, ... , or energy quantization in logarithmic increments 3 dB, as well as DFT-like phases. The second factor is updated more often. Essentially the system thus allows for the indices to specify a large range of possible M matrices. Effectively the index specification tries to provide enough M choices to approximate the SVD matrix. The same type of precoding is used uplink, but with a smaller number of antennas usually. Various training methods attempt to find the permitted matrix closest to M. Chapter 7 addresses more completely training. Repetition of transmission, essentially STBC, is possible as well for situations in which the current coding options may benefit from a lower dimensionality loss and better use of SNR, but of course with consequent significant rate reduction. Again at low SNRs, this can extend range for the minimum data rates.

Type I Cellular Precoders: Figure 4.27 illustrates a cellular antenna panel. Both horizontal and vertical directions appear and the overall number of (used) antennas is $L_x = L_{x,2} \cdot L_{x,1}$. Actual systems offer a delay multiple for an analog delay circuit that is adjacent to the antenna and adjusts its inphase-quadrature phase rotation so that the antenna appears as if in one of the positions in each subarray. Only one antenna in each subarray is used, but the delay offset allows any of the positions to be used. Nominal delay is zero and the darker green color position. A delay of half the propagation time between antennas would produce the lighter green position, and so on. The antennas in either direction have a precoder factor that is the size $L_{x,n} \cdot O_n$ DFT of any $L_{x,n}$-dimensional input. The cartesian product of the two DFTs, following the panel's antenna configuration, then applies to create an $L_x = L_{x,1} \cdot L_{x,2}$ preocder matrix that applies to L_x input scalars. These results each re-stack into $L_x \times 1$ vectors. These vectors then are divided into groups of $L_x/2$ or $L_x/4$ subgroups to which phases of 0°, 180° or $0^0, 90^o, 180^o$, or 270° are applied to complete the precoder matrix. Inputs can be divided into a number of streams that should not exceed L_y (which may be all at one user or more likely across the antennas of many users as in Chapter 2), and the successive stream vectors are processed by the same precoder or by various rotations of these for each stream.

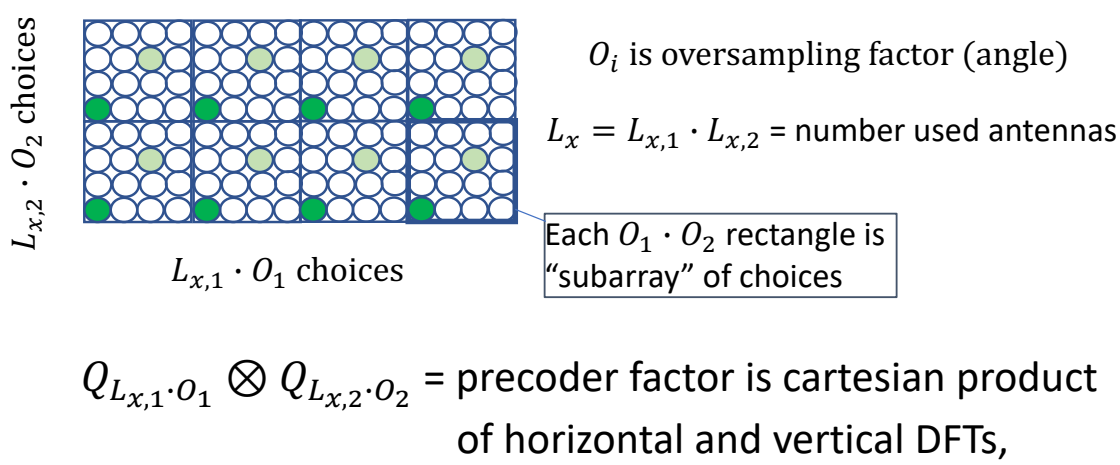


Figure 4.27: Cellular antenna panel

There are some restrictions on sizes and choices of $L_{x,n}, O_n$, but basically these "Type I precoders" offer a wide range of transform choices although the magnitude of each precoder matrix $\widehat{M}$ remain constant and thus may not fully approximate a real SVD choice of M. The choices are exchanged by indices known as PMI's (precoder matrix indices) in cellular.

Type II Cellular Precoders: Type II precoders eventually join cellular's choice of precoders, expanding upon the type I choices. These precoders correspond to a formula that essentially specifies the precoder matrix as a weighted sum :

$$W_{ss} = \sum_{\ell=1}^{L_x} \boldsymbol{w}_{\ell,ss} \cdot a_{\ell,ss} \quad . \tag{4.290}$$

The $\boldsymbol{w}_{\ell,ss}$ are Type I column vectors where ss is the stream index. The scalars $a_{\ell,ss}$ allow non-unit gain adjustment by selection from a set of $2^{-i/2}$ for $i \in \{0, ..., 5\}$ and phase choices $e^{2\pi j/j}$ where $j \in 0, ..., L_x - 1$. This of course allows yet better approximation to the M matrix of SVD. The $a_{\ell,ss}$ factor typically subfactors into two components: The first is a slower-time variation "wideband" factor that is updated less often and largely a function of stationary dynamics; The second factor is a more narrowband factor that may be different for different tones (see Vector DMT/OFDM in Section 4.7) and can be updated more often to fine tune the precoder.

4.7 Discrete Orthogonal Partitioning in Frequency, DMT, OFDM, and Filtered MT

This section details **Discrete MultiTone (DMT)** and **Orthogonal Frequency Division Multiplexing (OFDM)** as two very common forms of Vector Coding that each add the same restriction to reduce implementation complexity. Both DMT and OFDM use the same channel partitioning – they differ in the loading strategy: OFDM puts equal bits and energy on a fixed pre-agreed dimension set, rather than optimizing individual 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 DSL and cable-modem connections. OFDM is also used in bi-directional wireless time-varying channels with minimal feedback of $[|C|\ r]$ on the pre-agreed set of tones as per Section 4.4. Wireless systems often use DMT-like variable energy/bits on spatial dimensions (sets of dimensions called spatial streams) while coded-OFDM over time-frequency, so a hybrid of the two loading strategies. These codes recover dimensions lost to frequency-domain multipath fading. The terms **Vector DMT** and **Vector OFDM** are the MIMO forms of both methods where all spatial dimensions share a common symbol boundary.

With DMT/OFDM partitioning, each of the L_x input dimensions per sampling time has a scalar transfer to each of the L_y output dimensions. The following analysis will apply identically to each of these $L_y \cdot L_x$ scalar channels. For this reason, the introduction of DMT and OFDM will avoid extra MIMO notational burden. Subsection 4.7.3 will revisit the vector-partitioning generalization for $L_x > 1$ or $L_y > 1$ (or both).

Cyclic/Periodic Channels: DMT/OFDM channel partitioning uses Section 4.6's guard period to cause periodic symbol appearance with $x_{-k} = x_{\overline{N}-k}$ for $k = 1, ...\nu$. The repeat of the last ν samples at the symbol beginning is called a **cyclic prefix.**. Large processing simplification occurs with the cyclic prefix. For Subsection 4.7.3's MIMO case, $\boldsymbol{x}_{-k} = \boldsymbol{x}_{\overline{N}-k}$ for $k = 1, ..., \nu$. With cyclic prefix, the matrix channel description H simplifies to a square[52] $\overline{N} \times \overline{N}$ circulant matrix:

$$\begin{bmatrix} y_{\overline{N}-1} \\ y_{\overline{N}-2} \\ \vdots \\ y_0 \end{bmatrix} = \begin{bmatrix} h_0 & h_1 & ... & h_\nu & 0 & ... & 0 \\ 0 & h_0 & \ddots & h_{\nu-1} & h_\nu & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & ... & 0 & h_0 & h_1 & ... & h_\nu \\ h_\nu & 0 & ... & 0 & h_0 & ... & h_{\nu-1} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ h_1 & ... & h_\nu & 0 & ... & 0 & h_0 \end{bmatrix} \begin{bmatrix} x_{\overline{N}-1} \\ \vdots \\ x_0 \end{bmatrix} + \begin{bmatrix} n_{\overline{N}-1} \\ n_{\overline{N}-2} \\ \vdots \\ n_0 \end{bmatrix} \tag{4.291}$$

$$\boldsymbol{y} = \widetilde{H} \cdot \boldsymbol{x} + \boldsymbol{n} \quad . \tag{4.292}$$

Singular value decomposition generally produces unique non-negative real-valued singular values even when H or $\widetilde{H}$ is complex. For transmission, this non-negative-real singular-value restriction is superfluous, and SVD variants are simpler to implement. For (4.291)'s circulant matrix, designs may replace the SVD by the circulant (cyclic) $\widetilde{H}$'s eigendecomposition or "spectral factorization":

$$\widetilde{H} = M \cdot \Lambda \cdot M^* \quad , \tag{4.293}$$

where $M \cdot M^* = M^* \cdot M = I$ and Λ is a square diagonal matrix with possibly complex eigenvalues $\lambda_n,\ n = 0, ..., \overline{N} - 1$ on the diagonal. The magnitude of eigenvalues are equal to the singular values. Further, effectively, SVD's M and F become the same matrix for this special case. Equation (4.293)'s product SNR is the same as for SVD-factorization because the magnitude of each eigenvalue equals a corresponding singular value.[53] This transmission might be called **Eigenvector Coding**, analogous to

[52] This matrix is $\overline{N} \times \overline{N}$ complex $\widetilde{H} \in \mathbb{C}^{\overline{N} \times \overline{N}}$ for complex baseband with $\widetilde{N} = 2$ and $\widetilde{H} \in \mathbb{R}^{N \times N}$ for real channels, but with conjugate symmetric eigenvalues $\lambda_n = \lambda^*_{-n}$ for $n = 1, ..., N/2$.

[53] A caution to the reader using Matlab's eigenvalue/vector computation – Matlab's eigenvalues are usually ordered differntly than Matlab's singular-value decomposition.

modal modulation's eigenfunction use. While this method's dimensional SNRs are the same as vector coding for the cyclic $\widetilde{H}$, there are many choices for the "transform" matrix M in (4.292). Since all have the same performance, the next subsection investigates a very heavily used choice that has two advantages – (1) the matrix M is not a function of the channel, and (2) it is efficiently implemented.

4.7.1 The Discrete Fourier Transform

A DFT review is helpful for this cyclic-prefix-style channel partitioning:

Definition 4.7.1 [Discrete Fourier Transform] *The* **Discrete Fourier Transform (DFT)** *of a $\overline{N}$-dimensional input symbol $\boldsymbol{x}$ is*

$$\boldsymbol{X} \triangleq \begin{bmatrix} X_{\overline{N}-1} \\ X_{\overline{N}-2} \\ \vdots \\ X_0 \end{bmatrix} \tag{4.294}$$

where (the reason for the capital X_n as a "transform" is now apparent)

$$X_n = \frac{1}{\sqrt{N}} \sum_{k=0}^{\overline{N}-1} x_k \cdot e^{-\jmath \frac{2\pi}{\overline{N}} kn} \;\; \forall \;\; n \in [0, \overline{N}-1] \quad , \tag{4.295}$$

and

$$\boldsymbol{x} = \begin{bmatrix} x_{\overline{N}-1} \\ x_{\overline{N}-2} \\ \vdots \\ x_0 \end{bmatrix} \quad . \tag{4.296}$$

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

$$x_k = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} X_n \cdot e^{\jmath \frac{2\pi}{\overline{N}} kn} \;\; \forall \;\; k \in [0, \overline{N}-1] \quad . \tag{4.297}$$

In matrix form, the DFT and IDFT are

$$\boldsymbol{X} = \mathcal{Q} \cdot \boldsymbol{x} \tag{4.298}$$

$$\boldsymbol{x} = \mathcal{Q}^* \cdot \boldsymbol{X} \quad , \tag{4.299}$$

where $\mathcal{Q}$ is the unitary matrix ($Q \cdot Q^* = Q^* \cdot Q = I$)

$$\mathcal{Q} = \frac{1}{\sqrt{\overline{N}}} \begin{bmatrix} e^{-\jmath \frac{2\pi}{\overline{N}}(\overline{N}-1)(\overline{N}-1)} & \dots & e^{-\jmath \frac{2\pi}{\overline{N}} 2(\overline{N}-1)} & e^{-\jmath \frac{2\pi}{\overline{N}}(\overline{N}-1)} & 1 \\ e^{-\jmath \frac{2\pi}{\overline{N}}(\overline{N}-1)\cdot(\overline{N}-2)} & \dots & e^{-\jmath \frac{2\pi}{\overline{N}} 2(\overline{N}-2)} & e^{-\jmath \frac{2\pi}{\overline{N}}(\overline{N}-2)} & 1 \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ e^{-\jmath \frac{2\pi}{\overline{N}}(\overline{N}-1)} & \dots & e^{-\jmath \frac{2\pi}{\overline{N}} 2} & e^{-\jmath \frac{2\pi}{\overline{N}}} & 1 \\ 1 & \dots & 1 & 1 & 1 \end{bmatrix} \quad . \tag{4.300}$$

Both $\boldsymbol{x}$ and $\boldsymbol{X}$ can be represent one period of a periodic sequence that also generates DFT/IDFT values for indices k and/or n outside the interval $(0, \overline{N}-1)$. The matrix $\mathcal{Q}^*$ appears the same as $\mathcal{Q}$ in (4.300) except that the exponents all have a plus $\jmath$ where the minus $\jmath$ now occurs. A further definition is of the vectors $\boldsymbol{q}_n$ according to

$$\mathcal{Q}^* = [\boldsymbol{q}_{\overline{N}-1}, \dots, \boldsymbol{q}_0] \quad , \tag{4.301}$$

where (using $e^{\jmath\frac{2\pi}{\overline{N}}n\cdot(\overline{N}-i)} = e^{-\jmath\frac{2\pi}{\overline{N}}n\cdot i}$)

$$\boldsymbol{q}_n = \begin{bmatrix} 1 \\ e^{-\jmath\frac{2\pi}{\overline{N}}n} \\ \vdots \\ e^{-\jmath\frac{2\pi}{\overline{N}}(\overline{N}-1)n} \end{bmatrix} \tag{4.302}$$

These $\boldsymbol{q}_n$ vectors are useful in the Lemma 4.7.1's proof.

The IDFT, X_n, and DFT, x_k, values can be complex. For real time-domain signals, the restriction $X_{\overline{N}-n} = X_n^*$ for $n = 1, ..., \overline{N}-1$ holds. This restriction implies that there are $\overline{N} = N/2$ complex dimensions when N is even[54]. The IDFT generally corresponds to $\overline{N}$ or N input samples when complex or real respectively. Thus, there is a slight inconsistency in notation. This inconsistency is resolved simply by considering $\overline{N}$ in the IDFT and DFT definitions to be the maximum number of used tones, complex or real. The FFT size should then be equal to the number of real dimensions only in the latter case of real baseband signals, but only half are energized (and the other half have conjugate symmetry to ensure a real signal). With complex baseband signals, 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)
```

Lemma 4.7.1 [DFT is M for circularly prefixed VC] *The circulant matrix $\widetilde{H} = M \cdot \Lambda \cdot M^*$ has a eigendecomposition with $M = \mathcal{Q}^*$ - that is the eigenvectors are the columns of the IDFT matrix and channel independent. Furthermore, the diagonal values of Λ are $\lambda_n = H_n$ - that is, $\widetilde{H}$'s eigenvalues are the DFT values. Note $\lambda_n \in \mathbb{C}$, but $|H_n|$ are equal to the non-negative-real singular values.*

Proof: By direct use and inspection of $\boldsymbol{q}_n$ in the eigenvalue definition, $\widetilde{H}$'s eigenvalues and corresponding eigenvectors follow:

$$\lambda \cdot \boldsymbol{q}_n = \widetilde{H} \cdot \boldsymbol{q}_n \tag{4.303}$$

$$= \begin{bmatrix} h_0 & h_1 & ... & h_\nu & 0 & ... & 0 \\ 0 & h_0 & \ddots & h_{\nu-1} & h_\nu & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ h_1 & ... & h_\nu & 0 & ... & 0 & h_0 \end{bmatrix} \frac{1}{\sqrt{\overline{N}}} \cdot \begin{bmatrix} 1 \\ e^{-\jmath\frac{2\pi}{\overline{N}}n} \\ \vdots \\ e^{-\jmath\frac{2\pi}{\overline{N}}(\overline{N}-1)n} \end{bmatrix} \quad ,(4.304)$$

which has matrix equivalent $\mathcal{Q}^* \cdot \Lambda = \widetilde{H} \cdot \mathcal{Q}^*$. For example, by investigating the top row on right of Equation (4.304),

$$\frac{1}{\sqrt{\overline{N}}} \cdot \left(h_0 + h_1 \cdot e^{-\jmath\frac{2\pi}{\overline{N}}n} + ... + h_{\overline{N}-1} \cdot e^{-\jmath\frac{2\pi(\overline{N}-1)}{\overline{N}}n} \right) \cdot e^{-\jmath\frac{2\pi}{\overline{N}}n} = \widetilde{H}_n \cdot e^{-\jmath\frac{2\pi}{\overline{N}}n} \quad , \tag{4.305}$$

which is the top row of the left side with $\lambda_n = \widetilde{H}_n$. The bottom row is directly $\lambda_n = \widetilde{H}_n$. The next row down from the top on the right is then

$$\frac{1}{\sqrt{N}} \cdot \left(h_0 \cdot e^{-\jmath\frac{2\pi}{\overline{N}}\cdot n} + h_1 \cdot e^{-\jmath\frac{2\pi}{\overline{N}}\cdot 2n} + ... + h_{\overline{N}-1} \right) \cdot e^{-\jmath\frac{2\pi}{\overline{N}}n} = \widetilde{H}_n \cdot \frac{e^{-\jmath\frac{2\pi}{\overline{N}}\cdot 2n}}{\sqrt{\overline{N}}} \quad , \tag{4.306}$$

so again $\lambda_n = \widetilde{H}_n$. Repetition of this procedure for all DFT values of h_0,..., h_ν, produces H_n, $n = 0, ..., \overline{N}-1$ or

$$\Lambda \cdot \boldsymbol{q}_n = \widetilde{H} \cdot \boldsymbol{q}_n \quad , \tag{4.307}$$

[54] 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

making all $\boldsymbol{q}_n$ vectors the eigenvectors of $\widetilde{H}$. Thus, a valid eigenvector set choice is set of the IDFT-matrix$\mathcal{Q}^*$'s columns. For this choice, the eigenvalues are exactly the DFT values of $\widetilde{H}$'s rows. **QED.**

4.7.2 DMT Further Optimizes OFDM

Discrete Multi-Tone (DMT) transmission uses the cyclic prefix and $M = F = Q^*$ in vector coding for the constructed cyclic $\widetilde{H}$. The Q matrix (happily) is therefore channel independent. **Orthogonal Frequency Division Multiplexing (OFDM)** also uses these same vectors (but does not use DMT's optimum loading, and instead sends equal energy and information on a fixed known dimension set).

Complexity: 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 $\overline{N} \cdot log_2(\overline{N})$ operations, which is less than vector coding's the $\overline{N}$ equivalent operations. For large $\overline{N}$, this is a very large computational reduction. Because an additional cyclic-prefix restriction applies only to DMT/OFDM, then DMT/OFDM can only perform as well as or worse than VC without this restriction. When $\overline{N} >> \nu$, the difference will be small simply because the difference in $\widetilde{H}$ is small. The designer should lower DMT's energy constraint when executing loading algorithms by the factor $\overline{N}/(\overline{N}+\nu)$ because of the "wasted energy" in the cyclic prefix, the price for the computational simplification. This wasted-energy loss also goes to zero as $\overline{N}$ becomes infinite.

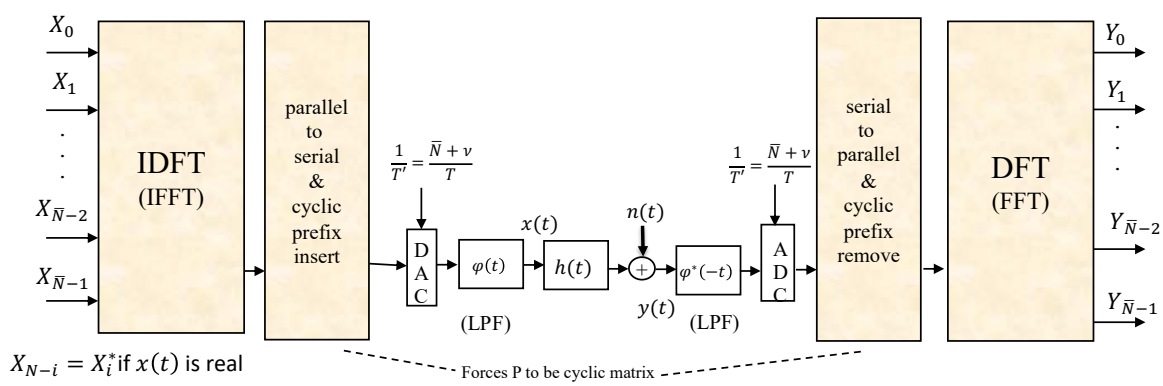


Figure 4.28: DMT and OFDM partitioning block diagram.

Figure 4.28 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 can use computationally efficient FFT algorithms for the DFT and IDFT implementations.

Asymptotic Equivalence: For the case of $\overline{N} \to \infty$, the cyclic prefix becomes negligible and vector coding performs the same as DMT, and all the corresponding individual subchannels of DMT and VC have the same SNR$_n$ as $\overline{N} \to \infty$. 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 modal modulation while DMT is a discrete form of MT. In the limit, with sufficiently high sampling rate, DMT, MT, VC and modal modulation all perform the same. Only DMT is easily implemented. No other channel partitioning can perform better as long as $\overline{N}$ is sufficiently large.

The following list may help summarize Chapter 3's two SNR's of interest, SNR_{MFB} and $SNR_{DMT,u}$:

- $SNR_{DMT,u}$ is not a bound and is attainable, and $SNR_{DMT,u} \leq SNR_{MFB}$.
- outside of the cyclic-prefix loss, which tends to zero for infinite block length, $SNR_{DMT,u}$ is the highest possible SNR that can be achieved, so it is a tighter bound than Chapter 3's SNR_{MFB} that bounds uncoded ISI channels' performance with equalization,

- If $\Gamma = 0$ dB with good codes, then there is no better partitioning system than DMT as the block length becomes large.
- An apparent quandry[55] might be that in Chapter 8, MLSD attains the matched filter bound SNR for various finite-length impulse-response channels, but those situations effectively have a smaller Γ than uncoded and the sequence detector is in fact taking advantage of a code. If a code with zero gap had been consistently used, MLSD for intersymbol interference will not exceed the performance of $SNR_{DMT,u}$ with zero dB gap. (So, the difference is different gaps, not an optimality violation.)
- Thus, as will be discussed in Chapters 10 and beyond, the designer should not confuse a coding gain with a gain caused by channel-ISI partitioning, for which there is no better method than DMT as $\overline{N} \to \infty$.

EXAMPLE 4.7.1 (DMT for $1 + .9 \cdot D^{-1}$ Channel) Continuing the example for the $1 + .9 \cdot D^{-1}$ channel for DMT, the circulant channel matrix for $\overline{N} = 8$ and $\nu = 1$ becomes

$$\widetilde{H} = \begin{bmatrix} .9 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & .9 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & .9 & 1 & 0 & 0 & 0 & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & .9 \end{bmatrix} . \tag{4.308}$$

Table 4.3 summarizes the results of waterfilling with **8** units of energy (not 9, because the cyclic prefix loses 1 energy unit on average). Only 7 of the 8 subchannels were used - note that 6 of the subchannels are effectively 3 two-dimensional QAM subchannels.

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

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

The SNR is

$$\text{SNR}_{DMT} = \left[\prod_{n=0}^{6}(1 + \text{SNR}_n)\right]^{1/9} - 1 = 7.6 \text{ dB} < \ 9.2 \text{ dB from Example 4.1.1.} \tag{4.309}$$

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 $\overline{N} = 16$, this SNR increases to its maximum value of 8.8 dB with far less complexity than VC.

[55] Readers who are not yet familiar with sequence detection (Chapter 9) coding (Chapter 10) can ignore this and the next bullet. Those readers who are familiar will recognize that the channel itself, combined with sequence detection, forms a "code" on the transmitted symbol sequences prior to the addition of WGN. Use of "outer" codes on such a system would thus NOT produce those codes' nominal gain nor gap. MLSD systems with ISI do not follow the gap approximation. DMT systems by contrast can directly use the codes and achieve the full gain and thus maintain the codes' gap.

This example progresses to compare this DMT design's exact complexity and its exact SNR now correctly computed with that of Chapter 3's MMSE-DFE studied earlier and also against the VC system. To get 7.6 dB, Section 3.7's MMSE-DFE required 3 feedforward taps and 1 feedback tap for a complexity of 4 multiply-accumulates 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), higher than the MMSE-DFE. 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 performance level 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, Chapter 3's MMSE-DFE incurs a precoding loss of 1.2 dB. DMT is clearly higher performance and lower complexity in this example (than "single carrier with DFE). This is typical and the reason that almost all high-performance transmission uses DMT or OFDM partitioning (each with codes).

Zero-Stuffed Guard-Period Paradox: Some may observe that the cyclic prefix could be zero and still periodic, with consequently no energy loss. The DFT size increases in this case to $\overline{N}+\nu$ where ν of the included time-domain dimensions must be zero. While this all is astute observation, the restriction also introduces cross-dimensional interference in the frequency domain, essentially the equivalent of windowing the time-domain with a rectangular window (which introduces convolution in the frequency domain, see Section 4.8. This undoes any computational advantage, and thus removes the original complexity advantage of DMT/OFDM.

4.7.3 Vector DMT and OFDM

Application of SISO DMT/OFDM partitioning to every input dimension, the channel input-output relationship on each dimension readily generalizes to

$$\underbrace{\boldsymbol{Y}_n}_{L_y\times 1} = \underbrace{\widetilde{\boldsymbol{H}}_n}_{L_y\times L_x} \cdot \underbrace{\boldsymbol{X}_n}_{L_x\times 1} + \underbrace{\tilde{\boldsymbol{N}}_n}_{L_y\times 1} \quad . \tag{4.310}$$

While the matrices $\widetilde{\boldsymbol{H}}_n$ remain independent over frequency-dimensional index $n = 1, ..., \overline{N}$, crosstalk remains within $\widetilde{\boldsymbol{H}}_n$'s spatial dimensions. An SVD can be implemented on each tone as

$$\widetilde{\boldsymbol{H}}_n = F_n \cdot \boldsymbol{\rho}_n \cdot M_n^* \quad , \tag{4.311}$$

where there will be no more than $L = \min(L_x, L_y)$ nonzero singular values. A complete set of $\overline{N}L$ parallel channels will then have been generated with each tone having input

$$\boldsymbol{X}_n = M_n \cdot \boldsymbol{V}_n \quad , \tag{4.312}$$

where $\boldsymbol{V}_n$ is an $L \times 1$ input n (acts on first L columns of M_n) that has independent data-symbol inputs with energies $\mathcal{E}_{n,\ell}$ on each of its dimensions. The corresponding output (after the receiver FFTs and aggregation of the same tone's L_y dimensions) will be

$$\widehat{\boldsymbol{V}}_n = F_n \cdot \boldsymbol{Y}_n \tag{4.313}$$

The set of parallel channels now will have for each tone (up to) L parallel channels with SNR gains equal to the non-zero singular values of $\boldsymbol{\rho}_n$, or $\lambda_{n,\ell}$. Loading can then occur across the $\overline{N} \cdot L$ total dimensions in use.

Such systems are known as **Vector DMT** with full adaptive loading and **Vector OFDM** when the energy on each dimension is the same for a known dimension set. Unfortunately, the singular values and the matrices M_n and F_n are indeed a function of the channel (unlike the DFT vectors) and will require the transmitter be told what the channel looks like so it can apply the appropriate M_n on each tone. Chapter 7 discusses methods to select an M_n from a set of viable candidates pre-agreed in cellular wireless systems. Wireline systems adaptively learn the values of M_n and F_n (or their equivalents), as also in Chapter 7.

Vector DMT Modulator: Figure 4.29 illustrates the Vector DMT (or Vector OFDM) modulator. There are L_x parallel IDFT's used to modulate the data symbols for each index n value. The system synchronizes symbol-period boundaries and uses the same-length cyclic prefix size (large enough for all the crosstalk channels' intersymbol interference duration). Each stream thus has a cyclic prefix for itself added in the same time position before conversion to analog (and possibly radio-frequency passband-signal conversion with a carrier, as in Chapter 1) and transmission.

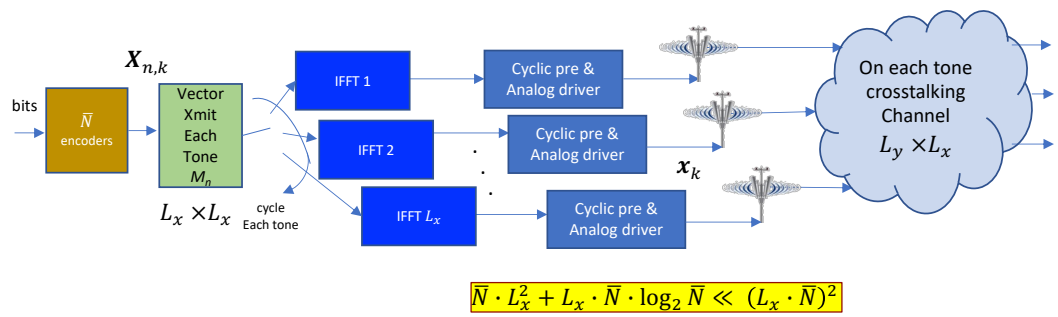


Figure 4.29: Vector DMT/OFDM Modulator Architecture.

Vector DMT Demodulator: Figure 4.30 illustrates the corresponding vector DMT/OFDM demodulator that essentially reverses the modulator operations, but now with L_y FFT's.

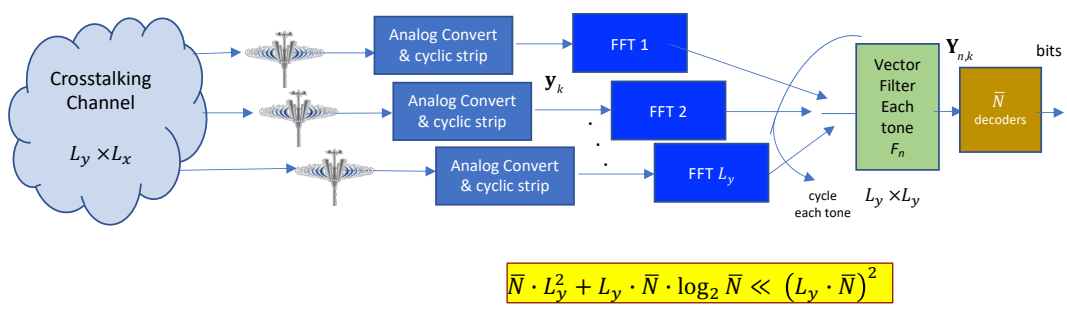


Figure 4.30: Vector DMT/OFDM Receiver Architecture.

By aggregating and aligning the samples for each of the $\overline{N}$ tones, there are $L_y \times L_x$ MIMO channels on each. The system aggregates these independent $L_y \times L_x$ tonal systems with inter-spatial-dimension interference for vectoring coding (which is shown only as vector encode or decode respectively in the two figures). There are LN channels thus created. The complexity is $\overline{N} \cdot L^2 \cdot \log_2(\overline{N})$ computations per symbol instead of $(\overline{N} \cdot L)^2$ if all vector coding were used. If $\overline{N}$ is large, this is a very significant savings in computation. Usually $L << \overline{N}$. (Massive MIMO systems typically do not use all of the available modes, a concept visited in detail in Chapter 5, often these systems choose the modulator matrix M from a known set that is easy to implement and requires less reverse-channel feedback, often themselves a set of matrices circulant across the spatial dimensions).

The loading algorithms for individual antenna-energy constraints proceed exactly as in the vector-coding case with $L_x > 1$. For Vector OFDM systems with constant energy on used input dimensions, the

spatial modulation of Section 4.7 now applies to each tone of the Vector-OFDM (and can apply of course as well to a Vector DMT system). Often some level of energy loading in space applies to Vector OFDM systems so that different spatial streams may have different energy - this is really a hybrid between DMT and Coded-OFDM, but allows aversion of dimensions that should not carry energy because the corresponding singular values for that spatial dimension are too small on average across all the tones.

4.7.3.1 Cyclic-Delay Diversity

Wireless designs sometimes use cyclic-delay diversity. Cyclic diversity corresponds in some ways to STBC and Alamouti codes. In the cyclic-delay case, before the input repeats, it is is cyclically shifted in the time domain (which corresponds to a linearly increasing (or decreasing) delay in frequency across the tones. This has properties comparable to STBC and of course is again, rarely close to the spatial SVD modulator M matrix that is best used.

4.7.4 Matlab DMT loading programs and examples

The program listings appear in Appendix G for the readers reference and potential use. Readers may want also to check the Cioffi Stanford web site https://cioffi-group.stanford.edu for any updated versions of the programs. DMTra.m is a rate-adaptive loading algorithm for a temporal channel pulse response p.

The DMTra.m loading program A DMT-specific rate-adaptive (single-user, temporal channel) loading program is listed in Appendix G, as DMTRA.m.

```
function [gn,en_bar,bn_bar,Nstar,b_bar,SNRdmt]=DMTra(h,NoisePSD,Ex_bar,N,gap)

  CAUTION - the user must know if the input channel p is complex baseband
  or real baseband in interpretting the outputs, as per comments below

  INPUTS
  h - the complex or real baseband sampled (temporal) pulse response
  NoisePSD and Ex_bar - noise power and energy/dimension or cpx-sample.
     These two parameters need to be consistent in terms of # dimensions.
  N is the DFT size, N>2, and is "Nbar" for complex channels and N
      for real channels
  The sampling rate (on each of Inphase or Quad for complex) is (N+v)/T
     where 1/T is DMT symbol rate and nu = length(p) - 1
  gap is the gap in dB

  OUTPUTS
  gn is vector of channel gains or magnitude-squared |P|^2 / NoisePSD
  En - vector of energy allocation.
       Per real dimension for real channels;
       Per tone for complex channel.
     For real channels, the upper image frequencies duplicate the lower
       frequencies.
     For complex channels, each En output is per that tone (= cmplx dim)
  bn_bar is the vector of bit/dim for all subchannels
     For complex channels, double bn_bar for bits/tone.
     For real channels, the upper image frequencies duplicate lower
       frequencies, but bn_bar remains bits per real dimension.
  Nstar is the number of used input-size DFT (real or complex) dimensions
      (2*Nstar is number of real dimensions for complex chan)
  b_bar is the number of bits per (real) dimension
      (so 2*(N+v)*b_bar is total number of bits/symbol for a complex chan )
```

```
  SNRdmt is the equivalent DMT "geometric" SNR (complex or real) in dB
```

A DMT-specific margin-adaptive (single-user, temporal channel) loading program is listed in Appendix G, as DMTma.m.

```
function [gn,en,bn,Nstar,b_bar_check,margin]=DMTma(h,NoisePSD,Ex_bar,b_bar,N,gap)

  Margin Adaptive DMT Loading

  CAUTION - when using this program, you should know if your input channel
  is h is complex baseband or real in interpretting the outputs as per
  comments below.

  The sampling rate (on each of Inphase or Quad for complex) is (N+v)/T
     where 1/T is DMT symbol rate

  INPUTS
  h - the complex or real baseband sampled pulse response;
      nu=length(P) -1
  NoisePSD and Ex_bar - noise power and energy/dimension or cpx-sample.
     These two parameters need to be consistent in terms of # dimensions.
  b_bar is the target normalized bit rate in bits/real dimension (so 1/2
      sum of bits/tone for complex P)
  N is the DFT size, N>2, and is Nbar from notes for complex channels and N
      for real channels
  gap is the gap in dB

  OUTPUTS
  gn is vector of channel gains or magnitude-squared |P|^2 / NoisePSD
  en - vector of energy allocation.
       Per real dimension for real channels;
       Per tone for complex channel.
     For real channels, the upper image frequencies duplicate the lower
       frequencies.
     For complex channels, each En output is per that tone (= cmplx dim)
  bn is the vector of bit/dim for all subchannels
     For complex channels, double bn_bar for bits/tone.
     For real channels, the upper image frequencies duplicate lower
       frequencies, but bn_bar remains bits per real dimension.
  Nstar is the number of used input-size DFT (real or complex) dimensions
      (2*Nstar is number of real dimensions for complex chan)
  b_bar is the number of bits per (real) dimension
      (so 2*(N+v)*b_bar is total number of bits/symbol for a complex chan )
  SNRdmt is the equivalent DMT "geometric" SNR (complex or real) in dB


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


  Levin Campello's Method with DMT - REAL BASEBAND ONLY, allows PAM on
  Nyquist and DC and QAM on others.  Beta is forced to 1.

  Inputs
  h is the pulse response(an nu is set to length(p) - 1
  NoisePSD is the noise PSD on same scale as Ex_bar
```

```
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 (DC to Nyquist)
En is the vector energy distribution from DC to Nyquist
bn is the vector bit distribution from DC to Nyquist
b_bar is the number of bits per dimension in the DMT symbol

The first and last bins are PAM; the rest are QAM.


function [gn,En,bn,b_bar_check,margin]=DMTLCma(h,NoisePSD,Ex_bar,b_bar,N,gap)

Levin Campello's Method for margin-adaptive DMT - UNEQUAL MOD, REAL BB ONLY

h is the sampled pulse response (nu = length(p) - 1)
NoisePSD is noise PSD on same scale as Ex_bar
Ex_bar is the normalized energy
N is the total number of real/complex subchannels, N>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 in dB

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

Example uses of DMTra

Immediately below are the matlab commands to run the usual $1 + .9 \cdot D^{-1}$ example channel with noise per dimension $\frac{\mathcal{N}_0}{2} = .181$. First using rate-adaptive loading with $N = 8, 16$, and 1024. The reader should note the symmetry about the "Nyquist" or middle frequency in these examples. If the channel is real, this symmetry can essentially be ignored and the values up to Nyquist specify the used-tone values. However, if this channel is a basband equivalent for a carrier-modulated passband channel, then the subchannels or "tones" just happen to have the symmetry because the channel is coincidentally all real. A complex channel with possibly complex channel coefficients would not (usually) have such symmetry. ;

$1 + .9 \cdot D^{-1}$ Example Repeated with Matlab, $\frac{\mathcal{N}_0}{2} = .181$, $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$, $N = 8$, and 0 dB coding gain.

```
>> [gn,en_bar,bn_bar,Nstar,bbar,SNRdmt]=DMTra([.9 1],.181,1,8,0)

gn = 19.9448   17.0320   10.0000    2.9680    0.0552    2.9680   10.0000   17.0320

en_bar = 1.2415    1.2329    1.1916    0.9547         0    0.9547    1.1916    1.2329

bn_bar = 2.3436    2.2297    1.8456    0.9693         0    0.9693    1.8456    2.2297
```

```
Nstar =    7

bbar = 1.3814 bits/dim

SNRdmt =  7.6247 dB
```

$1 + .9 \cdot D^{-1}$ Example Extended $N = 16$, $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$.

```
>> [gn,en_bar,bn_bar,Nstar,bbar,SNRdmt]=DMTra([.9 1],.181,1,16,0)

gn =

  Columns 1 through 9
   19.9448   19.1878   17.0320   13.8057   10.0000    6.1943    2.9680    0.8122    0.0552
  Columns 10 through 16
    0.8122    2.9680    6.1943   10.0000   13.8057   17.0320   19.1878

en_bar =
  Columns 1 through 9
    1.2882    1.2863    1.2797    1.2659    1.2384    1.1769    1.0015    0.1072         0
  Columns 10 through 16
    0.1072    1.0015    1.1769    1.2384    1.2659    1.2797    1.2863

bn_bar =
  Columns 1 through 9
    2.3692    2.3413    2.2553    2.1038    1.8712    1.5257    0.9950    0.0602         0
  Columns 10 through 16
    0.0602    0.9950    1.5257    1.8712    2.1038    2.2553    2.3413

Nstar =  15
bbar =    1.4514
SNRdmt =  8.1152 dB
```

$1 + .9 \cdot D^{-1}$ Example Extended $N = 1024$, $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$.

```
>> [gn,en_bar,bn_bar,Nstar,bbar,SNRdmt]=DMTra([.9 1],.181,1,1024,0);

bbar =        1.54
SNRdmt =  8.7437 dB
```

Complex Channels: The programs also work for complex baseband channels (read the comments), but the user must know they are using complex channels then to adjust the output as per the commands run after the program. Note the Nyquist symmetry in output results is typically lost.

$1 + .9 \cdot D^{-1}$ Example Extended with $N = 8$ and $\bar{\mathcal{E}}_{\boldsymbol{x}} = 2$ now as a complex baseband-equivalent channel

```
>> [gn,en_bar,bn_bar,Nstar,bbar,SNRdmt]=DMTra([.9 1],.181,2,8,0)

gn = 19.9448   17.0320   10.0000    2.9680    0.0552    2.9680   10.0000   17.0320
en_bar =  2.3843    2.3758    2.3345    2.0976         0    2.0976    2.3345    2.3758
bn_bar =  2.8008    2.6869    2.3028    1.4266         0    1.4266    2.3028    2.6869

Nstar =    7
bbar =    1.7370
SNRdmt = 10.0484 dB
```

$1 + .9\jmath \cdot D^{-1}$ **Complex-Channel Example** $N = 8$, $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$.

```
>> [gn,en_bar,bn_bar,Nstar,bbar,SNRdmt]=DMTra([.9i 1],.181,1,8,0)
gn = 10.0000   17.0320   19.9448   17.0320   10.0000    2.9680    0.0552    2.9680
en_bar = 1.1916    1.2329    1.2415    1.2329    1.1916    0.9547         0    0.9547
bn_bar = 1.8456    2.2297    2.3436    2.2297    1.8456    0.9693         0    0.9693

Nstar =      7
bbar =        1.3814
SNRdmt =  7.6247 dB

bn=2*bn_bar = 3.6911    4.4594    4.6871    4.4594    3.6911    1.9387         0    1.9387
en=2*en_bar = 2.3833    2.4658    2.4830    2.4658    2.3833    1.9094        0    1.9094
bits_symbol = 2*9*bbar = 24.8655
Nreal=2*8 = 16
```

$1 + .9 + .4\jmath \cdot D^{-1}$ **Complex-Channel Example** $N = 8$, $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$.

```
>> [gn,en_bar,bn_bar,Nstar,bbar,SNRdmt]=DMTra([.4i+.9 1],.181,1,8,0)

gn = 20.8287   21.0413   15.3039    6.9773    0.9392    0.7266    6.4641   14.7906
en_bar = 1.3222    1.3226    1.3048    1.2269    0.3055         0    1.2155    1.3026
bn_bar = 2.4174    2.4248    2.1951    1.6285    0.1820         0    1.5734    2.1705

Nstar =      7
bbar =        1.3991
SNRdmt =  7.7491 dB

bn=2*bn_bar = 4.8349    4.8495    4.3902    3.2570    0.3639         0    3.1468    4.3410
en=2*en_bar = 2.6443    2.6453    2.6097    2.4537    0.6109         0    2.4309    2.6051
bits_symbol = 2*9*bbar = 25.1833
Nreal=2*8 = 16
```

Some margin adaptive examples are:

$1 + .9 \cdot D^{-1}$ **Margin Adaptive Example** $N = 8$, $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$.

```
[gn,en_bar,bn_bar,Nstar,bbar,margin]= DMTma([.9 1],.181,1,1,8,0)

gn =      19.9448   17.0320   10.0000    2.9680    0.0552    2.9680   10.0000   17.0320
en_bar = 0.6043    0.5958      0.5545    0.3175         0        0.3175     0.5545     0.5958
bn_bar = 1.8532    1.7393      1.3552    0.4790         0        0.4790     1.3552     1.7393
Nstar =       7
bbar =        1
margin =    3.5410 dB
```

$1 + .9 \cdot D^{-1}$ Margin Adaptive Example $N = 1024$, $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$.

```
>> [gn,en_bar,bn_bar,Nstar,bbar,margin]= DMTma([.9 1],.181,1,1,1024,0);

>> margin = 4.7267 dB
```

1+.9+D^{-1} Margin Adaptive Example $N = 8$, $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$, $\Gamma = 8.8$ dB.

```
>> [gn,en_bar,bn_bar,Nstar,bbar,margin]= DMTma([.9 1],.181,1,1,8,8.8)

gn =      19.9448   17.0320   10.0000    2.9680    0.0552    2.9680   10.0000   17.0320
en_bar = 4.5844    4.5193    4.2061    2.4088         0    2.4088    4.2061    4.5193
bn_bar = 1.8532    1.7393    1.3552    0.4790         0    0.4790    1.3552    1.7393
Nstar =       7
bbar =        1.0
margin =  -5.2590 dB
```

4.7.4.1 Discrete-Loading Matlab programs

Similarly, Section 4.3's LC program can be converted for DMT with generalization also to complex channels. The real baseband case has the unusual situation of having the DC and Nyquist subchannels have integer bits in PAM, and the rest have integer bits in QAM. This amounts to different modulation on different tones. Thus, there are really 4 programs listed here - the first two called DMTLCra and DMTLCma work only for baseband and this different modulation case. The second two called EqualDMTLCra and EqualDMTLCra allow only the same granularity "beta" on all subchannels, and are thus for passband systems. The program listings are in Appendix G.

Example Use of DMTLCra

Here is an example of the use of DMTLCra on the real baseband 1+.9D channel

```
>> [gn,En,bn,b_bar,SNRdmt]=DMTLCra([.9 1],.181,1,8,0)
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 =      1.3333
SNRdmt = 7.2832 dB
```

This example has unequal loading on DC versus other used complex dimensions.

Example Use of DMTLCma (baseband real channel)

Here is an example of the use of this unequal loading LC program:

```
>> [gn,En,bn,b_bar,margin]=DMTLCma([.9 1],.181,1,1,8,0)

gn = 19.9448   17.0320   10.0000    2.9680    0.0552
En = 0.7521    0.8220    1.4000    0.6739         0

bn = 2     3     3     1     0
b_bar = 1
margin = 3.4105 dB
```

A complex use example of Equal-loading programs (again listed in Appendix G) is:

```
>> [gn,En,bn,b_bar,SNRdmt]=EqualDMTLCra([.9*i 1],.181,2,8,0,0.5)

gn = 10.0000   17.0320   19.9448   17.0320   10.0000    2.9680    0.0552    2.9680
En      3.0000    1.7614    1.5042    1.7614    3.0000    2.0216         0    2.0216

bn =   4     4     4     4     4     2     0     2
b_bar =   1.3333
SNRdmt = 7.4625 dB
```

A complex use example for margin adaptive is:

```
>>  [gn,En,bn,b_bar,margin]=EqualDMTLCma([-.9*i 1],.181,1,1,8,0,0.5)

gn =  10.0000    2.9680    0.0552    2.9680   10.0000   17.0320   19.9448   17.0320
En =   0.6000         0         0         0    0.2000    0.3523    0.3008    0.3523

bn =     2     0     0     0     1     2     2     2
b_bar =    1
margin =   6.4652 dB
```

4.7.4.2 Wireline Examples

EXAMPLE 4.7.2 [DSLs] DSL today provides internet service on telephone lines. It is also the first wide commercial use of DMT (or C-OFDM) technology and uses dynamic loading. xDSL uses larger bandwidths on shorter lines, as in Figure 4.31. The xDSL internet connection from the telco, or more generally internet service provider, can be entirely copper twisted pair or some combination of fiber and twisted pair with the conversion device/location known as the Optical Network Unit (ONU) or the Optical Line Terminal (OLT) or sometimes just LT for short[56]. Shorter copper has less attenuation and so dynamic loading uses more dimensions. xDSL systems use DMT to get the best performance from the remaining run of copper twisted pair into the residence.

[56]Installing fiber all the way to the residence remains very expensive in most of the world, particularly the last segment where the high cost of labor, permits, digging is shared by fewer and fewer customers. Often telephone companies use the term "homes passed" by fiber - what this really means is Figure 4.31 with a copper line from the OLT to the home, the "passed" means there is not yet fiber to the home, but fiber closer to it

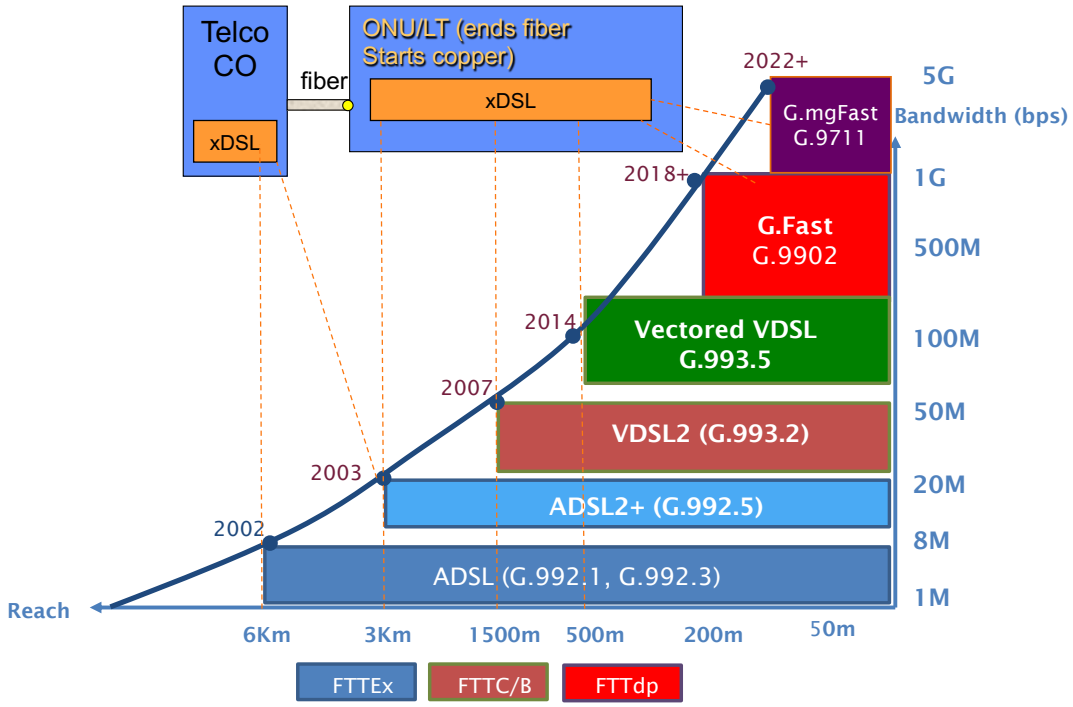


Figure 4.31: xDSL use example.

Sharing directions: The xDSL's typically have to-the customer or **downstream** data rates roughly about 2x to 8x the **upstream** rate from the customer to the telephone company. However, modern DSLs will allow for complete symmetry and use the same data rates in both directions. VDSL separates up from down by frequency bands (up to 8 of them) while G.fast/mgfast separates up and down in a time-domain (think "ping - pong") fashion. Later subsections discuss duplexing of bi-directional transmission.

Table 4.4 below summarizes the various DMT basic parameters[57] that each DSL type uses. (Δf abbreviates $\frac{1}{T} \cdot (1 + \frac{\nu}{N})$). G.fast and G.mgfast use time-domain duplexing[58]

[57] It should be noted that ADSL's insert a dummy (known or "synch") symbol every 68 symbols. To keep the symbol rate exactly 4 kHz (or 48 kHz for the G.fast/mgfast), so that data rates are exact integer multiples of 4 kbps, the cyclic prefix is shortened to 32 samples with the extra 8 reserved (robbed) for construction of the dummy synch symbol. Thus, the actual symbol length is 512+32 or 544 samples with the "robbed" 68(8) = 544 samples then used to construct the synch symbol. In early DSL and digital networks, the synchronization to an 8 kHz network clock was thereby preserved. The pattern of 68 successive symbols is known as a "superframe." As the global networks converted to packets, particularly to IP networks, this synchronization was no longer as necessary, and thus VDSL actually inserts a synch symbol as the 257 symbol in a sequence of symbols or the VDSL superframe. The VDSL numbers in Table 4.4 are correct but the data rates are thus actually integer multiples of (256/257) 4 kHz.

[58] .This introduces frames of 36 successive symbols, some are assigned to upstream and some to downstream with two silent periods that add to one symbol used for reversals of direction in each such frame, plus another remote-management-channel (RMC) symbol that only carries overhead information in half its content (one RMC up and one RMC down) for another full symbol of total data rate loss. There are 8 such frames per superframe, which superframe adds another 2 synch symbols (one for up direction and one for down direction). The superframe of 288 symbols thus has 18 that do not carry any user data. Thus, G.fast data rates are actually multiples of 48 KHz (15/16) - note 15/16 is the same as (270/288) 48 kHz.

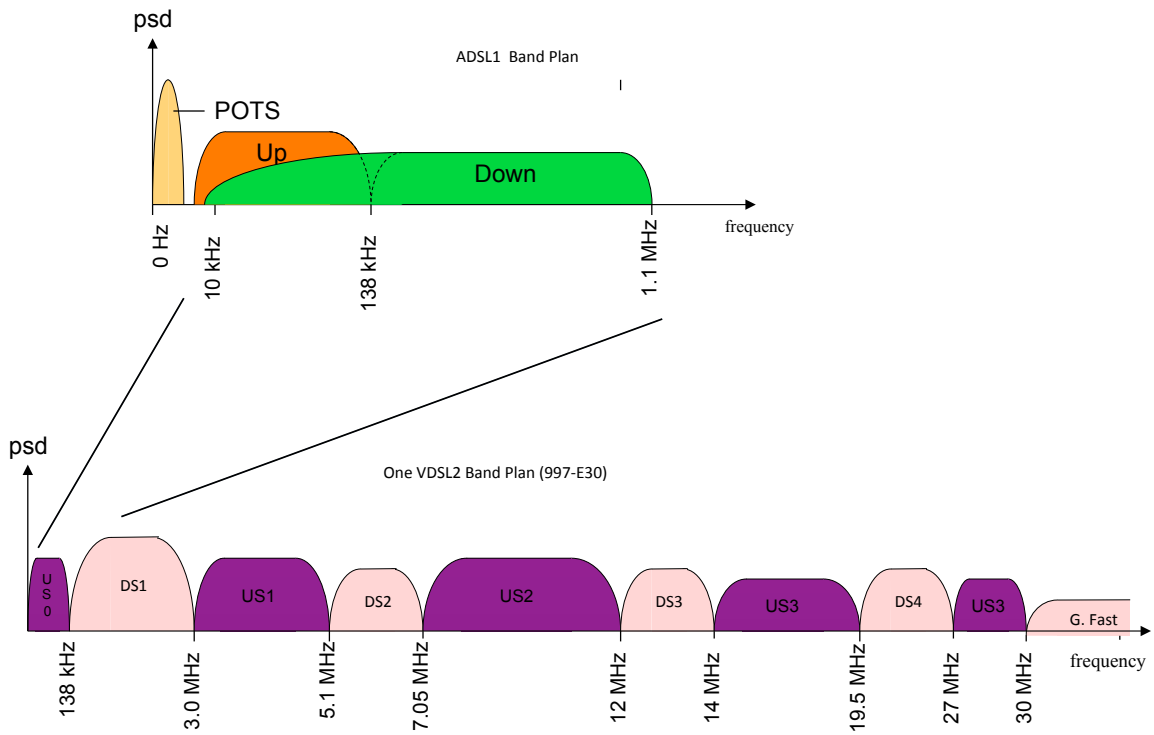


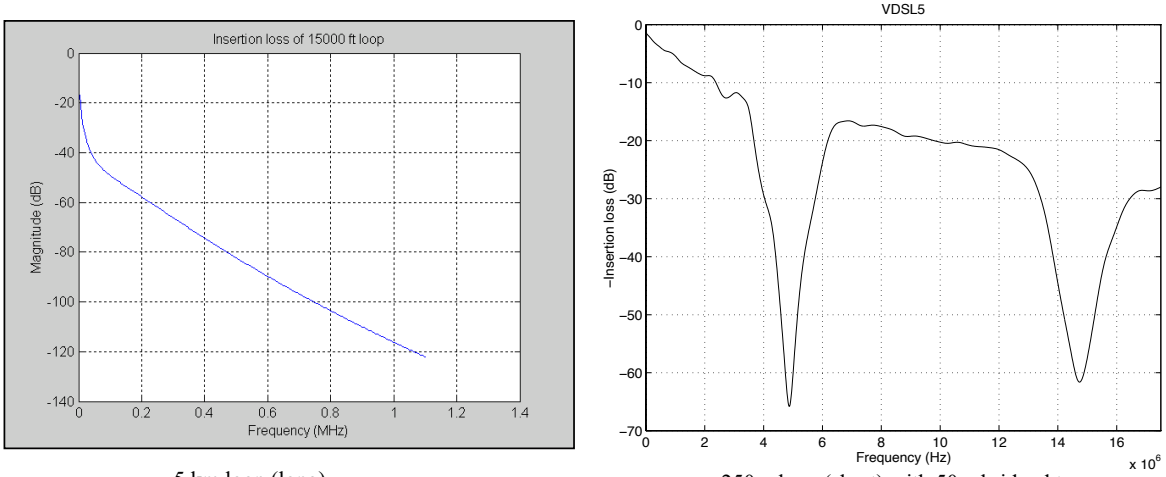
Figure 4.32: xDSL spectrum use - further adaption by loading within each band.

ITU std	Name	$1/T$ (kHz)	N	ν	Δf (khz)	$1/T'$ (MHz)	P_x (dBm)	b_{max}	R_{max} (Mbps)
G.992.1	ADSL1 ds	4	512	40	4.3125	2.208	20.5	15	8
	ADSL1 us	4	64	5	4.3125	0.276	14.5	15	0.8
G.992.5	ADSL2+ ds	4	1024	80	4.3125	4.416	20.5	15	24
	ADSL2+ us	4	128	10	4.3125	0.552	14.5	15	1.5
G.993.2	VDSL2	4	4096	320	4.3125	35.328	14.5	15	150
G.993.5	Vectored VDSL2	4	8192	640	4.3125	70.656	14.5	15	300
G.9900-02	G.fast	48	8192	640	51.75	416	8	14	1000
?	XG.fast	192	8192	640	207	1,696	8	15	10,000

Table 4.4: Table of xDSL basic modulation parameters.

Figure 4.32 shows the original upstream and downstream bands for the original ADSL1. Some overlap was permitted at low frequencies where echo cancellation was used to separate the different directions originally and crosstalk was not too high at these lower frequencies. However, the overlap later gave way to no overlap and a large number of similar VDSL band plans with up to 9 different up/down bands as shown in the lower portion of Figure 4.32 for one European plan known as 997-E30. G.fast (shown) uses frequencies (typically) above the local highest VDSL frequencies and uses time-domain duplexing to separate down and up. Some typical ADSL and VDSL channels (taken from the American National Standards Institute) have transfer functions shown in Figures 4.33.

Channel Variation: The lines have significant variation with length, and also with impairments like bridged-taps (open circuited extension phones) that cause the notch in the characteristics in the second graph in Figures 4.32 and 4.33. Also noises are highly frequency selective and could be nearby AM or Amateur radio stations captured by the phone-line or could also contain crosstalk from other DSLs with varying bandwidths. All this channel vari-



5 km loop (long)

250m loop (short) with 50m bridged tap

Figure 4.33: xDSL example lines, long and short with bridged tap.

ation among possible lines for deployment of xDSL creates a huge need for adaptive loading, which will optimize the DMT performance for each situation.

xDSL Loading: xDSL systems' loading is complicated by the presences of two (interleaved) codes. The inner code is Chapter 9's 4D Wei 16-state Trellis code that adds 1 redundant bit per every 4 dimensions. Loading algorithms best use 4D incremental energy tables for this code for pairs of tones that have similar (or equal gains). The receiver modem can specify a "tone-ordering" during loading to facilitate this loading (as well as for other performance reasons having to do with narrow band intermittent interferences that may be learned by the receiver). This particular code has an incremental energy table that monotonically increases with each added bit. In terms of the outer Reed-Solomon (probably hard-decoded) system, the receiver may select the number of parity bytes (it is a GF256 code so byte-wise information and byte-wise parity) along with interleaving depth (up to 20 ms). The overhead amount will be known before loading. However, the extra bits/bytes will affect the gap, and thus the "stopping" criterion for the total number of bits when the energy constraint is met or dually the FM-allowed (or minimized with MA) energy when the information bit-rate (which corresponds to a known number of parity bits) is attained. H. Levin endeavored to simulate a table for "gross" gain of RS codes for different levels of parity, or effectively different code rates $0 < r < 1$. The amounts shown can be used to reduce the gap on all calculations with good accuracy. Some of the gross coding gains are greater than the 10^{-7} gap of 9.5 dB. This does not imply a capacity violation because the code rate reduces the data rate that would be calculated from direction application of the gap formulate –

$$\bar{b}_{info} = \frac{r}{2}\log_2\left(1 + \frac{\mathrm{SNR}_d mt}{\Gamma_{gross}}\right) \quad . \tag{4.314}$$

The code rate is well below unity for those situations where the gross gain is very high.

Gross Coding Gain at 1e-7 probability of error for GF(256) RS codes (hard decoding) – in dB

Block/Parity	2	4	6	8	10	12	14	16
20	2.76	4.54	5.85	6.88	7.73	8.46	9.09	9.65
40	2.62	4.35	5.64	6.65	7.49	8.21	8.83	9.38
70	2.51	4.20	5.46	6.47	7.30	8.01	8.63	9.18
110	2.42	4.08	5.32	6.32	7.14	7.85	8.46	9.01
180	2.32	3.94	5.17	6.15	6.97	7.67	8.28	8.82
255	2.25	3.84	5.05	6.03	6.84	7.54	8.15	8.69

Table 4.5: Levin's Gross Coding Gains for byte-wise Reed Solomon Codes

Cable Broadband Figure 4.34 illustrates a broadcast coaxial cable-TV system. While originally used for unidirectional downstream television transmission, these systems eventually evolved to support broadband access. A single cable connects many homes, possibly with (unidirectional originally) amplifiers and a common signal. The conversion of this system to broadband replaces the amplifiers with bidirectional amplifiers and separates the downstream and upstream transmissions by frequency band.

EXAMPLE 4.7.3 *[Cablemodem DOCSIS 3.1]*

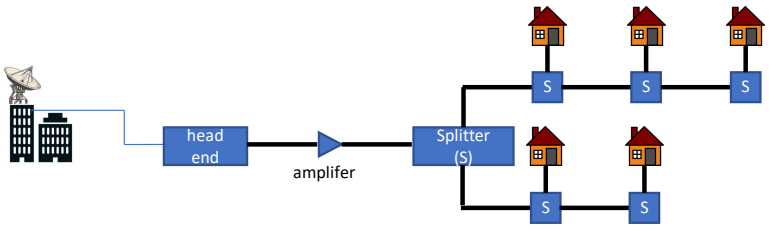


Figure 4.34: Illustration of a broadcast cable neighborhood.

The cable broadband specification is known as DOCSIS (Data over Cable Service Interface Specification). The original TV channels were nominally 6 MHz wide in USA and 8 MHz in much of the rest of the world. Single-carrier QAM transmission (with Chapter 3's equalization and some transmit "pre-equalization" was originally used (downstream) in one or more of the TV channels where transmission could be digital TV or simply broadband internet data. Uplink signals traversed the cable below 40 MHz, because the lowest TV channel had its band-edge near 50 MHz, allowing for some separation of upstream and downstream. Various layer 2 protocols allowed separation of different users' signals digitally. Upstream symbol rates of 1-2 MHz and QPSK allowed 2-4 Mbps. As the industry progressed and data needs shifted from broadcast TV to individual user data streams, wider bandwidths better use

the channel but then incur significant ISI and equalization is no longer sufficient. John Chapman[59] proposed and guided DOCSIS 3.1 (a third-generation cable standard) DMT/C-OFDM use. Figure 4.35 from the DOCSIS standard appears here also and shows the bit loading, IFFT, etc. The bit loading only has 16 choices (profiles) possible (which can be set by the system) and then homes are grouped so that various subsets use a profile.

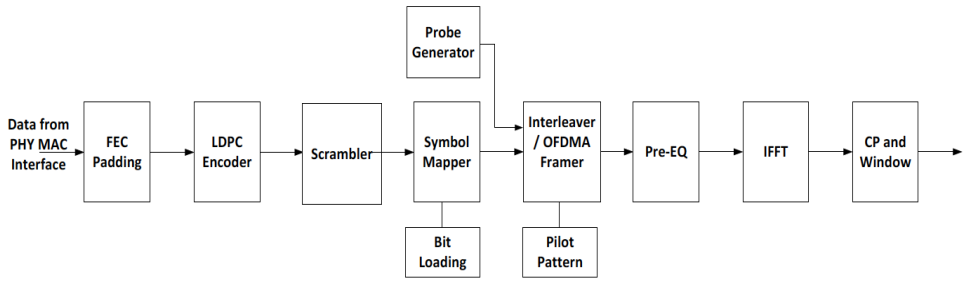


Figure 4.35: Basic DOCSIS 3.1 architecture

[59]Cisco CTO of Broadband Technologies introduced DMT, sometimes called OFDM in standards as it is somewhat of a hybrid in that one of 16 bit-distribution profiles must be selected, for DOCSIS 3.1 over wider bandwidths than one TV channel to increase significantly the data rates for broadband cable service roughly in 2011-12 time frame.

Parameter	4K mode	8K mode
Downstream master clock frequency	10.24 MHz	
Downstream Sampling Rate (fs)	204.8 MHz	
Downstream Elementary Period (Tsd)	1/(204.8 MHz)	
Channel bandwidths	24 MHz … 192 MHz	
IDFT size	4096	8192
Subcarrier spacing	50 kHz	25 kHz
FFT duration (Useful symbol duration) (Tu)	20 µs	40 µs
Maximum number of active subcarriers in signal (192 MHz channel) Values refer to 190 MHz of used subcarriers.	3800	7600
Maximum spacing between first and last active subcarrier	190 MHz	

Parameter	4K mode	8K mode
Cyclic Prefix	0.9375 µs (192 * T_{sd}) 1.25 µs (256 * T_{sd}) 2.5 µs (512 * T_{sd}) 3.75 µs (768 * T_{sd}) 5 µs (1024 * T_{sd})	
Windowing	Tukey raised cosine window, embedded into cyclic prefix 0 µs (0 * T_{sd}) 0.3125 µs (64 * T_{sd}) 0.625 µs (128 * T_{sd}) 0.9375 µs (192 * T_{sd}) 1.25 µs (256 * T_{sd})	

Table 4.6: Basic DOCSIS 3.1 downstream transmission parameters

The widest (DOCSIS3.1) bandwidth used is 192 MHz[60]. There may be several 192 MHz channels used, but each has a DMT system within it. Tables 4.6 and 4.7 enumerates some of the basic DOCSIS 3.1 DMT parameters for use downstream and upstream respectively. The sampling period is Tds in Table 4.6 (same as T' in this text). The symbol period in samples is equal to the sum of the IFFT size and system's choice of cyclic prefix in Table 4.6. The constellations allowed use $b \in \{2, 4, 6, 8, 10, 12\}$ bits.

[60]This is either 24 8-MHz side channels or 32 6-MHz-wide channels aggregated into a single channel.

Parameter	2k Mode	4k Mode
Upstream Sampling Rate (f_{su})	102.4 MHz	
Upstream Elementary Period Rate (T_{su})	1/102.4 MHz	
Channel bandwidths	10 MHz, …, 96 MHz	6.4 MHz, …, 96 MHz
IDFT size (depending on channel bandwidth)	2048	4096
Subcarrier spacing	50 kHz	25 kHz
FFT duration (Useful symbol duration) (T_u)	20 µs	40 µs
Maximum number of active subcarriers in signal (96 MHz channel) Values refer to 95 MHz of active subcarriers	1900	3800
Upstream Cyclic Prefix	0.9375 µs (96 * T_{su}) 1.25 µs (128 * T_{su}) 1.5625 µs (160 * T_{su}) 1.875 µs (192 * T_{su}) 2.1875 µs (224 * T_{su}) 2.5 µs (256 * T_{su}) 2.8125 µs (288 * T_{su}) 3.125 µs (320 * T_{su}) 3.75 µs (384 * T_{su}) 5.0 µs (512 *Tsu) 6.25 µs (640 *Tsu)	

Table 4.7: Basic DOCSIS 3.1 upstream transmission parameters

Table 4.8 lists the LDPC codes parameters used by DOCIS.

Code	Code rate	Codeword size in bits (N_i)	Information bits (K_i)	Parity bits (P_i)
Long code	89% (8/9)	16200	14400	1800
Medium code	85% (28/33)	5940	5040	900
Short code	75% (3/4)	1120	840	280

Table 4.8: Basic DOCSIS 3.1 LDPC Code Parameters

Newer DOCSIS 4.0 systems allow flexible assignment of the up/down cut-off frequency as well.

4.7.4.3 Wireless Examples

Wireless transmission often uses C-OFDM transmission because little transmit optimization of b_n is possible if the channel varies too rapidly with time, or if the channel is unidirectional (for instance a broadcast signal in radio or television). Instead, over a wide bandwidth in wireless transmission, multipath ISI typically results in transmission-band notches. Consequently, part of the signal (some dimensions or tones) is (are) lost. C-OFDM's codes recover these lost dimensions' information if the code is sufficiently powerful, no matter where those lost tones are located (that is, no matter which tone-index set $\{n\}$ is affected). Also, in broadcast wireless transmission, there may be several different "channels," differing with customer location, for the same service. Thus the b_n can not be optimized as a function of any one identified channel (which necessary must occur at the receiver, which is the only place the channel can be known practically and estimated.

C-OFDM's greatest asset is the practical advantage of much less (or no) feedback for loading (an IFFT operation with fixed bit loading and energy on a pre-agreed fixed tone set) – C-OFDM receivers

estimate each dimension's gain and phase, but only for receiver use. Feedback information often is simply $[|C|\ r]$.

EXAMPLE 4.7.4 [Wi-Fi – IEEE802.11a, 11g, 11n, 11ac, and 11ax Wireless LAN]

The IEEE 802.11 standard[61] is "Wi-Fi." Wi-Fi uses unlicensed frequency bands (usually). The original 802.11a and 802.11g methods transmited up to 54 Mbps between wireless users in relative close proximity (nominally, within a building or complex 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 $\overline{N} = 64$ (so up to 128 real dimensions). Only one direction of transmission occurs at any point in time between the Wi-Fi access point and a Wi-Fi device/station. The symbol rate is $1/T' = 20$ MHz and thus $1/T = 250$ kHz with a cyclic prefix of $\nu = 16$ complex samples (or 800 ns). $\overline{N} + \nu = 80$ complex samples. The Δf is then $(1 + \nu/N) \cdot 250 = 312.5$ kHz.

Carrier Frequencies - Passband Complex: Figure 4.36 shows the lowest Wi-Fi carrier frequencies in use as $2407 + i \cdot 5$ MHz, where i is the channel number from 1 to 11. Channels $i = 1, 6$, and 11 find most common use; These 3 channels uniquely do not overlap, so there really only 3 active Wi-Fi channels in the 2.4 GHz frequency band, as in Figure 4.36. More recent Wi-Fi also uses these same carriers with Vector C-OFDM with $L_x \leq 4$ and $L_y \leq 4$. More recent Wi-Fi expands carrier-frequency choices to the 5 GHz range and are exactly (in MHz) $5180 + (i - 36) \cdot 20$ MHz where i is the channel number is a multiple of 4 and $i = 36 + 4m$, with $m = 0, ...7, 16, ...26$ and also $5745 + (i - 149) \cdot 20$ MHz for $i = 149 + 4m$ and $m = 0, ..., 4$ as partially shown in Figure 4.36. More recent Wi-Fi expands further to the frequencies roughly between 6 GHz and 7 GHz also shown.

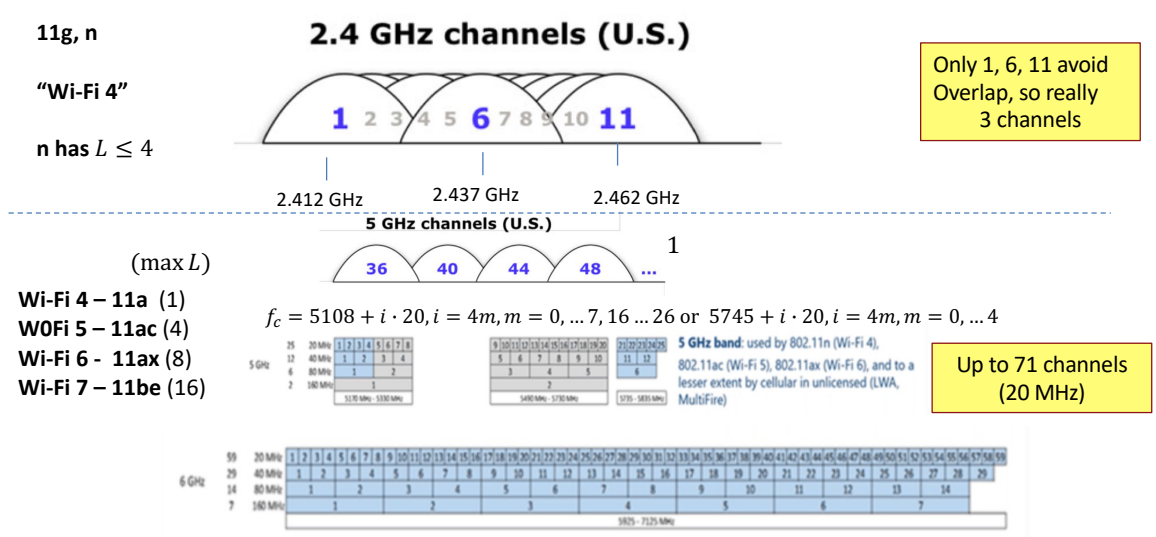


Figure 4.36: Wi-Fi Channel Possibilities in USA.

Complex Baseband Equivalent Partitioning: At baseband, the earliest 20-MHz bandwidth transmission systems number the tones from -31 to +31 (the 64th carrier at the band

[61] IEEE Standard Part 11, Wireless LAN Medium Access Control and Physical Layer Specifications, December 2016, IEEE 3 Park Ave, New

R (Mbps)	constellation	code rate	b_n	$\bar{b}_n$	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 4.9: Table of early 20MHz Wi-Fi data rates and associated parameters.

edge is not used). Of the remaining 63 possible tones, the tone 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 carry zero energy 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 for M-SQ QAM (M=2 for BPSK) will thus be

$$R = \log_2(M) \cdot (\text{code rate}) \cdot (48 \text{ tones}) \cdot 250 \text{ kHz} = [0.5,\ 1,\ 2, \text{ or } 3] \cdot (12 \text{ or } 18\ ) \text{ Mbps}. \quad (4.315)$$

Coding: Table 4.9 summarizes some simpler Wi-Fi coding options $[|C|\ r]$ for transmission. The code is a well-known 64-state $r = \frac{1}{2}$ convolutional code mapped on to 4 possible constellations in basic Wi-Fi, and Chapters 9 describes this code with $d_{free} = 10$ more completely. A code rate of 1/2 allows considerable recovery of lost/notched tones, although the early standardized codes in Wi-Fi are not the most powerful that could have been used. The $r = 3/4$ code deletes ("punctures") some encoder-output bits, reducing the free distance to $d_{free} = 6$.

constellation	code rate	$1/T' = 20$ $\nu = 16$ Mbps	$1/T' = 20$ MHz $\nu = 8$ Mbps	$1/T' = 40$ $\nu = 16$ Mbps	$1/T' = 40$ MHz $\nu = 8$ Mbps
BPSK	1/2	6.5	7.2	13.5	15
4QAM	1/2	13	14.4	27	30
4QAM	3/4	19.5	21.7	40.5	45
16QAM	1/2	26	28.9	54	60
16QAM	3/4	39	43.3	81	90
64QAM	2/3	52	57.8	108	120
64QAM	3/4	58.5	65	121.5	135
64QAM	5/6	65	72.2	135	150
256QAM	3/4	78	86.6	162	180
256QAM	5/6	86.7	96.3	180	200

Table 4.10: Table of Wi-Fi 6 data rates and associated parameters without MIMO.

Spectrum and Power: 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.

Wi-Fi 5 and Wi-Fi 6 Advances: IEEE standards IEEE 802.11n, ac (Wi-Fi 5), and ax (Wi-Fi 6) have extended 802.11(a) to vector OFDM and they allow aggregation of adjacent

20-MHz channels (2 in 11n) and (up to 4 or 8 in 11ac and 11ax, with 8 sometimes using disjoint channel bands in the aggregration) channels together to double, quadruple, or 8x the possible raw data rates with respect to 11a and 11g. For one 20 MHz channel, these standards allow use of tones ± 27 and ± 28 for data transmission, thus increasing the number of data-carrying tones to 52 (from 48 in g and a) - this means that data rates now are multiples in Equation (4.315) $(52/48) \cdot 12 = 13$ Mbps or $(52/48) \cdot 18 = 19.5$ Mbps. Further an optional cyclic-prefix length of only $\nu = 8$ complex samples is allowed as an option, then increasing possible data rates to multiples of $(80/72) \cdot 6.5 = 14.4$ (or 21.7) Mbps. For two bonded adjacent channels, there are a maximum of 128 tones, but again carrier (0) and 64 are not used, and tones ± 1 are also not used to assist carrier recovery. There are 6 pilots on tones at $\pm$ 11, 25, and 53. Carriers $\pm$ 59, 60, 61, 62, and 63 are not used to allow channel spacing. Thus, there are 108 used tones. Data rates are thus

$$R = \log_2(M) \cdot (\text{code rate}) \cdot (108 \text{ tones}) \cdot 250 \text{ kHz} = [0.5,\ 1,\ 2, \text{ or } 3] \cdot (27 \text{ or } 40.5\ ) \text{ Mbps}. \tag{4.316}$$

When the shortened cyclic prefix is used, this increases to $k \cdot (80/72) \cdot 13.5 = k \cdot 15$ Mbps. Table 4.10 then summarizes the possible data rates (only ac and ax use the 256 QAM options).

Modulation and coding schemes for single spatial stream

MCS index[a]	Modulation type	Coding rate	Data rate (in Mb/s)[b]							
			20 MHz channels		40 MHz channels		80 MHz channels		160 MHz channels	
			1600 ns GI[c]	800 ns GI	1600 ns GI	800 ns GI	1600 ns GI	800 ns GI	1600 ns GI	800 ns GI
0	BPSK	1/2	4(?)	4(?)	8(?)	9(?)	17(?)	18(?)	34(?)	36(?)
1	QPSK	1/2	16	17	33	34	68	72	136	144
2	QPSK	3/4	24	26	49	52	102	108	204	216
3	16-QAM	1/2	33	34	65	69	136	144	272	282
4	16-QAM	3/4	49	52	98	103	204	216	408	432
5	64-QAM	2/3	65	69	130	138	272	288	544	576
6	64-QAM	3/4	73	77	146	155	306	324	613	649
7	64-QAM	5/6	81	86	163	172	340	360	681	721
8	256-QAM	3/4	98	103	195	207	408	432	817	865
9	256-QAM	5/6	108	115	217	229	453	480	907	961
10	1024-QAM	3/4	122	129	244	258	510	540	1021	1081
11	1024-QAM	5/6	135	143	271	287	567	600	1134	1201

[h]

Table 4.11: Update of Table 4.10 for Wi-Fi 6 standard

MIMO Wi-Fi: Modern Wi-Fi also allow use of 2, 4, and 8 (and soon 16) antennas with exactly the same formats, so then the data rates in Table 4.10 can correspondingly double, quadruple, or multiply by 8 exactly if the channel is such that there are this number of independent non-interfering (non cross-talking) parallel channels (on each tone - so vector COFDM). These spatial dimensions support spatial streams. The energy and constellation-size and code rate may be different in Wi-Fi for each spatial stream. Thus, there is loading over space (and this can be water-fill based or Levin-Campello based with a tone gain becoming the geometric gain for each spatial stream. The elimination of inter-dimensional interference between the spatial streams is possible by using vector coding and SVD on each of the $L_x \times L_x$ subchannels. These SVD's are learned through training of each packet (several symbols per packet) and communicated to the transmitter (see Chapter 7, but the M matrix needs return and is provided as a series of 2-dimensional rotations for each energized tone),

but $b_n = b$ is constant on all used tones regardless of the SVD result, so the optimization is somewhat incomplete but nevertheless provides gain. The number of rotations is limited so the M is approximated. Some systems may use the same M matrix on all tones to reduce the amount of required feedback. The use of multiple antennas also allows the parity of the C-OFDM system to be effectively multiplied when only some of the possible paths experience severe multipath distortion. Thus, for example the maximum speed (last 64 QAM line in Table 4.10) for 4 antennas at both transmitter and receiver in the corresponding Wi-Fi is then $4 \times 150 = 600$ Mbps. 11ac and ax also use multi-user MIMO MAC and BC methods as in Chapter 2 on each tone/dimension.

More Tones in Advanced Wi-Fi: However, recent Wi-Fi also allow up to 4-16 channels to be bonded (up to 320 Mhz instead of a single 20 MHz channel). For the 4-channel bonding (80 MHz), again Channel 128 and 0 ± 1 are not used. Pilots are on tones $\pm$ 11, 39, 75, and 103. Channels $\pm$ 123, ... , 127 are also not used for spacing. This means that 234 tones carry data. Thus, the 40 MHz numbers in Table 4.10 can be multiplied by (234/108) to get new data rates for 4-channel bonding. Similarly for 8-channel bonding, Channel 256, 0, and $\pm$ 1-5 are not used. Pilots occur on channels $\pm$ 25,53,89,117, 139, 167, 203, and 231. This leaves 484 data-carrying tones with 8-channel bonding (160 MHz), and correspodingly then (484/108) times the numbers for 40 MHz in Table 4.10. Table 4.11 also updates the earlier Table 4.10 for several improvements (along with 8 antennas maximum allowed) in reducing the number of pilots, reducing the cyclic-prefix further, and using larger constellations. The maximum data rate then for Wi-Fi 6 with 8 antennas approaches 10 Gbps, presumably under nearly ideal conditions with short connection, no noise or interference, etc.

MU-MIMO Wi-Fi: Wi-Fi rarely attains Table 4.11's high speeds because the wider bandwidths makes it highly likely that other Wi-Fi users in the same band (other Wi-Fi systems that are not coordinated in these unlicensed bands) will collide, along with general limits of channel attenuation and noise. Since the channel bonding causes all the bandwidth likely to be used, collisions occur too frequently. This leads to a higher level strategy called carrier-sense multiple access (see Chapter 2) being used with a retransmission of the packet upon collision a random time period later. Further, Wi-Fi6 allows the spatial streams to go to different users simultaneously, up to 4 such users, as per Chapter 2's BC (downlink) and MAC (uplink) multi-user systems, using an assigned time-slot method for both directions that also helps avoid collisions. Wi-Fi increasingly tries to use frequency blocks (channels of 20MHz multiples wide) and spatial streams – that is dimensionality – to allocate sufficient data-rate access to many simultaneously active users.

IEEE standards 802.11ad (Wi-Gig 5) and 11ay (Wi-Gig 6) may also be of interest to the reader, and use a similar C-OFDM methodology but at high carrier frequencies in the 58-60 GHz range and with even higher speeds possible because of wider channels and bands used. Propagation at these frequencies may be more severely limited.

EXAMPLE 4.7.5 **[Digital Video Broadcast] Digital Terrestrial Television Broadcast (DTTB)** and **Digital Video Broadcast (DVB)** transmit audio and video respectively on traditional[62] FM and TV channels respectively with C-OFDM. There is no feedback (no uplink at all) channel, so no dynamic loading is possible. These systems use Figure 4.37's **SFN (single frequency network)**. 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 a 6-7.6 MHz-wide television channels, such delay multiple in-band notches in the transmission band. C-OFDM's codes recover tones lost to any such notches.

[62] And sometimes on other allocated bands that depend on national regulatory provisions.

DTTB/DVB Complex Baseband Equivalent Partitioning: The first television carrier is at 52 MHz for channel 2 and then asymmetric channel tone use about this carrier and others at $52 + i \cdot 6$ MHz, creating complex baseband equivalent channels when carrier demodulated to baseband. Other possibilities have the carriers separated by 8 MHz instead of 6 MHz. The DVB C-OFDM design over this 6-8 MHz, and is used in most of the world except the USA[63]. These systems use either $\overline{N} = 2048$ or 8192 tones, and energizes 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 approximately 1.116 kHz (896μs) for 8192 tones and 4.464 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 f $1/T' \approx 9.142$ MHz ($T' = 109.375$ ns) for each of two real sampling devices. (A single sampling device at higher speed, e.g. "oversampled ADC and/or DAC," may also be used with digital decimation/interpolation returning signals to the 9.142 MHz.)

DTTB/DVBCodes: The broadcaster selects DTTB/DVB constellations allow 4QAM, 16QAM, or 64QAM transmission with various code rates of (1) first code rate (172/204)$\times$ (2) 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. The chosen parameters are embedded in control packets that the corresponding field receivers can decode and then implement correspondingly. This system at least doubles the number video signals with respect to analog TV, and often increases by a factor of 8, enabling greater choice by consumer and higher revenue by broadcaster.

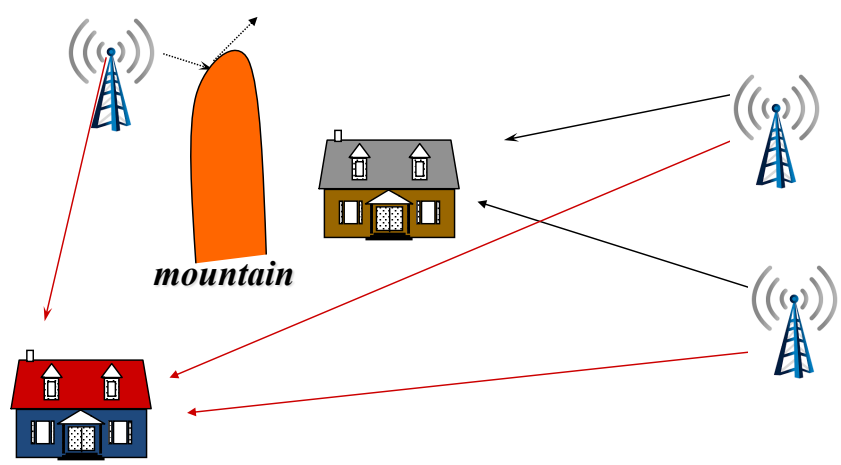


Figure 4.37: **Single Frequency Network example for television-broadcast channel partitioning.**

[63]The USA moved perhaps too early to digital television and used a single-carrier method, while most of the rest of the world slightly later benefitted from C-OFDM systems. As digital broadcast television migrated rapidly to cable and satellite delivery in the USA, the digital format was never upgraded to formats used in the rest of the world.

EXAMPLE 4.7.6 **C-OFDM in 4G and 5G Cellular**

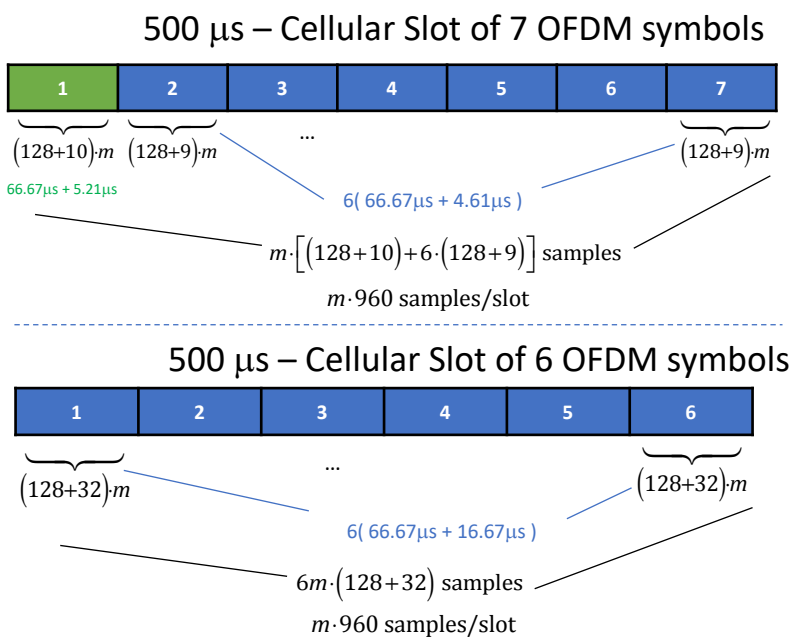


Figure 4.38: **LTE's prefix options - short and long**

Figure 4.38 displays 4G and 5G cellular wireless systems' use of C-OFDM with downlink slots with integer index $m = 1, 2, 4, 8, 12, 16$. There is a long cyclic-prefix option of 1/4 the FFT size $\overline{N}$ with 6 successive OFDM symbols in a 500 μs "slot." The short cyclic-prefix option is more complicated in that 7 OFDM symbols fit into the same 500 μs slot, the first with cyclic prefix of $10 \cdot m$ samples and the remaining 6 with cyclic prefix of $9 \cdot m$ samples. The total number of samples in one slot for both is $960 \cdot m$ samples in either the short or long cyclic-prefix formats. Example 4.7.6's table lists the sampling rates that correspond to $\Delta f = 15$ kHz, or sampling rate $\overline{N} \cdot m \cdot 15$ kHz.

The short-prefix symbol rate follows from Figure 4.38 with any m so let $m = 1$ and

$$\frac{1}{T_s} = \frac{7}{500\mu s} = 14 \text{ kHz} \quad . \tag{4.317}$$

Similarly, the long-cyclic-prefix symbol rate is

$$\frac{1}{T_l} = \frac{6}{500\mu s} = 12 \text{ kHz} \quad , \tag{4.318}$$

so that data rates for the long cyclic prefix are $12/14 = 6/7$ data rates with short prefix. Cellular uses a **resource element** as one sub symbol on one tone of one OFDM symbol. These effectively carry bits at multiples of 14kbps or 12 kbps respectively. A **resource block** is 12 tones allocated to a single user corresponding to a smallest user data rate granularity of 168 kbps and 144 kbps respectively. Extra tones in a symbol beyond those in a resource block are for pilots or other synchronization symbols (which is complicated in cellular and beyond this text's scope).

The lowest all users' sum data rate corresponds to QPSK on all cellular tones with $m = 1$ and $L = 1$ at 2.016 Mbps (and 1.728 Mbps with long prefix). These data rates are essentially

those of an encoder output. As mentioned before, and studied in Chapter 10, codes are used. The codes use puncturing of a rate 1/3 bit-interleaved turbo code with various coding gains depending on the code rate, which can be as low as 1/3 and as high as .95. Higher rate codes with only 5% redundancy are used on "easy" channels with high SNR on the used tones. Lower rate codes (with larger coding gains or smaller gaps) are used on "difficult" channels with low SNRs. Section 4.4's C-OFDM discrete loading determines the specific modulation and coding scheme $[|C|\ r]$. The receiver measures SNR_{geo} for a specific resource block and returns a quality indicator (one of 16 levels) to the transmitter to indicate how many bits/tone (same on all tones) and how much coding power is necessary. Different users downlink may however use different codes (so within different resource blocks, the number of bits and code would be the same but differ from other resource blocks. With multiple antennas ($L > 1$), the transmit M matrix in Vector OFDM may differ for different users (that is for each resource block).

bwdth MHz	m	$1/T'$ MHz	$\overline{N}+\nu_s$ ($1/T_s = 14$ kHz) / $\overline{N}+\nu_l$ (*) ($1/T_l = 12$ kHz)	$\overline{N}^*$	samples slot	Δf kHz	RBs	b_{min} / b_{mid} / b_{max}	L / 1 / 1	R_{min} / R_{mid} / R_{max} Mbps *
1.25	1	1.92	128+6.17	76	960	15	6	2	1	2.016
			128+32		960			4	1	4.032
								6	1	6.048
								6	2	12.096
								6	4	24.192
3	2	3.84	256+12.33	181	1920	15	15	2	1	5.04
			256+64		1920			4	1	10.08
								6	1	15.12
								6	2	30.24
								6	4	60.48
5	4	7.68	512+24.67	301	3840	15	25	2	1	8.40
			512+128		3840			4	1	16.80
								6	1	25.20
								6	2	50.40
								6	4	100.8
10	8	15.36	1024+49.33	601	7680	15	50	2	1	16.80
			1024+256		7680			4	1	33.60
								6	1	50.4
								6	2	100.8
								6	4	200.16
15	12	23.04	1536 +74	901	11520	15	75	2	1	25.20
			1536+384		11520			4	1	50.40
								6	1	75.6
								6	2	151.2
								6	4	302.4
20	16	30.72	2048 +98.67	1201	15360	15	100	2	1	33.6
			2048+384		15360			4	1	67.20
								6	1	100.8
								6	2	201.60
								6	4	403.20

Table 4.12: Table downlink cellular's basic modulation parameters. (*Long-prefix data rates are 6/7 of short-prefix data rates listed.)

EXAMPLE 4.7.7 [Cellular Expanded spectra and carriers] 5G uses the same mod-

ulation formats as 4G, but with more carrier-frequency choices, and also with yet wider channel bandwidths (larger numbers of dimensions $\overline{N}$) and yet more antennas L_x and L_y.

The cyclic-prefix method and and slot structure remain the same as in Example 4.7.6. 5G's updates allow larger FFT sizes, wider channel bandwidths, wider Δf, and more (and virtual) antennas. Additionally higher carrier frequencies are allowed up to 6 GHz in **Frequency Range 1 (FR1)** and between 24.25 GHz and 52.6 GHz in a new "millimeter wave" band called **Frequency Range 2 (FR2)**. Table 4.13enumerates these channels from Table 5.2.1 of the 3GPP TS38.101-1 document.

NR Operating Band	Uplink (UL) operating band BS receive UE transmit	Downlink (DL) operating band BS transmit UE receive	Duplex Mode
	F_{UL_low} – F_{UL_high}	F_{DL_low} – F_{DL_high}	
n1	1920 MHz – 1980 MHz	2110 MHz – 2170 MHz	FDD
n2	1850 MHz – 1910 MHz	1930 MHz – 1990 MHz	FDD
n3	1710 MHz – 1785 MHz	1805 MHz – 1880 MHz	FDD
n5	824 MHz – 849 MHz	869 MHz – 894MHz	FDD
n7	2500 MHz – 2570 MHz	2620 MHz – 2690 MHz	FDD
n8	880 MHz – 915 MHz	925 MHz – 960 MHz	FDD
n20	832 MHz – 862 MHz	791 MHz – 821 MHz	FDD
n28	703 MHz – 748 MHz	758 MHz – 803 MHz	FDD
n38	2570 MHz – 2620 MHz	2570 MHz – 2620 MHz	TDD
n41	2496 MHz – 2690 MHz	2496 MHz – 2690 MHz	TDD
n50	1432 MHz – 1517 MHz	1432 MHz – 1517 MHz	TDD
n51	1427 MHz – 1432 MHz	1427 MHz – 1432 MHz	TDD
n66	1710 MHz – 1780 MHz	2110 MHz – 2200 MHz	FDD
n70	1695 MHz – 1710 MHz	1995 MHz – 2020 MHz	FDD
n71	663 MHz – 698 MHz	617 MHz – 652 MHz	FDD
n74	1427 MHz – 1470 MHz	1475 MHz – 1518 MHz	FDD
n75	N/A	1432 MHz – 1517 MHz	SDL
n76	N/A	1427 MHz – 1432 MHz	SDL
n78	3300 MHz – 3800 MHz	3300 MHz – 3800 MHz	TDD
n77	3300 MHz – 4200 MHz	3300 MHz – 4200 MHz	TDD
n79	4400 MHz – 5000 MHz	4400 MHz – 5000 MHz	TDD
n80	1710 MHz – 1785 MHz	N/A	SUL
n81	880 MHz – 915 MHz	N/A	SUL
n82	832 MHz – 862 MHz	N/A	SUL
n83	703 MHz – 748 MHz	N/A	SUL
n84	1920 MHz – 1980 MHz	N/A	SUL
n257	26500 MHz – 29500 MHz	26500 MHz – 29500 MHz	TDD
n258	24250 MHz – 27500 MHz	24250 MHz – 27500 MHz	TDD
n260	37000 MHz – 40000 MHz	37000 MHz – 40000 MHz	TDD

Table 4.13: 5G NR Radio Channels

The number of options rapidly defies tables like the 4G case above, so instead the carrier spacing (Δf) is specified along with the existing single-channel bandwidths and enlarged bandwidths in Table 4.14.

NR band / SCS / UE Channel bandwidth												
NR Band	SCS kHz	5 MHz	10[1,2] MHz	15[2] MHz	20[2] MHz	25[2] MHz	30 MHz	40 MHz	50 MHz	60 MHz	80 MHz	100 MHz
n1	15	Yes	Yes	Yes	Yes							
	30		Yes	Yes	Yes							
	60		Yes	Yes	Yes							
n2	15	Yes	Yes	Yes	Yes							
	30		Yes	Yes	Yes							
	60		Yes	Yes	Yes							
n3	15	Yes	Yes	Yes	Yes	Yes	Yes					
	30		Yes	Yes	Yes	Yes	Yes					
	60		Yes	Yes	Yes	Yes	Yes					
n5	15	Yes	Yes	Yes	Yes							
	30		Yes	Yes	Yes							
	60											
n7	15	Yes	Yes	Yes	Yes							
	30		Yes	Yes	Yes							
	60		Yes	Yes	Yes							
n8	15	Yes	Yes	Yes	Yes							
	30		Yes	Yes	Yes							
	60											
n20	15	Yes	Yes	Yes	Yes							
	30		Yes	Yes	Yes							
	60											
n28	15	Yes	Yes	Yes	Yes							
	30		Yes	Yes	Yes							
	60											
n38	15	Yes	Yes	Yes	Yes							
	30		Yes	Yes	Yes							
	60		Yes	Yes	Yes							
n41	15		Yes	Yes	Yes			Yes	Yes			
	30		Yes	Yes	Yes			Yes	Yes	Yes	Yes	Yes
	60		Yes	Yes	Yes			Yes	Yes	Yes	Yes	Yes
n50	15	Yes	Yes	Yes	Yes			Yes	Yes			
	30		Yes	Yes	Yes			Yes	Yes	Yes	Yes	
	60		Yes	Yes	Yes							
n51	15	Yes										
	30											
	60											
n66	15	Yes	Yes	Yes	Yes							
	30		Yes	Yes	Yes			Yes				
	60		Yes	Yes	Yes			Yes				
n70	15	Yes	Yes	Yes	Yes	Yes						
	30		Yes	Yes	Yes	Yes						
	60		Yes	Yes	Yes	Yes						
n71	15	Yes	Yes	Yes	Yes							
	30		Yes	Yes	Yes							
	60											
n74	15	Yes	Yes	Yes	Yes							
	30		Yes	Yes	Yes							
	60		Yes	Yes	Yes							
n75	15	Yes	Yes	Yes	Yes							
	30		Yes	Yes	Yes							
	60		Yes	Yes	Yes							
n76	15	Yes										
	30											
	60											
n77	15		Yes		Yes			Yes	Yes			
	30		Yes		Yes			Yes	Yes	Yes	Yes	Yes
	60		Yes		Yes			Yes	Yes	Yes	Yes	Yes
n78	15		Yes		Yes			Yes	Yes			
	30		Yes		Yes			Yes	Yes	Yes	Yes	Yes
	60		Yes		Yes			Yes	Yes	Yes	Yes	Yes
n79	15							Yes	Yes			
	30							Yes	Yes	Yes	Yes	Yes
	60							Yes	Yes	Yes	Yes	Yes
n80	15	Yes	Yes	Yes	Yes	Yes	Yes					
	30		Yes	Yes	Yes	Yes	Yes					
	60		Yes	Yes	Yes	Yes	Yes					
n81	15	Yes	Yes	Yes	Yes							
	30		Yes	Yes	Yes							
	60											
n82	15	Yes	Yes	Yes	Yes							
	30		Yes	Yes	Yes							
	60											
n83	15	Yes	Yes	Yes	Yes							
	30		Yes	Yes	Yes							
	60											
n84	15	Yes	Yes	Yes	Yes							
	30		Yes	Yes	Yes							
	60		Yes	Yes	Yes							

Table 4.14: Uses for different Δf = SCS in 5G NR specification (sub-carrier-spacing) FR1

NR band / SCS / UE Channel bandwidth					
NR Band	SCS kHz	50 MHz	100 MHz	200 MHz	400 MHz
n257	60	Yes	Yes	Yes	Yes
	120	Yes	Yes	Yes	Yes
n258	60	Yes	Yes	Yes	Yes
	120	Yes	Yes	Yes	Yes
n260	60	Yes	Yes	Yes	Yes
	120	Yes	Yes	Yes	Yes

Table 4.15: Uses for different Δf = SCS in 5G NR specification (sub-carrier-spacing) FR2

Additionally, Table 4.16 lists the number of resource blocks and

SCS (kHz)	5MHz	10MHz	15MHz	20 MHz	25 MHz	30 MHz	40 MHz	50MHz	60 MHz	80 MHz	100 MHz
	N_{RB}	N_{RB}	N_{RB}	N_{RB}	N_{RB}	N_{RB}	N_{RB}	N_{RB}	N_{RB}	N_{RB}	N_{RB}
15	25	52	79	106	133	[TBD]	216	270	N/A	N/A	N/A
30	11	24	38	51	65	[TBD]	106	133	162	217	273
60	N/A	11	18	24	31	[TBD]	51	65	79	107	135

FR1

FR2

SCS (kHz)	50MHz	100MHz	200MHz	400 MHz
	N_{RB}	N_{RB}	N_{RB}	N_{RB}
60	66	132	264	N.A
120	32	66	132	264

Table 4.16: Resource Blocks N_{RB}

can be used to compute the corresponding data rate at

$$R = 168 \text{ kHz} \times N_{RB} \times b \times L \times \frac{\Delta f}{15 \text{ kHz}} \quad . \tag{4.319}$$

256 QAM is allowed, so the bits per tone (b can be as large as 8 (instead of 6 earlier) and the number of antennas can be up to 32 (but only 8 spatial channels used virtually) downlink and 4 uplink (per user). The maximum per-user spatial increase in data rate thus remains at 4.

Thus an FR1 system with 50 MHz bandwidth and Δf = 30 kHz could have a data rate (reduced by coding rate) as high as

$$R = 168 \times 133 \times 4 \times 4 \times 2 = 715.008\text{Mbps} \tag{4.320}$$

Speeds in excess of 1 Gbps are possible under best conditions in FR1. Speeds approaching 5.68 Gbps are possible in FR2.

4.7.5 Noise-Equivalent Channel Interpretation for DMT

Section 1.7 established the equivalence of an additive ("colored") Gaussian noise channel to a matrix/filtered AWGN. The colored Gaussian noise generally has power spectral density $\frac{\mathcal{N}_0}{2} \cdot R_n(f)$. The channel pulse response of the equivalent AWGN is the pulse response of the original channel adjusted by the Gaussian noise's inverse square-root power spectral density[64]:

$$\widetilde{H}(f) = \frac{H(f)}{\sqrt{S_n(f)}} \quad . \tag{4.321}$$

A receiver "noise-whitening" filtering $1/\sqrt{S_n(f)}$ preprocesses the channel output prior 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 (4.321).

[64]The realizable causal and causally invertible filter, see the Appendices of Chapter 3.

For DMT/OFDM, the noise equivalent view discussed earlier leads to a dimension set with gains

$$g_n = \frac{|H(n/T)|^2}{S_n(n/T)} \tag{4.322}$$

and AWGN variance $\frac{\mathcal{N}_0}{2}$, as long as R_{nn} is also circulant. In practice for colored noise, $S_n(n/T)$ does not correspond to a circulant matrix. As $\overline{N} \to \infty$, a stationary noise deviates from a circulant noise by a vanishingly small amount if the noise autocorrelation sequence has finite length, which will occur for reasonable noises. Indeed, the autocorrelation matrix can be close to circulant, even for moderate values of $\overline{N}$. A more common view, which avoids the use of the whitening filter, is that each dimension has gain in (4.322) with $SNR_n = \bar{\mathcal{E}}_{\boldsymbol{x}} \cdot g_n$ (or $\widetilde{\mathcal{E}}_n \cdot g_n$ for complex baseband noise in (4.322)'s denominator). This avoids the use of a noise-whitening filter, but 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 $\overline{N}T'$ is large with respect to the noise autocorrelation function's nonzero duration. In practice, the values used for $\overline{N}$ 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.) Chapter 7 reviews noise spectra estimation.

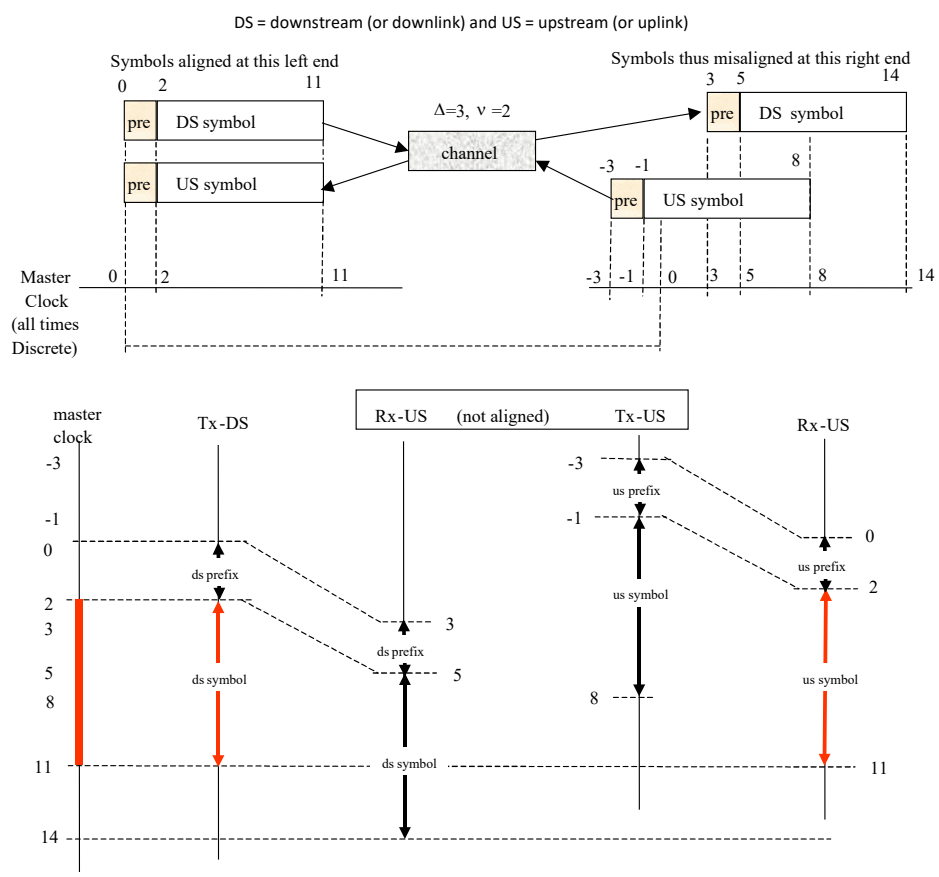


Figure 4.39: Illustration of alignment problem in DMT with only cyclic prefix.

4.7.6 Isaksson's Zipper

Often DMT (C-OFDM) is bi-directional. That means both transmission directions 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 system may also separate opposite-direction transmission by assignment of independent tone sets for each direction, known as frequency-division duplexing. Time-division duplexing sends different directions alternately in time. Each of echo cancellation, frequency-division duplexing, and time-division duplexing have enormous simplification of their signal processing if the DMT/C-OFDM signals in the two directions have common sampling rates and symbol rates. Alignment of the symbol boundary at one end is simple because the transmitter directly aligns symbol boundaries. However, it would appear that alignment at the other end is then impossible with any realistic channel delay, as per Figure 4.39 for the example of $\overline{N} = 10$, $\nu = 2$, and $\Delta = 3$ (the channel delay). Figure 4.39's DS and US refer to the downstream[65] (or downlink) and upstream (or uplink) direction transmissions respectively. Alignment at Figure 4.39's left leads to misalignment at the right.

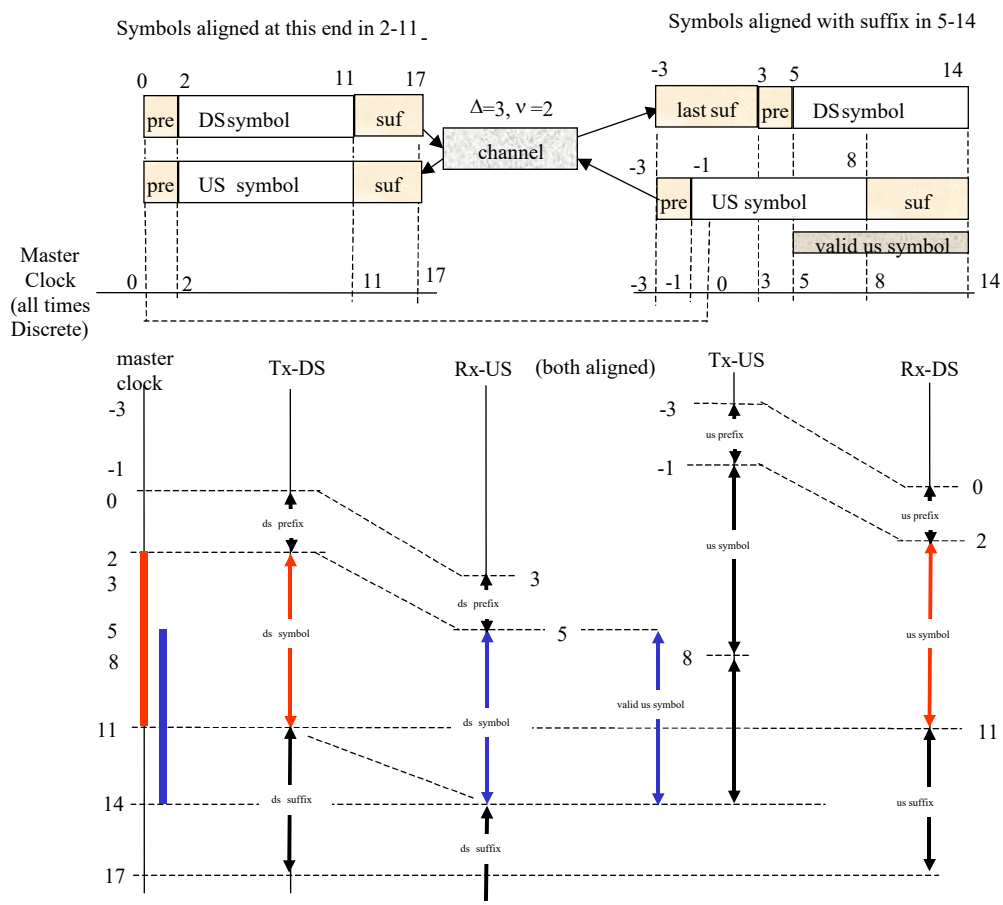


Figure 4.40: Use of cyclic suffix to align DMT symbols at both ends of transmission.

Figure 4.40 shows an Isaksson Zipper[66] solution that adds a 6 sample cyclic suffix after the DMT

[65] Downstream means toward the consumer end and upstream means from the consumer end.

[66] After Michael Isaksson, at the invention time of Telia Research in Sweden, and is standardized for use in what is known as VDSL2 (ITU G.993.2) and now also used in cellular and Wi-Fi. This duplexing method facilitates a certain type of

signal, repeating the beginning of the symbol (that is the beginning after the cyclic prefix). There are now 6 valid positions for receiver FFT alignment without interference among tones. One of those 6 positions at the left downstream is the same position as one of the 6 positions upstream, allowing alignment at the left and right simultaneously for the price of an additional 6 guard-period dimensions. The value 6 generalizes to $2 \cdot \Delta$. As $\overline{N} \to \infty$, the cyclic suffix overhead can be made negligible. Figure 4.41 shows a method to reduce the smallest cyclic suffix length to Δ by advancing the left-side downstream signal by Δ samples, allowing one of 3 valid downstream positions at the left to corresponding to one of 3 valid upstream positions, and correspondingly one overlap also at the right. Thus Δ is the minimum cyclic suffix length necessary for alignment at both ends simultaneously, adding to the existing overhead of ν.

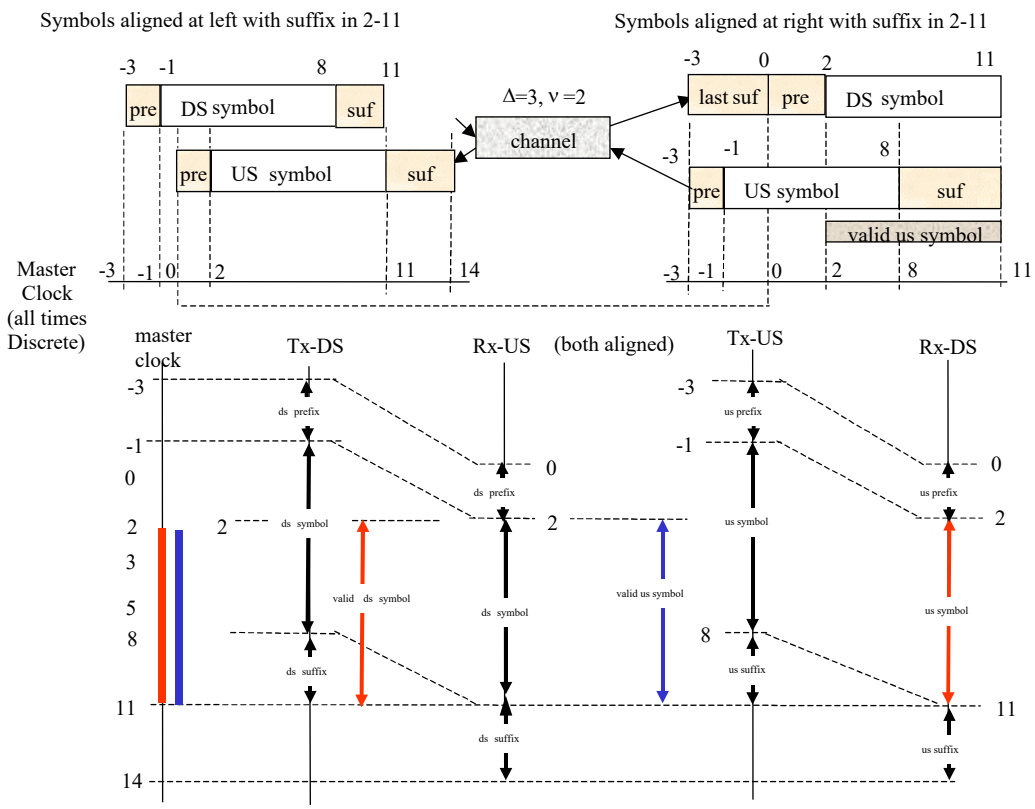


Figure 4.41: Use of both cyclic suffix and timing advance to align DMT signals at both ends of transmission.

EXAMPLE 4.7.8 [VDSL] VDSL from Example 4.7.2 has a larger-$\overline{N}$ and uses the cyclic suffix and timing advance of Zipper. The upstream and downstream tones separate through frequency-division duplexing with frequencies determined as best needed as a function of line, channel, and service-provider desires, although some power spectral density masks are specified to accommodate regional emissions constraints. An example frequency assignment appeared in Example 4.7.2. These example cyclic prefixes listed are actually the sum of the cyclic prefix length and the cyclic suffix length. In fact, these values also includes some dimensions that may also allow Section 4.8's time-domain windowing on dimensions outside

one-sided; one-sided means that dimensions may not be coordinated on one side of the connection, meaning that either M or F cannot be realized. This alignment was presumed in Chapter 2's multi-user systems.

those in the cyclic extension and FFT'd dimensions. The Zipper duplexing mitigates need for 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." This type of duplexing also can align multiple signals in both directions for multiple user systems, leading to each dimension having independent crosstalk so that systems can use vector DMT/OFDM bi-directionally.

4.7.7 Filtered Multitone (FMT)

Developed for SISO, Filtered Multi-Tone (FMT) uses 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 an approach, specifically given the name FMT, pioneered by Cherubini , Eleftheriou, and Olcer of IBM[18]. This latter FMT approach squarely addresses the issue of the channel impulse response's ($h_c(t)$'s) nominal removal of orthogonality. FMT approaches find a discrete-time filter realization that approximates the GNC-satisfying MT's "brick-wall" frequency-tone characteristic. Essentially, FMT realizes basis functions with very low frequency content outside a main "lobe" of the Fourier transform (i.e., the main lobe centers about the tone's carrier frequency and there are second, third, fourth, etc. lobes that decay in any realizable approximation to the ideal band-pass filter characteristic of a modulated 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 type of ISI 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 $\overline{N}$ 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.

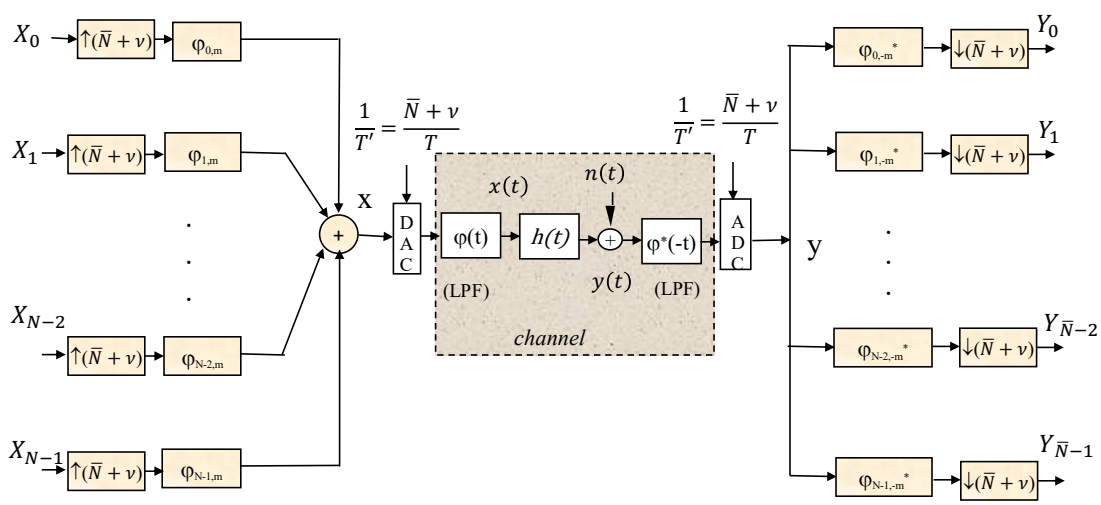


Figure 4.42: FMT partitioning block diagram.

The original modulated continuous-time waveform is again

$$x(t) = \sum_{j=-\infty}^{\infty} \sum_{n=0}^{\overline{N}-1} X_{n,j} \cdot \varphi_n(t - jT) \quad , \tag{4.323}$$

where j is a symbol index. Discrete-time implementations sample this waveform at sampling period T' where $T = (\overline{N}+\nu)\cdot T'$, but there is no cyclic prefix or time-domain guard period. In the FMT approach, ν is sometimes set to zero (unlike DMT). Figure 4.42 illustrates the discrete-time FMT modulator implementation. The up-arrow device in Figure 4.42 is an interpolator that simply inserts $(\overline{N}+\nu-1)$ zeros between successive possibly nonzero values. The down-arrow device is a decimator that selects every $(\overline{N}+\nu)^{th}$ sample from its input and ignores all others. The down-arrow devices are 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

$$\varphi_n(t) = \varphi(t)\cdot e^{\jmath\frac{2\pi}{T}\cdot\frac{\overline{N}+\nu}{\overline{N}}\cdot nt} \quad . \tag{4.324}$$

Each FMT dimension when $\nu > 0$ has excess bandwidth of $(1+\frac{\nu}{\overline{N}})\cdot\frac{1}{T}$ like DMT but without cyclic prefix. Said excess bandwidth permits a realizable approximation to an ideal bandpass filter for all tones (subchannels).

A dually indexed sampling time is

$$m = k\cdot(N+\nu)+i \;\; i = 0,...,N+\nu-1 \quad , \tag{4.325}$$

where i is a sample-offset sub-index within the kth symbol period. The continuous-time waveform's samples $x_m = x(mT')$ are then (with $\varphi_n(mT') = \varphi_{n,m}$):

$$\begin{aligned} x_m &= \sum_{j=-\infty}^{\infty}\sum_{n=0}^{\overline{N}-1} X_{n,j}\cdot\varphi_{n,m-j\cdot(\overline{N}+\nu)} & (4.326)\\ &= \sum_{j=-\infty}^{\infty}\sum_{n=0}^{\overline{N}-1} X_{n,j}\cdot\varphi_{n,k\cdot(\overline{N}+\nu)+i-j\cdot(\overline{N}+\nu)} \;\forall i = 0,...,\overline{N}+\nu-1 & (4.327)\end{aligned}$$

Each of the $\overline{N}+\nu$ basis filters is also indexed by the sample offset i and essentially decomposes into $\varphi_{n,k\cdot(\overline{N}+\nu)+i-j\cdot(\overline{N}+\nu)} = \varphi^{(i)}_{n,k-j}$, each of which then has symbol-rate implemention (rather than the original single filter at the sampling rate). Further substitution of this expression in (4.324) finds

$$\begin{aligned} \varphi^{(i)}_{n,k-j} &= \varphi^{(i)}_{k-j}\cdot e^{\jmath\frac{2\pi}{\overline{N}}n\cdot\left[(k-j)\cdot(\overline{N}+\nu)+i\right]} & (4.328)\\ &= \varphi^{(i)}_{k-j}\cdot e^{\jmath\frac{2\pi}{\overline{N}}n\cdot\left[(k-j)\cdot(\overline{N}+\nu)\right]}\cdot e^{\jmath\frac{2\pi}{\overline{N}}n\cdot[i]} \quad . & (4.329)\end{aligned}$$

Substitution of (4.329) into (4.327) provides for all $i = 0,...,\overline{N}+\nu-1$

$$\begin{aligned} x_{k\cdot(\overline{N}+\nu)+i} &= \sum_{j=-\infty}^{\infty}\sum_{n=0}^{\overline{N}-1} X_{n,j}\cdot\varphi^{(i)}_{k-j}\cdot e^{\jmath\frac{2\pi}{\overline{N}}n\cdot\left[(k-j)\cdot(\overline{N}+\nu)\right]}\cdot e^{\jmath\frac{2\pi}{\overline{N}}n\cdot[i]} & (4.330)\\ &= \sum_{j=-\infty}^{\infty}\varphi^{(i)}_{k-j}\cdot\sum_{n=0}^{\overline{N}-1} X_{n,j}\cdot e^{\jmath\frac{2\pi}{\overline{N}}n\cdot\left[(k-j)\cdot(\overline{N}+\nu)\right]}\cdot e^{\jmath\frac{2\pi}{\overline{N}}n\cdot[i]} \quad . & (4.331)\\ &= \sum_{j=-\infty}^{\infty}\varphi^{(i)}_{k-j}\cdot x_{i,j}\left([k-j]\nu\right) & (4.332)\end{aligned}$$

where $x_{i,j}\left([k-j]\nu\right)$ is the $\overline{N}$-point inverse discrete Fourier transform of $X_{n,j}\cdot e^{\jmath\frac{2\pi}{\overline{N}}n\cdot(k-j)\cdot\nu}$ in the ith sampling offset.[67] Nominally, this IDFT would need to be recomputed for every value of $(k-j)$ - however one notes that the term $e^{\jmath\frac{2\pi}{\overline{N}}n\cdot(k-j)\cdot\nu}$ that depends on $(k-j)$ is simply a circular shift of the time-domain output by $(k-j)\nu$ samples. Figure 4.43 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 $\pm L$,

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

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

Table 4.17: Comparison of DMT and FMT

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 an interpolation to the sampling rate associated with the excess bandwidth that is $(1+\nu/\overline{N})$ times faster than with no excess bandwidth.

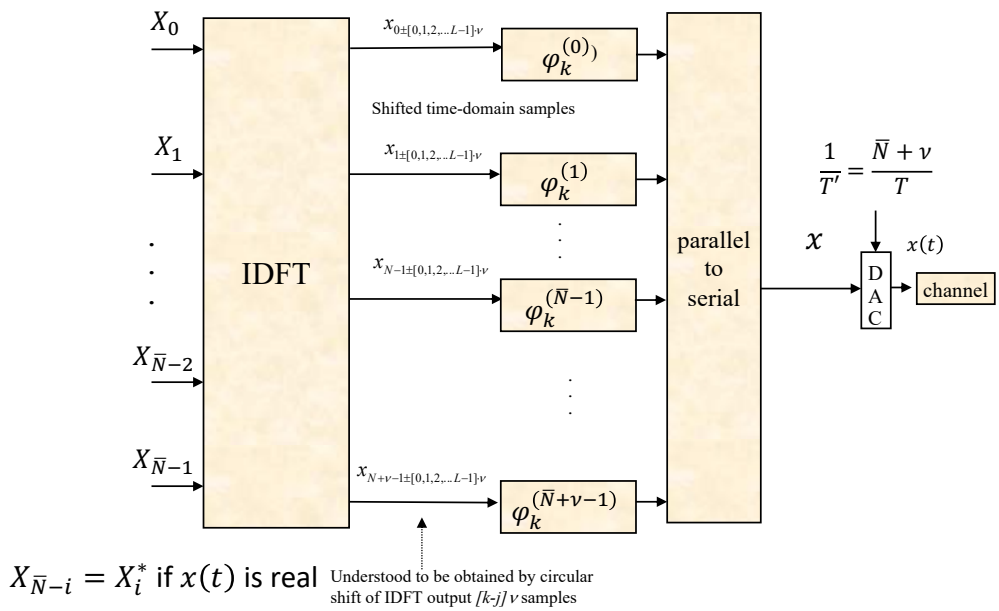


Figure 4.43: FMT partitioning using DFT's and symbol-rate filters.

Clearly if $\nu = 0$ and $\varphi_k = \delta_k$, then FMT and DMT are the same (assuming same $1/T = 1/(\overline{N}T')$). However, the two systems differ in how exact use of excess bandwidth (DMT in time-domain cyclic prefix, FMT directly in wider tones than would otherwise be necessary without cyclic prefixes by same ratio $\nu/\overline{N}$). FMT also allows a shaping filter to be used to try to approximately synthesize ideal-multitone bandpass subchannels, while DMT averts ISI in the guard period. Table 4.17 compares DMT and FMT.

Any FMT system always has at least some residual intersymbol interference and interchannel interference that a good choice of φ_k hopefully minimizes. However, equalizers may be necessary on each dimension and possibly between a few adjacent dimensions as in Figure 4.44. Usually, the functions have such low out-of-band energy that only intersymbol interference on each subchannel is of concern, so the receiver reduces to the FFT followed by a bank of symbol-rate DFE's as in Figure 4.44.

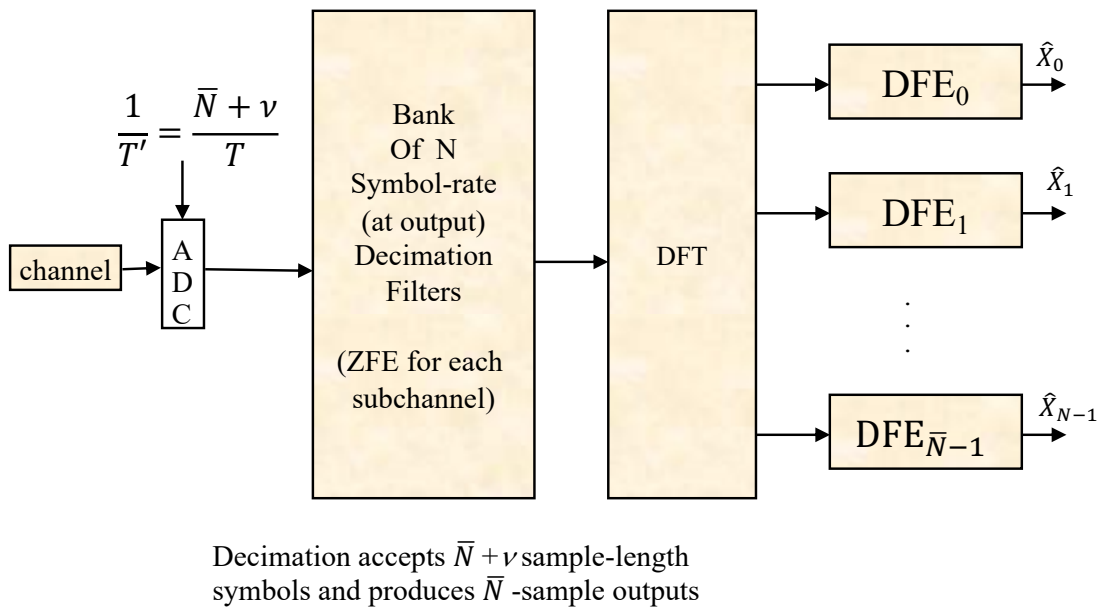


Figure 4.44: FMT receiver block diagram assuming negligible or no inter-channel interference.

4.7.7.1 Filter Selection for FMT

FMT admits an enormous number of possible $\varphi(t)$ choices. Some are 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 has to-date been futile.

Instead, Cherubini [18] has suggested some functions for DSL applications that appear promising:

EXAMPLE 4.7.9 [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

$$\Phi(D) = \sqrt{\frac{1+\rho}{2}} \cdot \frac{1+D}{1+\rho \cdot D} \quad . \tag{4.333}$$

As $\rho \to 1$, the functions become very sharp and the impulse response φ_k is very long. Figure 4.45 illustrates these functions for a $\rho = .1$. The curves 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). The associated receiver typically uses a DFE with Cherubini's coaster. The hope is that there is very little intersymbol or inter-channel interference and that $\overline{N}$ can be reduced (because the issue of $\nu/\overline{N}$ excess bandwidth being small is inpertinent when $\nu = 0$). The discrete-time response is given by

$$\varphi_k = \sqrt{\frac{1+\rho}{2}} \cdot \left[\delta_k + (1-\rho)(-\rho)^{k-1} \cdot u_{k-1}\right] \quad . \tag{4.334}$$

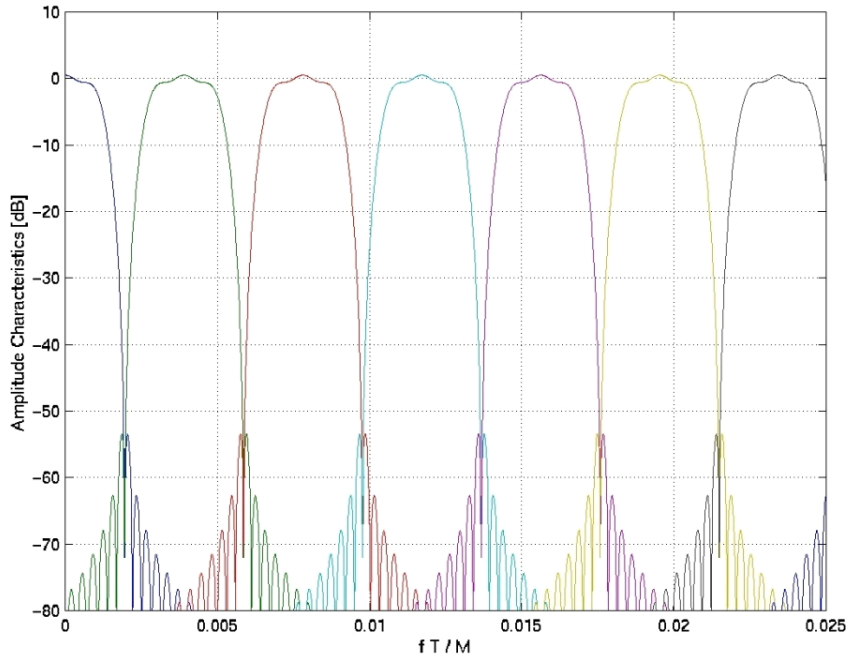


Figure 4.45: Cherubini's Coaster basis functions in frequency domain with $\rho = .1$.

Figure 4.45, 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 Figure 4.45 shows, 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 4.55 of Section 4.8 instead of the 55 dB shown with Cherubini's Coaster.

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

EXAMPLE 4.7.10 [Raised-Cosine with $\nu > 0$] Raised cosine functions with excess bandwisth $(\nu/\overline{N})$ 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

$$\varphi_k^{(i)} = \frac{4\sqrt{1+\frac{\overline{N}}{\nu}}}{\pi\sqrt{\nu T'}} \cdot \frac{\cos\left(\pi\left[k+\frac{i}{\overline{N}}\right]\right) + \frac{1+\frac{\overline{N}}{\nu}}{4\cdot\left[k+\frac{i}{\overline{N}}\right]} \cdot \sin\left(\pi\left[k+\frac{i}{\overline{N}}\right]\cdot\frac{\overline{N}-\nu}{\overline{N}+\nu}\right)}{1-16\cdot\left[k+\frac{i}{\overline{N}}\right]^2} \quad . \tag{4.335}$$

This function with sufficiently large number of taps in FIR approximation (for given excess bandwidth, smaller $\nu/\overline{N}$ 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.

4.8 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 extra ν-sample cyclic prefix or guard period, where $\nu + 1$ is the channel-pulse-response length in sampling periods. The required excess-bandwidth is then $\nu/\overline{N}$. 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, $\overline{N}$ consequently needs to be very large, potentially 10,000 samples or more. Complexity is still minimal with DMT/OFDM methods with large $\overline{N}$, when measured in terms of operation per unit time interval. However, large $\overline{N}$ implies large memory requirement (to store the bit tables, energy tables, FEQ coefficients, and intermediate FFT/IFFT results), often dominating the implementation. Further, large $\overline{N}$ 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 or interactive games).

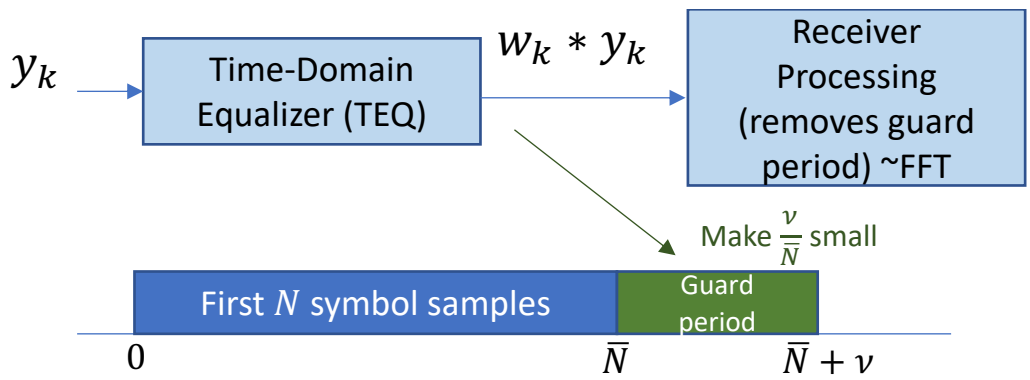


Figure 4.46: Time-Domain Equalizer Concept for Large-Symbol Designs (like DMT and Vector Coding).

Figure 4.46's **time-domain equalizer (TEQ)** concept, invented and first studied by J. Chow [22][68], uses a combination of short-length fixed (i.e. stationary) equalization and DMT/OFDM (or VC) to reduce ν, and thereby the required $\overline{N}$, to reach the highest performance levels with less memory and less latency. Chow's basic MMSE approach results 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: Essentially these methods attempt to maximize the overall SNR_{dmt} by optimizing equalizer settings and energy/bit distributions jointly. A heavily used "cyclic reconstruction" or "per-tone equalization" approach appears later in this section.

Subsection 4.8.1 studies the general data-aided equalization problem with infinite-length equalizers. The TEQ there generalizes the DFE's feedforward section. Subsection 4.8.2 then reviews finite-length TEQ performance calculation for a target ν that is presumably less than the actual channel pulse-response length. A teq.m matlab program for assessing design choices also appears there. 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 4.8.4 investigates the joint loading/equalizer design problem and then suggests a solution that could be implemented.

The TEQ's stationarity renders it suboptimum because the DMT/OFDM or VC system is inherently cyclostationary with period $N + \nu$ samples. There are 3 TEQ "simplifications" that may be questioned:

[68] Dr. Jackie Chow - not to be confused with Dr. Peter Chow of loading algorithms - both were Stanford PhD students in the 1990's and crucial Amati Employees in demonstrating the first real working broadband systems based on multicarrier transmission, including xDSL, cable-modems, digital-audio broadcast/Wi-Fi.

1. The equalizer is stationary and should instead be periodic with period $\overline{N} + \nu$.

2. The analysis assumes input sequence x_k to be "white" or i.i.d. and thus ignores any loading that changes the input autocorrelation matrix.

3. The analysis minimizes a scalar mean-square error rather than a multichannel signal-to-noise quantity SNR_{geo}.

Nonetheless, TEQs are often very effective in bringing performance very close to the infinite-length multitone/modal-modulation optimum. Chapter 5's optimum Generalized Decision Feedback Equalizer's (GDFEs) are cyclostationary, or more generally time-variant, and necessarily more complex. The GDFE addresses the above issues, especially 2 and 3[69], but is more complex. For the stationary fixed equalizer, Al-Dhahir [23] and Arslan/Evans [24] have considered solutions to problems 2 and 3, and generally follow a solution posed in Subsection 4.8.4.

The area of block equalization in its various forms is discussed in Subsection 4.8.5. This section visits a structure known as a Block DFE introduced by Kasturia in 1989 [25], but which has unnecessarily high complexity. Discussion then progresses on this basic structure with some practical observations that lead to a Modified FEQ first introduced by Bingham and Cioffi[3], and later re-introduced (with higher complexity, but from another interesting viewpoint) as a "per-tone" equalizer by Van Acker and Moonen [26]. Cheong's precoder[27] also provides an interesting way to implement these structures at the transmit side instead of the receiver.

Wireless systems often have a "short cyclic-prefix option," for which a designer could use a TEQ also (although many wireless designs simply use the long-prefix option on difficult channels instead). Wireline systems like DSL, cable, and powerline's DMT/OFDM designs often do use some form of TEQ.

4.8.1 The infinite-length TEQ

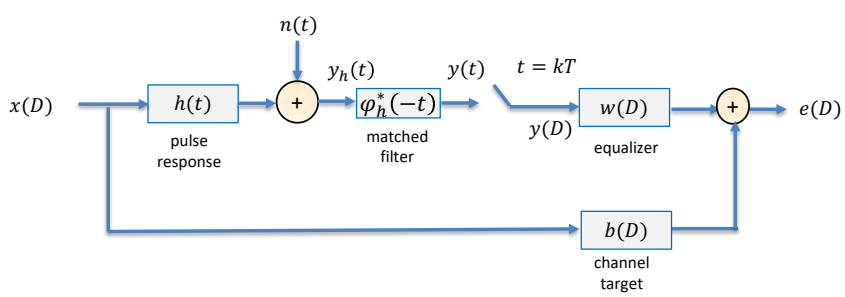


Figure 4.47: The input-aided infinite-length equalization system.

Figure 4.47 illustrates the general baseband complex decision-aided, or in our study **input-aided**, equalization system. The channel output $y_h(t)$ is

$$y_h(t) = \sum_k x_k \cdot h(t - kT) + n_h(t) \quad . \tag{4.336}$$

The receiver convolves the channel output signal $y_h(t)$ with a normalized matched (or anti-alias) filter $\varphi_h^*(-t)$ (where a superscript of $*$ denotes conjugate), samples at rate $1/T$, and then filters with 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 4.47 is equivalent in performance

[69] GDFE theory uses also a guard period of ν samples, but Section 5.7 generalizes to blocks of blocks of samples, which then eliminates finie ν considerations.

to the overall maximum-likelihood detector for the channel by Chapter 1's invertibility theorem, and this alternative equivalently performing ML detector and non-equalized ML detector may have different complexities, depending on $x(D)$ and $b(D)$.

Heuristically, the linear equalizer output appears like the second filter's output, which convolves 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 channel's pulse-response length. The MMSE between the equalizer output and the desired channel output measures equalizer performance. However, 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 $b(D)$ is also questionable unless this simplifies the ML detector. For DMT/OFDM or Vector Coding designs based on $b(D)$, a cyclic prefix of ν samples is essentially sufficient. So the equalizer "shortens" the channel length to the length of $b(D)$, which is Figure 4.46's main function. This method is effective in a large number of practical situations. The next subsection addresses design performance analysis.

4.8.1.1 Minimum Mean-Square Error Equalization

The **error sequence** in Figure 4.47 is

$$e_k = b_k * x_k - w_k * y_k \tag{4.337}$$

where $*$ denotes convolution. The "**signal**" becomes the sequence $b_k * x_k$, which has signal energy $\mathcal{E}_s = \mathcal{E}_{\boldsymbol{x}} \cdot \|b\|^2$. The **D-Transform** of a sequence x_k remains $X(D) \triangleq \sum_k x_k \cdot D^k$. The notation $x^*(D^{-*}) = \sum_k x^*_{-k} \cdot D^{-k}$ represents the sequence's time-reversed conjugate. Thus,

$$E(D) = B(D) \cdot X(D) - W(D) \cdot Y(D) \quad . \tag{4.338}$$

The **minimum mean-square error** is

$$\sigma^2_{\text{MMSE}} = \min_{w_k\ ,\ b_k} \mathbb{E}\left[|e_k|^2\right] \quad . \tag{4.339}$$

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, $\|b\|^2 =$ constant $= \|h\|^2 > 0$. For flat "white" excitation (i.e., $S_x(D) = \bar{\mathcal{E}}_{\boldsymbol{x}}$) of the channel $B(D)$,

$$\text{SNR}'_{\text{MFB}} = \frac{\mathcal{E}_s}{\sigma^2_{\text{MMSE}}} \quad . \tag{4.340}$$

Maximizing SNR'_{MFB} is then equivalent to minimizing σ^2_{MMSE} since $\|b\|^2$ is constant. The TEQ does **not** restrict $B(D)$ to be causal nor monic.

MMSE estimation's orthogonality principle (See Appendix D) 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, $\mathbb{E}\left[e_k \cdot y^*_l\right] = 0\ \forall\ k, l$, which is compactly written

$$\begin{aligned} R_{ey}(D) &= 0 && (4.341)\\ &= \mathbb{E}\left[E(D) \cdot Y^*(D^{-*})\right] && (4.342)\\ &= B(D) \cdot \bar{R}_{xy}(D) - W(D) \cdot \bar{R}_{yy}(D) \quad . && (4.343) \end{aligned}$$

then,

$$W(D) = B(D) \cdot \frac{R_{xy}(D)}{R_{yy}(D)} \quad . \tag{4.344}$$

Then,

$$Y(D) = \|h\| \cdot Q(D) \cdot X(D) + n(D) \quad , \tag{4.345}$$

where $Q(D) \triangleq \frac{1}{2\pi\|h\|^2}\int_{-\pi}^{\pi}\sum_n |H\left(\frac{\omega-2\pi n}{T}\right)|^2 d\omega$. Then,

$$\bar{R}_{xy}(D) = \bar{\mathcal{E}}_{\boldsymbol{x}} \cdot \|h\| \cdot Q(D) \tag{4.346}$$

$$\bar{R}_{yy}(D) = \bar{\mathcal{E}}_{\boldsymbol{x}} \cdot \|h\|^2 \cdot Q(D) \cdot \left(Q(D) + \frac{1}{\text{SNR}_{\text{MFB}}}\right) \quad . \tag{4.347}$$

Recalling the canonical factorization from Chapter 3 DFE design,

$$Q(D) + \frac{1}{\text{SNR}_{\text{MFB}}} = \gamma_0 \cdot G(D) \cdot G^*(D^{-*}) \quad . \tag{4.348}$$

This factorization is sometimes called the "key equation."

The error autocorrelation sequence is

$$R_{ee}(D) = B(D) \cdot \left(R_{xx}(D) - \frac{R_{xy}(D)}{R_{yy}(D)} \cdot R_{yx}(D)\right) \cdot B^*(D^{-*}) \quad , \tag{4.349}$$

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

$$R_{ee}(D) = \left(\frac{\bar{\mathcal{E}}_{\boldsymbol{x}}}{\gamma_0 \cdot \text{SNR}_{\text{MFB}}}\right) \cdot \frac{B(D) \cdot B^*(D^{-*})}{G(D) \cdot G^*(D^{-*})} \quad . \tag{4.350}$$

Defining the (squared) norm of a sequence as

$$\|x\|^2 \triangleq \sum_{k=-\infty}^{\infty} |x_k|^2 \quad , \tag{4.351}$$

the mean square error from (4.350) is

$$\text{MSE} = \left(\frac{\bar{\mathcal{E}}_{\boldsymbol{x}}}{\gamma_0 \cdot \text{SNR}_{\text{MFB}}}\right) \cdot \|\frac{b}{g}\|^2 \quad . \tag{4.352}$$

From the Cauchy-Schwarz lemma, we know

$$\|b\|^2 = \left\|\frac{b}{g} \cdot g\right\|^2 < \left\|\frac{b}{g}\right\|^2 \cdot \|g\|^2 \tag{4.353}$$

or

$$\left\|\frac{b}{g}\right\|^2 > \frac{\|b\|^2}{\|g\|^2} \quad . \tag{4.354}$$

Thus, the MMSE is

$$\sigma^2_{\text{MMSE}} = \left(\frac{\bar{\mathcal{E}}_{\boldsymbol{x}}}{\gamma_0 \cdot \text{SNR}_{\text{MFB}}}\right) \quad , \tag{4.355}$$

which can be achieved by a multitude of $B(D)$ if the degree of $G(D)$ is less than or equal to ν[70], only one of which is causal and monic (Chapter 3's MMSE-DFE).

The MMSE solution has any $B(D)$ that satisfies

$$B(D) \cdot B^*(D^{-*}) = G(D) \cdot G^*(D^{-*}) \quad . \tag{4.356}$$

From (4.344) to (4.348), one finds $W(D)$ from $B(D)$ as

$$W(D) = \frac{1}{\gamma_0 \cdot \|h\|} \frac{B(D)}{G(D) \cdot G^*(D^{-*})} \quad , \tag{4.357}$$

which is always invertible when $\frac{\mathcal{N}_0}{2} > 0$. The various solutions differ in phase only. Such a solution has bias, but its address is different than in Chapter 3's suboptimal SBS detection.

[70] If the degree is greater than ν, then the methods of Section 4.8.2 must instead be used directly.

4.8.1.2 Solutions

As mentioned earlier, there are many choices for $B(D)$, each providing then a corresponding TEQ $W(D)$ in Equation (4.357).

The MMSE-DFE: Chapter 3's Minimum-Mean-Square Error Decision Feedback Equalizer (MMSE-DFE) chooses causal monic $B(D)$ as

$$B(D) = G(D) \text{ so } W(D) = \frac{1}{\gamma_0 \cdot \|h\| \cdot G^*(D^{-*})} \quad , \tag{4.358}$$

is anti-causal.

In ML detection with DMT/OFDM or VC, unlike the MMSE-DFE, the TEQ does not implement the feedback filter $B(D)$, avoiding any issue of error propagation or need for precoders and their correspodning loss. That ML detector is instead easily implemented for DMT/OFDM or VC because they essentially partition the $B(D)$ channel into a set of parallel independent channels, each of which has an easier ML detector

The MMSE-AR Equalizer: The MMSE Auto Regressive Equalizer (MMSE-ARE) corresponds to the anti-causal monic $B(D)$ choice

$$B(D) = G^*(D^{-*}) \text{ so } W(D) = \frac{1}{\gamma_0 \cdot \|h\| \cdot G(D)} \quad , \tag{4.359}$$

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 (4.359) 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. So the MMSE-DFE and the MMSE-ARE raise the question of which $B(D)$ to choose? This leads to various approximation methods for the choice of $B(D)$ that improves performance the most, while keeping the guard-period length manageable.

4.8.1.3 Comparison of MMSE-DFE and MMSE-ARE

Both the MMSE-DFE and the MMSE-ARE choices are invertible (with enough delay) and thus have the same performance (with ML detection). Essentially the causality/noncausality conditions of the feedforward filters and feedback filters have been interchanged. Because the actual detector never implements $B(D)$ (which is only for the TEQ $W(D)$'s design calculations), either structure could be used in a practical situation. However, the TEQ length (complexity) is also important.

A finite-length TEQ might have very poor choice for a given channel with some $B(D)$'s of the same length, but very good performance for others. 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 system is one that chooses $B(D)$ and $W(D)$ to be both mixed-phase also. The designer's challenge then becomes, for a finite-length feedforward section ($W(D)$) of L 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), particularly when this is a DMT/OFDM or VC overall system design. Subsection 4.8.2 addresses this design challenge.

4.8.2 The finite-length TEQ

Like the Chapter 3's finite-length MMSE-DFE, the finite-length MMSE-TEQ problem has an error

$$\boldsymbol{e}_k(\Delta) = \boldsymbol{b}^* \cdot \boldsymbol{x}_{k-\Delta} - \boldsymbol{w}^* \cdot \boldsymbol{y}_k \tag{4.360}$$

where

$$\boldsymbol{x}_{k-\Delta} \triangleq \begin{bmatrix} x_{k-\Delta} \\ x_{k-\Delta-1} \\ \vdots \\ x_{k-\Delta-\nu} \end{bmatrix} \quad ; \tag{4.361}$$

$\boldsymbol{b}^* \triangleq [b_0^* \; b_1^* \; ... \; b_\nu^*]$ is not necessarily causal, monic, nor minimum-phase; and $\boldsymbol{w}^* \triangleq [w_0^* \; w_1^* \; ... \; w_{L-1}^*]$, L is the number of equalizer coefficients. The equalizer implementation may be at another sampling rate for the fractionally spaced case[71]; and

$$\boldsymbol{y}_k \triangleq \begin{bmatrix} y_k \\ y_{k-1} \\ \vdots \\ y_{k-L+1} \end{bmatrix} \quad . \tag{4.362}$$

The length ν is typically less than the true channel pulse-response length and equal to the value anticipated for use in a system like DMT/OFDM or VC.

The input/output relationship of the channel is

$$\boldsymbol{y}_k = H \cdot \boldsymbol{x}_k + \boldsymbol{n}_k \quad . \tag{4.363}$$

H need not be a convolution matrix and can reflect whatever guard bands or other constraints have been placed upon the transmit signal.

The MMSE TEQ: The MSE is minimized with a constraint of $\|\boldsymbol{b}\|^2 = c$ a constant, with c usually chosen as $c = \|h\|^2$ to force the flat-input channel's SNR_{MFB} to be the flat-input TEQ system's SNR_{MFB} . 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

$$\mathbb{E}[\boldsymbol{e}(\Delta) \cdot \boldsymbol{y}_k^*] = 0 \quad . \tag{4.364}$$

The solution is

$$\boldsymbol{w}^* = \boldsymbol{b}^* \cdot \boldsymbol{R}_{\boldsymbol{xy}}(\Delta) \cdot \boldsymbol{R}_{\boldsymbol{yy}}{}^{-1} \tag{4.365}$$

for any $\boldsymbol{b}$. Then, $\boldsymbol{e}(\Delta)$ becomes

$$\boldsymbol{e}(\Delta) = \boldsymbol{b}^* \cdot \left[\boldsymbol{x}_{k-\Delta} - \boldsymbol{R}_{\boldsymbol{xy}}(\Delta) \cdot \boldsymbol{R}_{\boldsymbol{yy}}{}^{-1} \boldsymbol{y}\right] \quad . \tag{4.366}$$

(4.366) has the minimized MSE finite-length LE vector within the square brackets. This MMSE-LE error vector has an autocorrelation matrix that is

$$R_{LE}(\Delta) = \mathcal{E}_{\boldsymbol{x}} \cdot I - \boldsymbol{R}_{\boldsymbol{xy}}(\Delta) \cdot \boldsymbol{R}_{\boldsymbol{yy}}{}^{-1} \cdot \boldsymbol{R}_{\boldsymbol{yx}}(\Delta) \quad . \tag{4.367}$$

Tacit in (4.367) is the assumption that $\boldsymbol{R}_{\boldsymbol{yy}}$ is a fixed $L \times L$ matrix, which is equivalent to assuming that the input to the channel is stationary along with 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 TEQ solution calculations occur before transmit optimization, usually

[71] 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, with very short equalizers with n oversampling factor greater than 1 can improve performance for fixed complexity, depending on channel details.

with a diagonal matrix, $\boldsymbol{R_{xx}} = [\bar{\mathcal{E}}_{\boldsymbol{x}}/(\overline{N}+\nu)] \cdot I$. With these restrictions/assumptions, the MMSE is then

$$MMSE = \min_{\boldsymbol{b},\Delta} \boldsymbol{b}^* \cdot R_{LE}(\Delta) \cdot \boldsymbol{b} \quad . \tag{4.368}$$

with the constraint that $\|\boldsymbol{b}\|^2 = c$. The minimization's solution is the eigenvector of $\boldsymbol{R_{ee}}$ that corresponds to the minimum eigenvalue. Thus

$$\boldsymbol{b} = \sqrt{c} \cdot \boldsymbol{q}_{min}(\Delta) \quad . \tag{4.369}$$

Equation (4.369) remains a function of the delay Δ. TEQ design then evaluates several reasonable delay values to find the minimum, now over Δ – so in addition to the minimization over eigenvalues for each Δ. That final delay choice is Δ_{opt}.

Final design analysis and bias address: Bias emoval best directly addresses the equalized channel $b_k \stackrel{\Delta}{=} \{w_k * h_k\}_{k=\Delta_{opt}\ldots\Delta_{opt}+\nu}$ because the use of, for instance, water-filling spectra, maximum-likelihood decoding, and noise coloring prevent the simple scalar bias removal's meaning. So, the design evaluates directly the equalized channel's ν largest values that correspond to the Δ_{opt} and corresponding w_k equalizer. This redefined b_k is not quite equal to the optimum (scaled minimum-eigenvalue eigenvector) but is what any DMT design will use. There is correspondingly an **ISI-error signal** that is

$$\beta_k \stackrel{\Delta}{=} b_k - w_k * h_k \quad . \tag{4.370}$$

This ISI-error sequence is not necessarily white and has (deterministic) autocorrelation $R_{\beta,k} = \beta_k * \beta^*_{-k}$. Similarly the TEQ colors the input white noise[72], then $R_{w,k} = w_k * w^*_{-k}$. The channel gains are then found for use in the TEQ'd channel's loading are the ratio $|FFT(R_{b,k})|$ to the sum $|FFT(R_{\beta,k})|+|FFT(R_{w,k})|$. With these gains, waterfill or other loading optimization can follow. Approximating this noise by "white noise" with the same variance will severely bias results. The correct channel gains should be used.

The $L \times L$ matrix $\boldsymbol{R_{yy}}$ need only be inverted once, The eigenvectors must be computed for all reasonable values of Δ since $[0...010...0] \cdot H$ depends on Δ. This computation can be intensive. For FIR channels, the matrix $\boldsymbol{R_{yy}}$ is given by

$$\boldsymbol{R_{yy}} = H \cdot \boldsymbol{R_{xx}} \cdot H^* + R_{\boldsymbol{nn}} \text{ general non-white input case} \tag{4.371}$$

$$= \bar{\mathcal{E}}_{\boldsymbol{x}} \cdot H \cdot H^* + R_{\boldsymbol{nn}} \text{ usual white-input case.} \tag{4.372}$$

Any valid $\boldsymbol{R_{xx}}$ can be used. The cross-correlation matrix is

$$\boldsymbol{R_{xy}}(\Delta) = E\left[\boldsymbol{x}_{k-\Delta} \cdot \boldsymbol{y}_k\right] = E\left[\boldsymbol{x}_{k-\Delta} \cdot \boldsymbol{x}_k^*\right] \cdot H^* \quad . \tag{4.373}$$

The cross-correlation matrix will have zeros in many positions determined by Δ unless again the input is optimum.

Matlab teq.m: The teq.m program (See Appendix G for full listing) finds the TEQ, and then proceeds to convolve it with the channel, compute both the residual ISI energy's power spectral density and the filtered noise's power spectral density, summing them, and then using as the noise power-spectral-density in a water-filling design program (waterfill_gn.m) and appropriately computes SNR's will all input dimensions included (so $N+\nu$). N and ν are inputs, as is L, the number of filter taps. The program internally uses a large FFT size (10,000) to approximate the infinite DMT design performance. The program also provides (white input) SNR_{MFB} for reference. The teq.m program also provides the TEQ-equalized DMT system's number of used dimensions.

```
[w, b, SNRmfb, SNRdmt_teq, SNRdmt, Nstar]=teq(h,L,nu,N,noise,Ex_bar,gap,cb)

programmed by J.M. Cioffi
Calculates a time domain equalizer and calculates the SNR for an infinite
```

[72] If the channel noise is not white and has then $R_{nn,k}$, then this filtered noise has autocorrelation $w_k * R_{nn,k} * w^*_{-k}$, and $\boldsymbol{R_{yy}}$ includes this noise also.

```
length DMT system to compare

Inputs
------
h: channel impulse response
(FIR channels: p= 1+0.5D+2.3D^2->p=[1 0.5 2.3]
IIR channels:  p=1/(2.4-0.5D)-> p=[2.4 -0.5])
L: number of equalizer taps (should be < N/2)
nu: target channel pulse response length < size(h)-1
N:  FFT size for eventual computation of SNRdmt_teq
noise: noise variance per sample
Ex_bar: Input energy per sample
cb = 2 for real, 1 for complex

Outputs
-------
w: teq equalizer
b: MMSE target channel shaping function (length=nu+1)
SNRmfb: matched filter bound SNR
SNRdmt_teq: DMT SNR for this equalized channel for N input (dB scale)
SNRdmt: DMT SNR for the unequalized first nu+1 samples of the channel
        Nbig=10000 (dB scale)
Nopt:  Number of used dimensions

Remarks
-------
calls "DMTra.m" and "waterfill_gn" functions

****************************************************
```

The following example illustrates the utility and benefit of the TEQ:

EXAMPLE 4.8.1 *[Mixed Phase Lowpass Channel with TEQ]* The real baseband channel below is mixed phase (which the reader can verify using matlab's roots command). The FFT size is 128, so relatively small and $L = 9$ and $\nu = 2$ for this channel, which has white noise with $\frac{\mathcal{N}_0}{2} = 0.1$.

```
>> h3 =[  0.7200    2.9240    4.4084    2.9240    0.7200 ];
>> L=9;
N=128;
nu=2;
noise=0.1;
>> [w, b, SNRmfb, SNRdmt_teq, SNRdmt, Nopt]=teq(h3,L,nu,N,noise,(N/(N+nu)),gap,2)

w =   -0.0741    0.2353   -0.3408    0.0565    1.1099    0.0565   -0.3408    0.2353   -0.0741

b =    2.7264    4.7648    2.7264

SNRmfb =   25.6811 dB
SNRdmt_teq =   17.0804 dB
SNRdmt =   17.4460 dB

Nopt =   120
```

Almost all dimensions (120 of 128 maximum) carry energy, which has been reduced on the teq.m input to model the target cyclic-prefix energy loss for the DMT system use[73]. $\Gamma = 0$ dB. The equalized channel with $\nu = 2$ approaches the infinite-length result, missing by only .35 dB. The MFB is not attainable on this channel. An alternative FFT size of $N = 1024 << 10,000$ would also get the maximum SNR on this channel (the reader can verify this using DMTra.m), which is complexity $log_2(N = 128) = 7$ per sample, half at transmitter and half at receiver. This is the lowest complexity. Reducing FFT size and adding an equalizer increases complexity, but will reduce delay and memory/storage requirements. The TEQ here is 9+7=16 calculations per sample.

A MMSE-DFE also provides good comparison at equal complexity to the TEQ, which could use 13 FF taps and 4 FB taps, all at receiver.

```
>> [dfseSNR,w_t,delay] = dfecolor(1,h3,13,4,-1,1,noise*[1,zeros(1,12)])

dfseSNR =   16.5967 dB
w =
    0.0023   -0.0059    0.0059    0.0028   -0.0203    0.0375   -0.0365   -0.0001
    0.0696   -0.1242    0.0518    0.3257    0.1543
b = 1 1.9488   1.6697   0.6856   0.1111
delay =    12
```

The MMSE-DFE at same complexity level underperforms the TEQ system by roughly 0.5 dB, and the full DMT system by almost 1 dB. A system with this SNR would likely use a Tomlinson Precoder and lose roughly another 1 dB.

The example illustrates the basics, but a more creative next example investigates an IIR channel model.

EXAMPLE 4.8.2 *[IIR Channel $1/1 - .9 \cdot D$]* 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

$$H(D) = \frac{1}{1 - .9 \cdot D} \quad , \tag{4.374}$$

with time-domain response easily found to be

$$h_k = .9^k \;\; \forall \;\; k \geq 0 \quad . \tag{4.375}$$

The gain of the channel is $\|h\|^2 = \frac{1}{1-.9^2} = 5.2632$. Let the AWGN variance per real dimension be $\sigma^2 = .1$ and the input energy be $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$. With PAM on this real baseband channel, $\text{SNR}_{\text{MFB}} = 17.2$ dB. The guard-band energy varies with ν. The total energy outside of a guard period is then

$$\|h\|^2 - \sum_{k=0}^{\nu} \frac{1 - (.81)^{\nu+1}}{1 - .81} \quad . \tag{4.376}$$

This lost energy, which becomes distortion for the DMT or VC system is in Table 4.18

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 4.18: Energy loss by truncation (rectangular windowing) of $h_k = .9^k \cdot u_k$.

To have only .1 dB loss, then $\nu > 16$, meaning $\overline{N} > 500$ for small loss in energy and bandwidth in a DMT/OFDM or VC design. Alternatively, a TEQ can reshape the channel

[73] TEQ use with VC would not include this reduction

to place 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 $1 - .9 \cdot D$ will reduce ν substantially while altering the noise. Thus, a TEQ 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

$$\boldsymbol{R_{xy}}(0) = \begin{bmatrix} 1 & 0 & 0 \\ .9 & 1 & 0 \end{bmatrix} \tag{4.377}$$

and

$$\boldsymbol{R_{yy}} = \begin{bmatrix} 5.3636 & .9(5.2636) & .81(5.2636) \\ .9(5.2636) & 5.3636 & .9(5.2636) \\ .81(5.2636) & .9(5.2636) & 5.3636 \end{bmatrix} \quad . \tag{4.378}$$

The error autocorrelation matrix is

$$\boldsymbol{R_{ee}} = \begin{bmatrix} .1483 & -.0650 \\ -.0650 & .1472 \end{bmatrix} \quad , \tag{4.379}$$

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 4.18 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 (4.369) is

$$B(D) = \|h\| \cdot [.7041\ .7101] = \sqrt{5.2632} \cdot [.7041\ .7101] = 1.6151 \cdot [1 + 1.0084 \cdot D] \quad . \tag{4.380}$$

$B(D)$ is not minimum phase in this example, illustrating that the TEQ is not simply a "DFE solution." The corresponding value for $\boldsymbol{w}$ from Equation (4.365) leads to $W(D) = 1.4803 \cdot (1 + .1084 \cdot D - .8907 \cdot D^2)$ and a new equalized channel

$$\begin{aligned} z(D) &= 1.4803 \cdot \left(1 + .1084D - .8907D^2\right) \cdot \frac{X(D)}{1 - .9D} + W(D) \cdot N(D) && (4.381) \\ &= 1.4803 \left[1 + (.1084 + .9)D + (.81 + .9 \cdot .1084 - .8907) \cdot \frac{D^2}{1 - .9D}\right] \cdot X(D) + W \cdot N \\ &= \underbrace{1.4803\,[1 + (.1084 + .9)D]}_{\frac{w_0}{b_0}B(D)} + \underbrace{1.4803(.01686) \cdot \frac{X(D) \cdot D^2}{1 - .9D} + W(D) \cdot N(D)}_{\frac{w_0}{b_0}E(D)} \quad . \end{aligned}$$

The more complicated bias analysis is not necessary with this "one-pole" channel. Equation (4.381)'s 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 actually not necessary in DMT systems because the system's multichannel normalizer automatically includes any such adjustment. The distortion's PSD is as

$$\begin{aligned} R_{EE}(D) \cdot \left(\frac{w_0}{b_0}\right)^2 &= 1.4803^2 \cdot (.01686)^2 \frac{1}{1 - .9^2} + \cdot .1 \cdot W(D) \cdot W^*(D^{-*}) && (4.382) \\ &= 1.4803^2 \cdot \left[(.001496) + .1 \cdot \left(1 + .1084^2 + .8907^2\right)\right] && (4.383) \\ r_{EE}(0) \cdot \left(\frac{w_0}{b_0}\right)^2 &= .3988 = .4538 \cdot \left(\frac{w_0}{b_0}\right)^2 \quad . && (4.384) \end{aligned}$$

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. 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 $\overline{N}$ in VC or DMT. The equalized channel is shown in Figure 4.48 where it is clear the first two positions

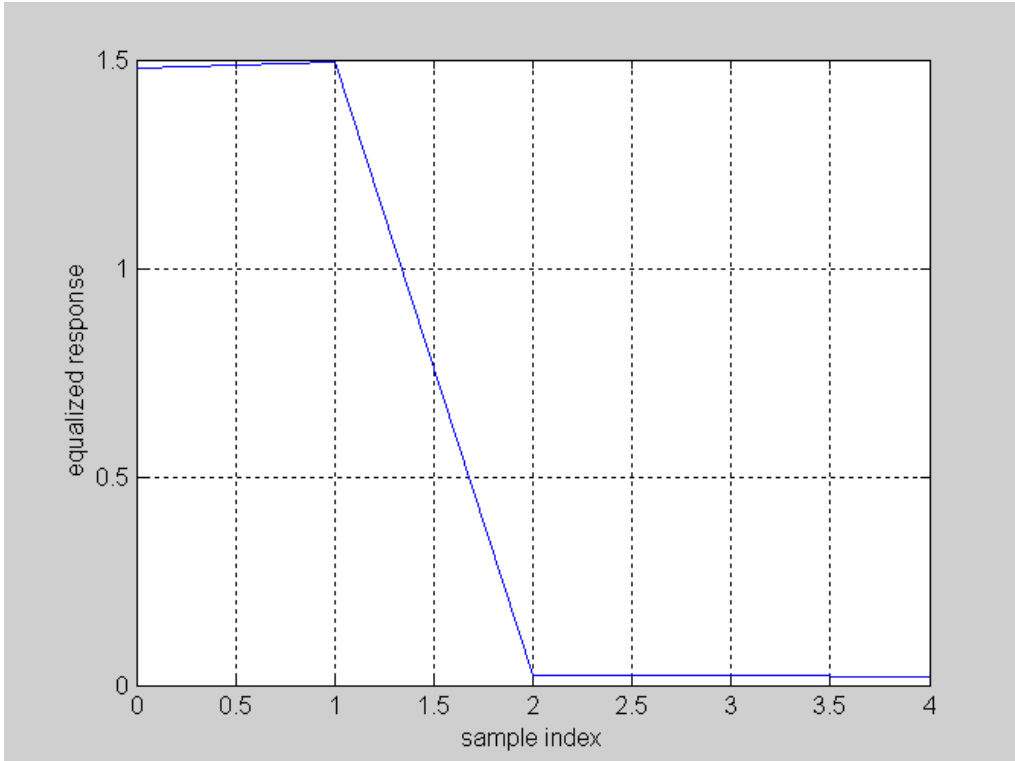


Figure 4.48: Equalized response for 3-tap TEQ with $1/(1-.9D)$ channel.

contain virtually all the energy. Figure 4.49 shows the error signal's power spectrum, 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 4.18 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 **??**.) The TEQ is not a DFE and does not maintain white noise nor channel magnitude in the finite-length case.

The following third example shows a near equivalence of a DFE and an equalized DMT system when both use exactly the same bandwidth as per the separation theorem.

EXAMPLE 4.8.3 *[Full-band TEQ]* A 7-tap FIR channel $h = [.729\ .81\ -.9\ 2\ -.9\ .81\ .729]$ 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 DMT tones carry energy. 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:

```
>>h=[ .729 .81 -.9 2 .9 .81 .729];
>> [snr,w]=dfecolor(1,h,20,6,-1,1,[.1 zeros(1,19)])

snr =   17.9909 dB

w =
    0.0036    0.0047    0.0012   -0.0064   -0.0122   -0.0102    0.0026    0.0208
    0.0313    0.0184   -0.0182   -0.0630   -0.0692   -0.0311    0.0922    0.1285
    0.2378   -0.1743    0.1248    0.1140
```

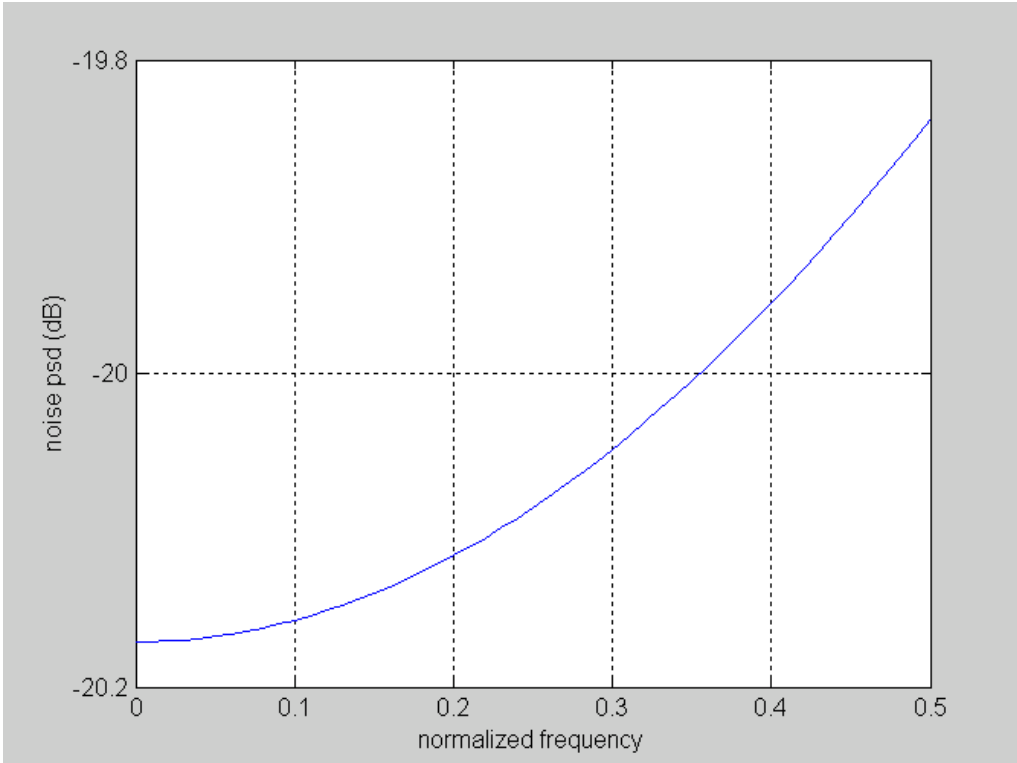


Figure 4.49: Distortion Spectrum for 3-tap TEQ on $1/(1-.9D)$ channel with white noise at $\sigma^2 = .1$.

```
b=  1   0.0168   0.2766   0.3725   0.0767   0.1833   0.0831
```

More equalizer taps do not appear to improve this DFE further. There will be precoder or error-progation loss of about 1 dB or so in practice, the best DFE SNR is really about 17 dB. An equal performing TEQ-DMT system occurs at about the same complexity with $L = 19$

```
>> nu =2;
>> L=19;
>> N=128;
>> [w, b, SNRmfb, SNRdmt_teq, SNRdmt, Nopt]=teq(h,L,nu,N,noise,(N/(N+nu)),gap,2)
w =
    0.0238    0.0209   -0.0059   -0.0543   -0.0574   -0.0498    0.0978    0.0490    0.3614   -0
b =
    1.5871   -1.4710    1.8200
SNRmfb =18.9609
SNRdmt_teq =   17.0380
SNRdmt =   17.9966
Nopt =   128
```

The infinite-length DMT system's 18 dB will be attained at $N = 1024$ will less complexity than either of the equalized systems (DFE or TEQ), but has longer delay. All dimensions carry energy, hence the separation theorem applies well, and thus the systems are basically all operating equally if then DFE's error-propagation/precoder loss is ignored, but the TEQ does not really help too much. (A longer TEQ will achieve the 18 dB at shorter FFT lengths if the setting $\nu = 0$ is attempted; essentially, these TEQ's invert the channel and water-filling reduces energy where there is noise enhancement.)

MIMO TEQ MIMO Channels use the same ν on all spatial-dimension inputs to their respective time-frequency channels. Essentially, every antenna input has the same-size guard period (cyclic prefix). In practice if this common ν is to be shortened, a receiver TEQ is necessary on all receive antennas that targets this ν. The common ν assures that Vector DMT systems have independent tonal channels. A common large matrix TEQ parallels the above development for Section 3.10's MIMO-DFE leads to a very complex problem to implement. Typically a scalar TEQ for each receiver antenna input, prior to its DFT implementation for coded-DMT or C-OFDM will suffice to reduce distortion. A scalar MMSE TEQ design includes any crosstalk from adjacent channels outside the $\nu + 1$ target interval can be used. However, each and every antenna's receiver design must consider a single common Δ_{opt}, or accommodate different Δ's in the alignment of symbol boundaries upon which the receiver FFT's act. Thus, the TEQ step of optimizing over the best Δ considers all block MMSE-dimensions' $trace\{\boldsymbol{R_{ee}}\}$ or equivalently $|\boldsymbol{R_{ee}}|$ optimization.

4.8.3 Zero-forcing Approximations of the TEQ

There are several alternative approaches to TEQ design, 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 $W(D) \cdot H(D)$ within a "window" of $\nu + 1$ samples and the energy outside that window in the other $N - 1$ samples, which second set of samples is called the "wall." The heurestic rationale is that the wall's energy should be minimum relative to the window's energy. 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, the approach ignores noise. 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 $c \cdot \bar{\mathcal{E}}_{\boldsymbol{x}} \cdot \|h\|^2$ in the MMSE-TEQ when the channel input is white. The problem of maximizing window to wall energy is the same as maximizing total equalized pulse-reponse 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 possibility among those considered for optimization, and thus is always the same or better.

EXAMPLE 4.8.4 *[Return to $1/(1 - .9 \cdot D)$ channel example of Section 4.8.2]* A designer might use a zero-forcing TEQ designs instead on the $\frac{1}{1-.9 \cdot D}$ channel. Zero-forcing ignores noise, so then a best equalizer is $w(D) = \sqrt{5.2636} \cdot (1 - .9 \cdot D)$, which flattens the channel so that even $\nu = 0$ is sufficient. In fact all zero-forcing equalizers for $\nu = 1$ follow the class $W(D) = (a + b \cdot D) \cdot (1 - .9 \cdot D)$ where $a^2 + b^2 = \|h\|^2$. The first zero-forcing equalizer has smallest enhanced noise given by .9527 relative to $\|h\|^2 = 5.2636$ 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 $W(D) = (1.65 - 1.59 \cdot D) \cdot (1 - .9D)$.[74], and has the same ratio instead of 10 dB, which is still less than 10.4 dB found with MMSE. In fact, no higher SNR 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 example reinforces the observation that it is really SNR_{dmt} that is important then leads to the Subsection 4.8.5

4.8.4 Joint Loading-TEQ Design

Usually TEQ design occurs before DMT (or VC) loading with an assumption of initially flat input spectrum $S_x(D) = \bar{\mathcal{E}}_{\boldsymbol{x}}$. For OFDM, this assumption is always true, but when loading is used in DMT, then it will likely not be true. In fact, an optimum design would maximize SNR_{dmt} over the TEQ settings

[74] 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 $-.626$ ($a = 2.21$).

as well as the $\mathcal{E}_n$ for each subchannel. Al-Dhahir [23] was the first to approach this problem and attempt solution. A better solution emanated from Evans and his colleagues [24]. While Evans beautifully poses the problem, he notes at the end that it requires very difficult and advanced optimization procedures must be precisely solved with large numerical precision. Evans 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) $\boldsymbol{R_{xx}}$, 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 defficient. 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 synchronizes the change of energy and bit tables in the corresponding modems, so thus should be synchronized to bit swapping.

4.8.5 Block Equalization for DMT

There are also some block equalization methods that well address the confinement of ν to a small value.

4.8.5.1 Block Decision Feedback

The overhead of the extra ν dimensions of vector coding and and DMT/OFDM is negligible for large $\overline{N}$. When large $\overline{N}$ leads to unacceptable decoding delay, smaller $\overline{N}$ becomes attractive. In this case, Kasturia's Block Decision Feedback Equalizer[75] (Figure 4.50, [25]) eliminates the extra ν samples but introduces an extra $\mathcal{O}(\overline{N} \cdot \nu)$ computational requirement, whichcharacterizes the feedback section's complexity.

[75]The original concept was introduced by Dr. Sanjay Kasturia in his 1989 Dissertation, but modified here considerably since that introduction.

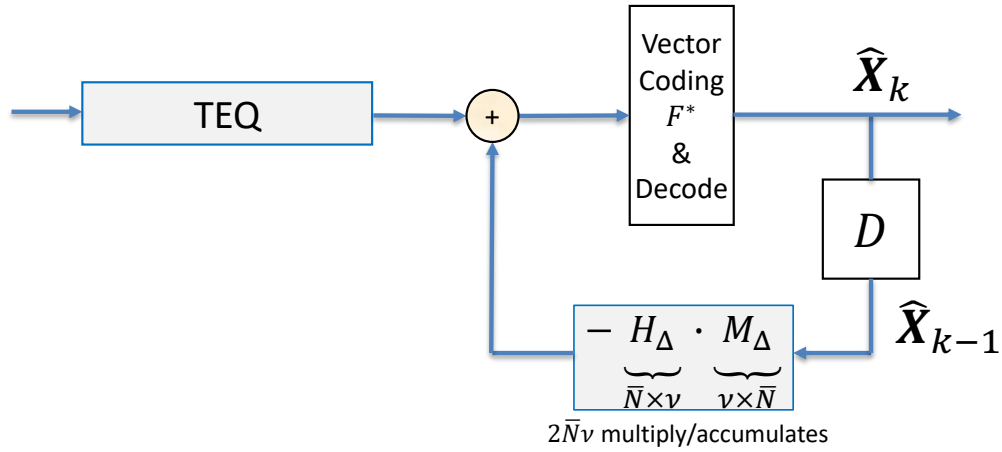


Figure 4.50: Kasturia's Block DFE.

Vector Coding Version: The channel matrix H partitions into a square matrix and the residual last ν columns that model the ISI:

$$H = \left[\underbrace{H_{block}}_{\overline{N}\times\overline{N}} \vdots \underbrace{H_\Delta}_{\overline{N}\times\nu} \right] . \tag{4.385}$$

The component $\hat{\boldsymbol{y}}_{k-1} = H_\Delta \cdot \boldsymbol{x}_{k-1}$ corresponds to ISI from the previous (symbol time $k-1$) symbol. The SVD is now $H_{block} = F \cdot \Lambda \cdot M^*$. If the previous decision is correct, then $\boldsymbol{x}_{k-1} = M \cdot \boldsymbol{X}_{k-1}$, and the BDFE regenerates this from the decoded output $\hat{\boldsymbol{X}}_{k-1} = \boldsymbol{X}_{k-1}$, the previous vector of decisions. The multiplication by M to get $\boldsymbol{x}_{k-1}$ need only compute the ν time-domain samples corresponding to the intersymbol interference from the previous symbol, which is actually $\nu \times \overline{N}$ matrix multiply (so requiring less than $\overline{N}^2$ calculations since $\nu < N$ in practice. Figure 4.50 shows this matrix as M_Δ and corresponds to the upper ν rows of M. The ν output samples then feed the $\overline{N} \times \nu$ matrix mutiply H_Δ that reconstructs the previous symbol's ISI, and the BDFE subtracts this from the TEQ output signal. The TEQ reduces the required ν to reduce computational complexity, even though the BDFE no longer uses any guard period samples.

Adaptive BDFE for DMT/OFDM: Figure 4.51 illustrates a DMT-BDFE alternative. The matrix filter W in what was previously the "TEQ position" has $\overline{N}+L$ inputs and $\overline{N}$ outputs. L can be viewed as a number of additional time-domain input samples (some before and after the $\overline{N}+\nu$ block of output samples so typically $L > \nu$. (This L should not be confused with Chapter 7's training-sequence length nor the number of antennas elsewhere in this text.) A matrix filter is equivalent to a time-varying scalar filter with different coefficients for each of the generated output samples and need not be Toeplitz (so zeros in some places before and after the set within the span upon which the filter coefficients act, but otherwise the same convolutional structure as the matrix channel H). This W minimizes the mean squared error for the corresponding output compared to the transmitted symbol on each tone, as shown in Figure 4.51 as (the mean square of) $\boldsymbol{E}_n$. There can also be a decision feedback matrix (that could be a full $\overline{N} \times \overline{N}$ tone-to-all-other-tones construction and subtraction of trailing intersymbol interference from the last block symbol sent).

The equalizer coefficients in the nth row are updated according to a MMSE/stochastic adaptive algorithm as

$$\boldsymbol{w}_{n,k+1} = \boldsymbol{w}_{n,k} + \mu \cdot E_{n,k} \cdot \boldsymbol{Y}^*_{n,k} \tag{4.386}$$

$$\boldsymbol{B}_{n,k+1} = \boldsymbol{B}_{n,k} + \mu_b \cdot E_{n,k} \cdot \hat{\boldsymbol{X}}^*_{k-1} \tag{4.387}$$

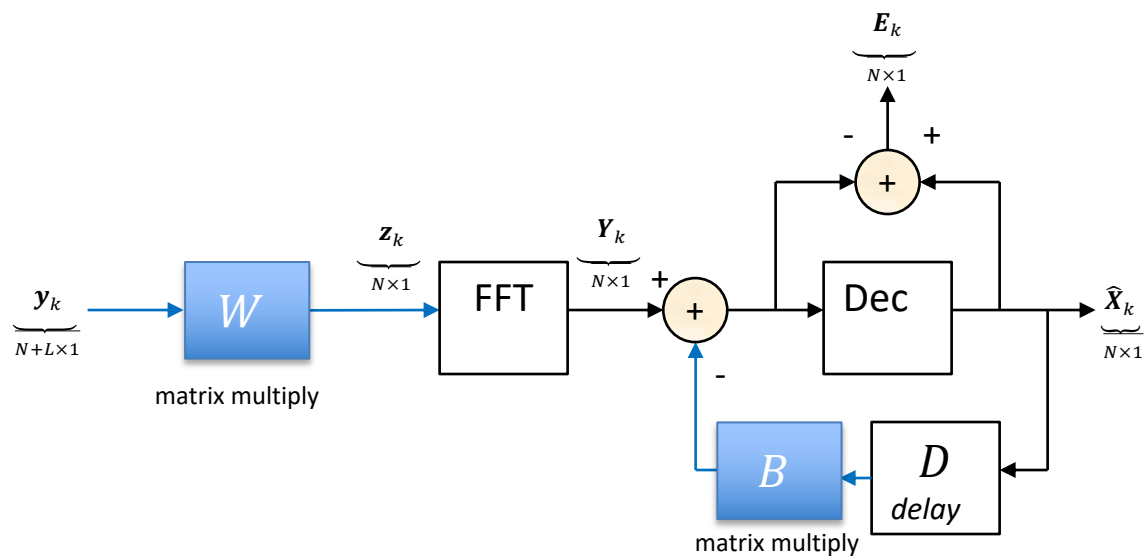


Figure 4.51: Mixed time/freq Block DFE

where $\boldsymbol{Y}_{n,k}$ is the block of input samples to the nth row of $\boldsymbol{W}_k$ (for symbol time k) and $\boldsymbol{X}_k$ (or $\hat{\boldsymbol{X}}_{n,k}$ when decisions are used instead of actual training values) is the $\overline{N}\times 1$ vector of input symbol values. The implementation of the "time-varying" (row-varying), equivalently matrix multiply, $\boldsymbol{W}$ clearly improves over the TEQ, which is a special case. However, Van Acker and Moonen [26] note that this structure is equivalent to a "sliding-block" FFT-out frequency-domain coefficient-variable convolution, and that in addition to a scalar W_n on each tone that the only other contributions to the nth tone would be (with $L = L_{for} + L_{back} + 1$ and L_{for} being the number of samples used in the equalizer that reach into the future with respect to nominal time 0 (delay implied) and L_{back} being the number of samples used in the equalizer similarly reaching backward in time.):

$$y_{N+L_{for}-1} \quad - \quad y_{Lfor-1} \tag{4.388}$$

$$y_{N+L_{for}-2} \quad - \quad y_{Lfor-2} \tag{4.389}$$

$$\vdots \quad \vdots \quad \vdots \tag{4.390}$$

$$y_{N+L_{back}+1} \quad - \quad y_{Lback+1} \tag{4.391}$$

These channel-output-sample differences characterize the deviation from cyclic convolution, while the cyclic part of the convolution corresponds directly a single-tap/tone FEQ. Thus, Figure 4.52 illustrates the coefficients with filter weightings of the equivalent of $\boldsymbol{W}$ contributing on each tone through these differences above. This was called a "Per-tone Equalizer" by Van Acker and Moonen.

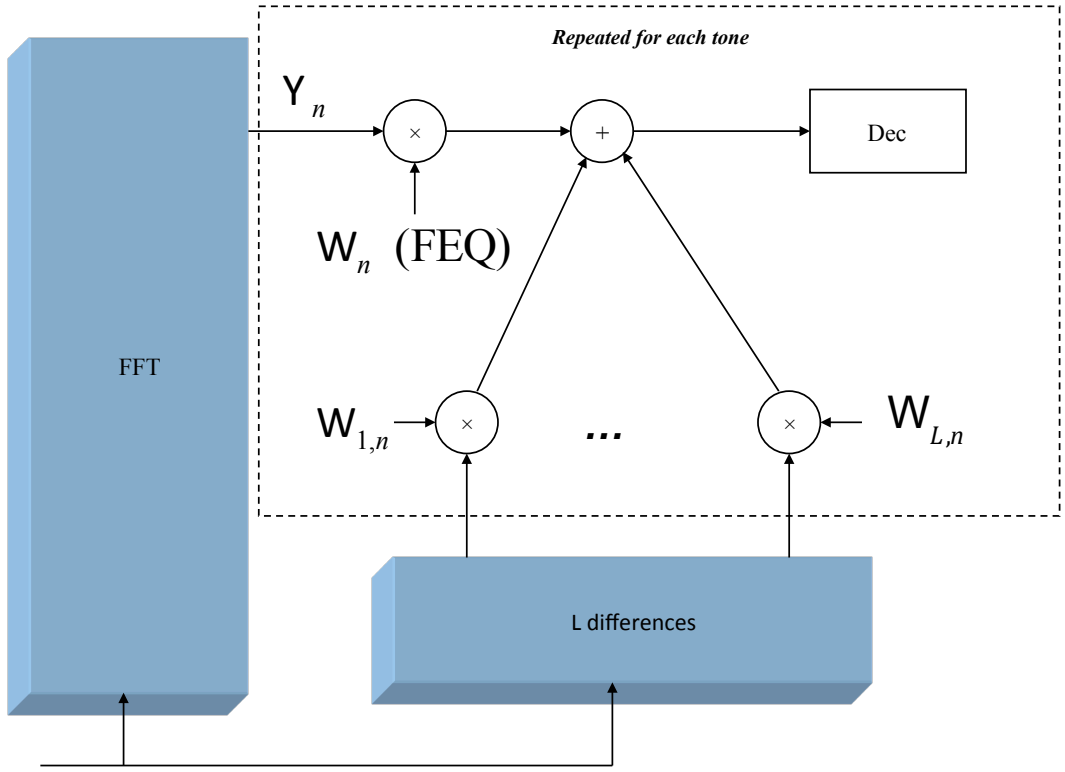


Figure 4.52: Per-tone equalizer.

4.8.5.2 MFEQ

Earlier than the per-tone equalizer, Bingham and Cioffi [3] note that the sliding-block FFT effect corresponds to a time-domain window (multiplication, and thus equivalent to a frequency-domain convolution). This concept is Figure 4.53's MFEQ; this MFEQ notes that the contributions from all the "per-tone" channel-output-sample differences to a specific tone n therefore concentrate largely in the adjacent tones. These tones could be on the last, current, and next symbols for that same tone and its adjacent tones. Figure 4.53 illustrates in a frequency-domain smaller "matrix" W. The complexity is typically much less than L operations/tone. TIn this case, the feedback (if and when used) is only a single previous decision on each tone or possibly a few previous decisions on the adjacent tones. The resultant MDFEQ will provide the highest performance of all these types of systems, only to be exceeded by the more exact Generalized DFE concept of Chapter 5 that directly addresses the overall (product) SNR for the symbol and maximization directly of overall data rate. The MFEQ can combine with 4.10's windowing to allow relatively short cyclic prefixes for reasonably small $\overline{N}$ and achieve nearly infinite-length performance.

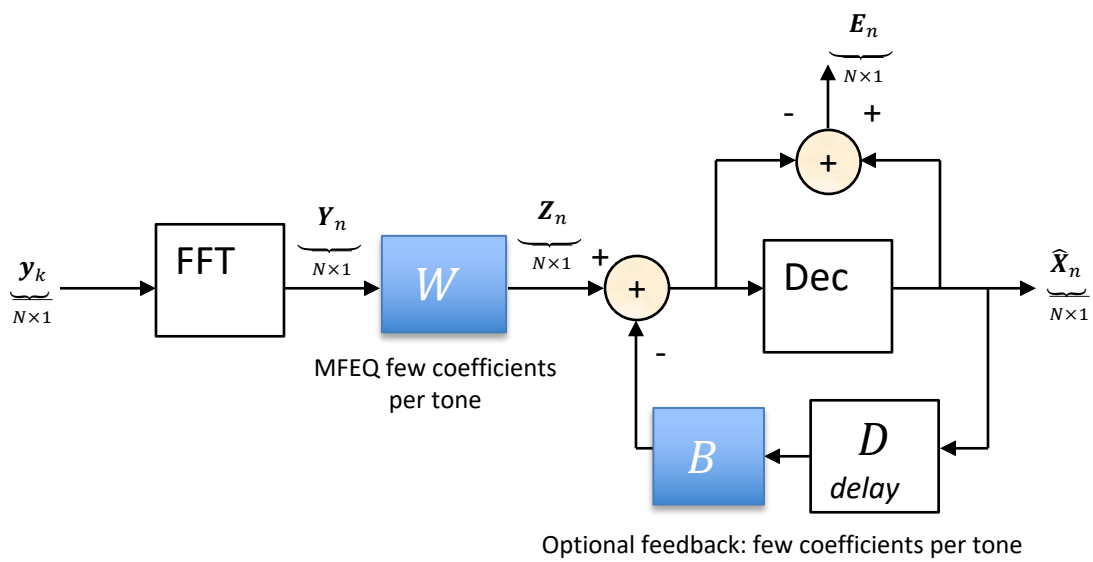


Figure 4.53: MFEQ

The MFEQ adaptive updates become somewhat simpler also in that that $\boldsymbol{Y}_{n,k}$ term for any tone update includes only the inputs that correspond to the smaller set of coefficients in $\boldsymbol{W}_{n,k}$, one for the current tone being updated, and a few more on each side of tone n in the update. The error $\boldsymbol{E}_n$ remains frequency domain. Similarly the feedback structure $\boldsymbol{B}$ that was present earlier can be banded also to a few adjacent tones for each frequency, and the $\hat{\boldsymbol{X}}_n$ is correspondingly limited to a few adjacent tones of the previously decided block's decisions.

$$\boldsymbol{W}_{n,k+1} = \boldsymbol{W}_{n,k} + \mu_W \cdot E_{n,k} \cdot \boldsymbol{Y}^*_{n,k} \tag{4.392}$$

$$\boldsymbol{b}_{n,k+1} = \boldsymbol{b}_{n,k} + \mu_b \cdot E_{n,k} \cdot \hat{\boldsymbol{X}}^*_{k-1} \tag{4.393}$$

The analysis of the MFEQ generally depends on the following channel decomposition.

$$\boldsymbol{y}_k = \widetilde{H} \cdot \boldsymbol{x}_k + H_1 \cdot \boldsymbol{x}_k + H_2 \cdot \boldsymbol{x}_{k-1} + \boldsymbol{n}_k \quad , \tag{4.394}$$

where

$$\widetilde{H} = \begin{bmatrix} h_0 & h_1 & \dots & h_L & 0 & \dots & 0 \\ 0 & h_0 & \ddots & h_{L-1} & h_L & \ddots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & h_0 & h_1 & \dots & h_L \\ h_L & 0 & \dots & 0 & h_0 & \dots & h_{L-1} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ h_1 & \dots & h_L & 0 & \dots & 0 & h_0 \end{bmatrix} \quad , \tag{4.395}$$

$$H_1 = \begin{bmatrix} \mathbf{0}_{(N-L+\nu)\times(N-L)} & \mathbf{0}_{(N-L+\nu)\times(L-\nu)} & \mathbf{0}_{(N-L+\nu)\times\nu} \\ \mathbf{0}_{(L-\nu)\times(N-L)} & -H_t & \mathbf{0}_{(L-\nu)\times\nu} \end{bmatrix} \quad , \tag{4.396}$$

$$H_2 = \begin{bmatrix} \mathbf{0}_{(N-L+\nu)\times(L-\nu)} & \mathbf{0}_{(N-L+\nu)\times(N-L+\nu)} \\ H_t & \mathbf{0}_{(L-\nu)\times(N-L+\nu)} \end{bmatrix} \quad , \tag{4.397}$$

and

$$H_t = \begin{bmatrix} h_L & 0 & ... & 0 \\ h_{L-1} & h_L & \ddots & 0 \\ \vdots & \ddots & \ddots & 0 \\ h_{\nu+1} & ... & h_{L-1} & h_L \end{bmatrix} \quad . \tag{4.398}$$

The DFT output is

$$Q \cdot \boldsymbol{y}_k = \Lambda \cdot \boldsymbol{X}_k + Q \cdot H_1 \cdot \boldsymbol{x}_k + Q \cdot H_2 \cdot \boldsymbol{x}_{k-1} + \boldsymbol{N} \tag{4.399}$$

or

$$\Lambda^{-1} \cdot Q \cdot \boldsymbol{y}_k = \left[I + \Lambda^{-1} \cdot Q \cdot H_1 \cdot Q^* \right] \cdot \boldsymbol{X}_k + \Lambda^{-1} \cdot \boldsymbol{N} \quad . \tag{4.400}$$

A reasonable block DFE design then sets

$$W = \left[I + \Lambda^{-1} \cdot Q \cdot H_1 \cdot Q^* \right]^{-1} \tag{4.401}$$

and

$$B = W \cdot \Lambda^{-1} \cdot Q \cdot H_2 \cdot Q^* \quad , \tag{4.402}$$

to find that the block equalizer forward section output is

$$W \cdot \Lambda^{-1} \cdot Q \cdot \boldsymbol{y}_k = \boldsymbol{X}_k + B \cdot \boldsymbol{X}_{k-1} + W \cdot \Lambda^{-1} \cdot \boldsymbol{N}_k \quad , \tag{4.403}$$

or after subtraction of $B \cdot \hat{\boldsymbol{X}}_{k-1}$ by the feedback section,

$$\boldsymbol{Z}_k = \boldsymbol{X}_k + W \cdot \Lambda^{-1} \cdot \boldsymbol{N}_k \quad , \tag{4.404}$$

which reproduces the channel input plus the filtered noise (clearly as $\nu \to$ Ł the noise remains uncorrelated from dimension to dimension). Thus, the MFEQ has approximately removed interference between the tones.

MIMO MFEQ: The MIMO MFEQ adds additional taps for other spatial dimensions, but otherwise follows the same MFEQ structure. The objective is also to cancel other dimensions residual crosstalk caused by the selected ν being insufficient. The MIMO receiver handles the same-tone spatial interference subsequent to the receiver FFT's; the MIMO MFEQ essentially handles too-short-guard-period crosstalk by adding dimensions for adjacent tones of other dimensions to the MFEQ structure above.

4.8.5.3 Cheong's Precoder

Former student Dr. Kok Wui Cheong [27] essentially produced the previous section's analysis and noted that Tomlinson Precoding effectively can be used on each transmitter dimension instead of the MFEQ's optional receiver feedback section. Figure 4.54's modulo operator $\Gamma_M(\cdot)$ is the same as in Chapter 3's Tomlinson Precoder, implemented independently for each tone's constellation. An equation corresponding to Figure 4.54 is

$$\begin{aligned} W^{-1} \cdot \boldsymbol{Z}_k &= \Gamma_M \left(\boldsymbol{X}_k - B \cdot \boldsymbol{X}'_{k-1} \right) && (4.405) \\ &= \Gamma_M \left(\boldsymbol{X}_k - B \cdot Q^* \cdot \boldsymbol{Z}_{k-1} \right) \quad . && (4.406) \end{aligned}$$

The receiver decision-device input is then, using Equation (4.394),

$$\begin{aligned} \hat{\boldsymbol{X}}_k &= \Gamma_M \left(\Lambda^{-1} Q \cdot \boldsymbol{y}_k \right) && (4.407) \\ &= \Gamma_M \left(\Lambda^{-1} \cdot Q \cdot \left[\widetilde{H} \cdot \boldsymbol{x}_k + H_1 \cdot \boldsymbol{x}'_k + H_2 \cdot \boldsymbol{x}'_{k-1} + \boldsymbol{n}_k \right] \right) && (4.408) \\ &= \Gamma_M \left(\Lambda^{-1} \cdot Q \cdot \widetilde{H} \cdot Q^* \cdot \boldsymbol{Z}_k + \Lambda^{-1} \cdot Q \cdot H_1 \cdot Q^* \cdot \boldsymbol{Z}_k + \Lambda^{-1} \cdot Q \cdot H_2 \cdot Q^* \cdot \boldsymbol{Z}_{k-1} + \tilde{\boldsymbol{N}}_k \right) \\ &= \Gamma_M \left(\boldsymbol{Z}_k + \left[W^{-1} - I \right] \cdot \boldsymbol{Z}_k + \boldsymbol{B} \cdot \boldsymbol{X}'_{k-1} \cdot \tilde{\boldsymbol{N}}_k \right) && (4.409) \end{aligned}$$

$$= \Gamma_M \left(W^{-1} \boldsymbol{z}_k + \boldsymbol{B} \cdot \boldsymbol{X}'_{k-1} \tilde{\boldsymbol{N}}_k \right) \tag{4.410}$$

$$= \Gamma_M \left(\boldsymbol{X}_k - \boldsymbol{B} \cdot \boldsymbol{X}'_{k-1} \cdot \boldsymbol{B} \cdot \boldsymbol{X}'_{k-1} \cdot \tilde{\boldsymbol{N}}_k \right) \tag{4.411}$$

$$= \boldsymbol{X}_k + \tilde{\boldsymbol{N}}_k \quad , \tag{4.412}$$

and the desired channel input appears (with noise) at the receiver's modulo output. The power increase of the filter W is often negligible on many channels of interest, as Cheong showed with many examples. Chapter 5's GDFE will refine this methodology and show that indeed that power increase (beyond that of the modulo element) can be avoided.

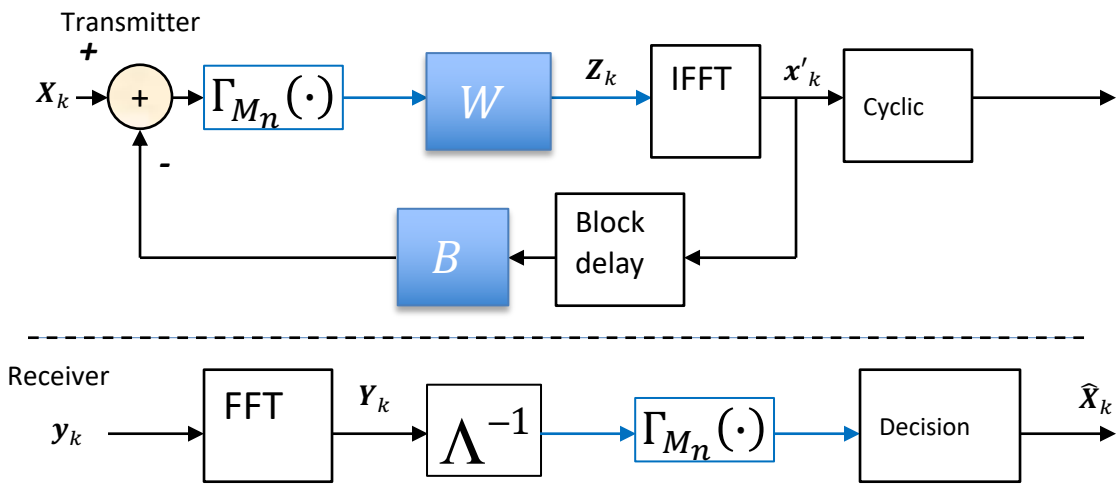


Figure 4.54: Cheong's Precoder

4.9 Windowing Methods for DMT

Cyclostationary[76] 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

$$x_n(t) = X_n \cdot e^{\jmath \frac{2\pi}{N} \cdot \frac{N+\nu}{T} \cdot nt} \cdot w_T(t) \quad , \tag{4.413}$$

where $w_T(t)$ is a window function. In Section 4.7's DMT/OFDM partitioning, the rectangular window function was

$$w_T(t) = \begin{cases} 1 & t \in [0, T) \\ 0 & t \notin [0, T) \end{cases} \quad . \tag{4.414}$$

Letting

$$\bar{f} \triangleq \left(1 + \frac{\nu}{N}\right) \cdot \frac{1}{T} = \frac{1}{NT'} \quad , \tag{4.415}$$

the product in (4.413) 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 (4.414) is

$$W_T(f) = \text{sinc}\left(\frac{f}{\bar{f}}\right) \cdot e^{-\jmath \pi f T} \quad , \tag{4.416}$$

which has the magnitude illustrated in Figure 4.55.

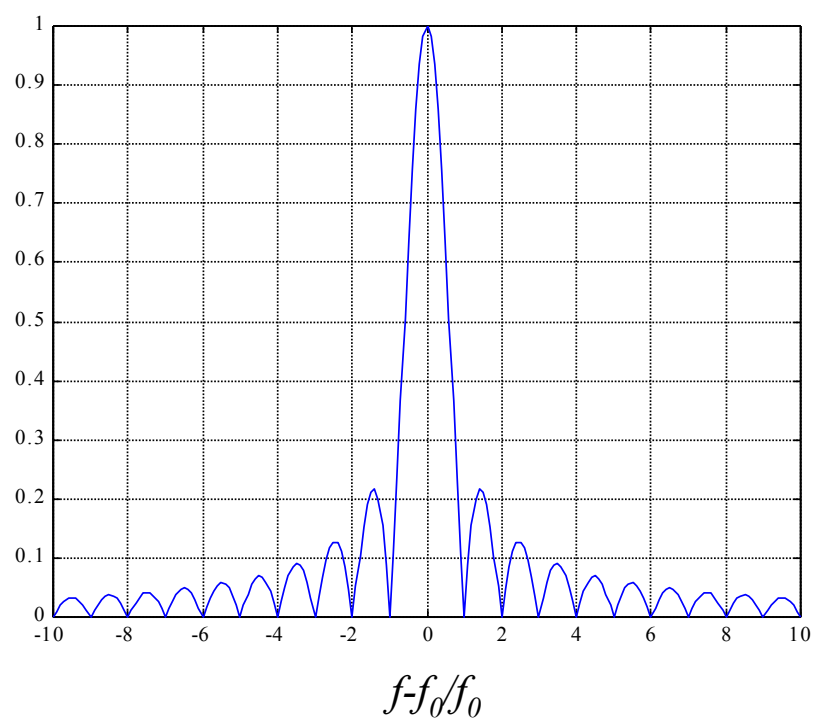


Figure 4.55: colorcardinalRectangular window fourier transform versus $\frac{f-\bar{f}}{\bar{f}}$.

Rectangular Window: While there is no interference between tones of DMT/OFDM because of the zeros in this function at the intermediate frequencies ($f \neq n \cdot \bar{f}$), there is significant non-zero energy between these frequencies. Thus the corresponding transmitted spectrum is not a "brick-wall" bandpass

[76] DMT/OFDM signals are stationary at symbol intervals, but statistically cyclic over a symbol period.

system of tones, but decays linearly (rather slowly) with frequency separation from any tone. Such energy may be of concern if there are emissions from the present transmitted channel into other transmission systems[77]. Even other DMT/OFDM transmissions systems that might sense crosstalk from the current DMT/OFDM system would need exact synchronization of sampling rate $1/T'$ to avert such crosstalk. A slight difference in 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 4.55.

Shaped Windows: Non-rectangular or shaped windows of the transmit DMT signal can re-introduce interference between tones, so window use requires care. Figure 4.56 illustrates two window methods that have no inter-tone interference through the use of extra cyclic prefix and suffix. Section 4.7's Zippered cyclic suffix 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) undergo the window shaping. A number of samples within the symbol remain un-windowed (or multiplied by 1, implying no change) so that DMT or OFDM partitioning exactly applies when the receiver FFT processes these samples. The remaining samples are shaped in a way that hopefully reduces the window function's out of band energy, through a more steep roll-off with frequency. The receiver ignores these samples that undergo window shaping. Steeper decrease with frequency occurs when the window function has as many continuous derivatives as possible. The windows for the current symbol's extra prefix and the window for the last symbol's extra suffix can overlap without change in performance if the sum of the two windows in the overlapping region equals 1 (to avoid any average energy increase). Such overlap reduces the window's increase of system excess bandwidth.

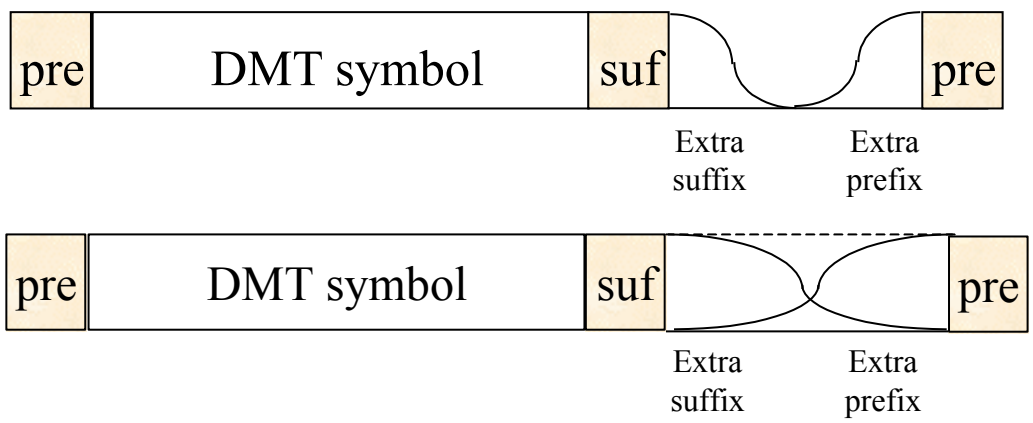


Figure 4.56: Illustration of extra prefix and suffix and non-overlapped and overlapped windows.

The most common choice of window in practice is a raised-cosine window of $2\beta + 1$ samples of extra cyclic extension (extra prefix plus extra suffix). This β is not the same as the unit of information used earlier in this chapter, and is local to this section to be consistent with other external literature terminology on windows. With $\nu = \Delta$ in Figure 4.57, this time-domain function is

$$w_{T,rcr}(t) = \begin{cases} 1 & |t| \leq \left(\frac{\overline{N}}{2} + \nu - \beta\right) T' \\ \frac{1}{2} \cdot \left\{1 + \cos\left(\pi \left[\frac{t - (\frac{\overline{N}}{2} + \nu - \beta)T'}{\beta T'}\right]\right)\right\} & \left(\frac{\overline{N}}{2} + \nu - \beta\right) T' < |t| < \left(\frac{\overline{N}}{2} + \nu\right) T' \\ 0 & |t| > \left(\frac{\overline{N}}{2} + \nu\right) T' \end{cases} . \quad (4.417)$$

[77]That may share the same channel or unfortunately sense radiation from it.

Figure 4.57 illustrates this window if the reader, momentarily, lets $\nu = \Delta$. The actual implementation occurs with samples $w_k = w_{T,rcr}(kT')$ for $k = -\frac{\overline{N}}{2} - \nu, ..., 0, ..., \frac{\overline{N}}{2} + \nu$. Letting

$$\bar{\alpha} = \frac{\frac{\beta}{2}}{\frac{\overline{N}}{2} + \nu - \beta} = \frac{\beta}{\overline{N} + 2 \cdot \nu - \beta} \quad , \tag{4.418}$$

and

$$\bar{f} = \left(1 + \frac{2\nu}{\overline{N}}\right) \frac{1}{T'} \quad , \tag{4.419}$$

the Fourier transform of this window becomes

$$W_{T,rcr}(f) = \frac{\beta}{\bar{\alpha}} \cdot \text{sinc}\left(\frac{f}{\bar{f}}\right) \cdot \frac{\cos \frac{\bar{\alpha} \cdot \pi \cdot f}{\bar{f}}}{1 - \left(\frac{2\bar{\alpha} f}{\bar{f}}\right)^2} \quad . \tag{4.420}$$

Figure 4.58 illustrates the sidelobes of the raised cosine function for $\bar{\alpha}$ values of 5% (blue) and 25% (red) along with those of $W_T(f)$ from Figure 4.55 (this time in decibels). Emissions into other systems can be significantly reduced for between 5% and 25% excess bandwidth. Section 4.7's FMT systems of offer even greater reduction if required.

EXAMPLE 4.9.1 *[Windowing in xDSL]* Figure 4.57 shows the raised-cosine window used by VDSL and G.fast. This method "robs" samples from the cyclic extension to use for windowing. If the cyclic prefix length is ν and the cyclic suffix length is Δ, the total number of used samples in the cyclic extension is $\nu + \Delta - \beta$ where β samples are allocated to the window shaping. In this case, the 2ν in the time-domain above is replaced by $\nu + \Delta$ or equivalently

$$\nu \rightarrow \frac{\nu + \Delta}{2} \quad , \tag{4.421}$$

and the phase of the window in Equation (4.420) is changed but not the amplitude and consequent swift frequency roll-off. The G.993 series of standards allows cyclic extention lengths of

$$\nu + \Delta - \beta = m \cdot \frac{m}{64} \quad , \; m = 2, ..., 16 \tag{4.422}$$

with a default m value of 5. The defaults are thus cyclic extensions of 320 for $N = 4096$ and 640 for $N = 8192$. Allowed β choices are

$$\beta = 0, ..., \; \min\left(\frac{\overline{N}}{8}, 255\right) \quad . \tag{4.423}$$

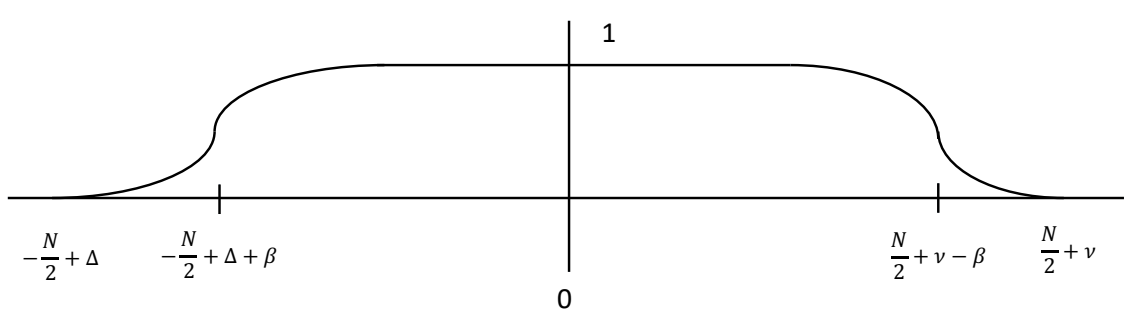


Figure 4.57: Raised Cosine Windowing for xDSL.

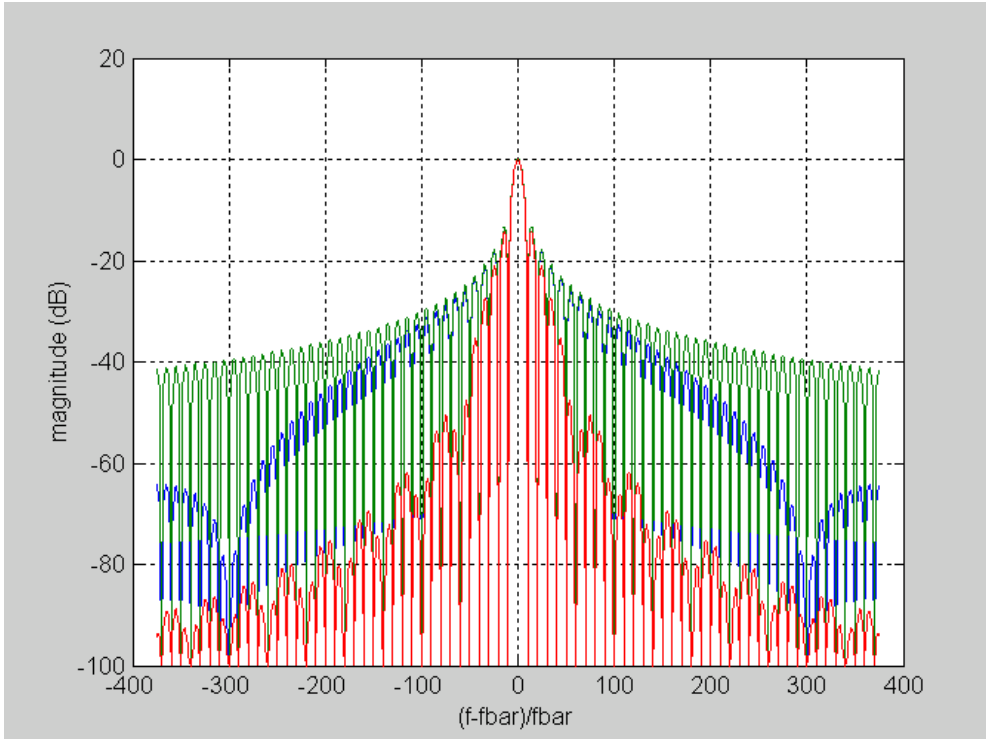


Figure 4.58: Illustration of transforms of rectangular window (green) and raised-cosine with 5% (blue) and 25% (red) excess bandwidth versus $(f - \bar{f})/\bar{f}$.

4.9.1 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 $\overline{N}$ may not be sufficiently large, and Section 4.8's TEQ may not adapt quickly enough to "notch" or reduce such noise. Also, the TEQ may not be sufficiently long to reduce such a noise's 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). Receiver windowing corresponds to frequency-domain convolution with the window function's Fourier transform. 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 4.58 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 simultaneous transmitter and receiver windowing).

4.10 Peak-to-Average Ratio reduction

The peak-to-average ratio **(PAR) of an $\overline{N}$-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 $\overline{N} = 1$ and $\overline{N} = 2$ 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:

$$x_{max} = \max_{k\ ,\ k=0,\ldots\overline{N}+\nu-1} |x_k| \quad . \tag{4.424}$$

This value determines the range of conversion devices (DACs and ADCs) used at the interfaces to the analog channel. The conversion device's number of bits, or precision, often determines its cost and also its power consumption. Thus, smaller precision is usually better (especially at high speeds). A higher ratio $x_{max}/\bar{\mathcal{E}}_{\boldsymbol{x}}$ usually leads to more conversion-device bits required. Also, subsequent (to DAC) or preceeding (to ADC) analog circuits (amplifiers, filters, or transformers) may exhibit increased nonlinearity as x_{max} increases. Finally, linear-amplifier power consumption also 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

$$x(max) = \max_{t\in[0,T)} |x(t)| \quad , \tag{4.425}$$

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. Gaussian x from a good code necessarily also will have very high or infinite peak values. However, even in the absence of such coding, the DMT/OFDM and VC PAR can still be an important design issue.

Subsection 4.10.1 begins with some fundamentals on peak-value probabilities and their relationship to data rate, showing that indeed very little data rate need be lost to eliminate large transmission signal peaks. 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 Stanford EE379 students and their colleagues!) where a few tones 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 coordination, 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 this section's approaches reduce the probability of nonlinear distortion to a level that can be accommodated in practice.[78]

A series of coded approaches have also been studied by various authors, which prohibit large constellation points 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 text will not consider them further.

4.10.1 PAR fundamentals

This section focuses 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.[79]

[78] 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.

[79] 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

$$x_k = \sum_{n=0}^{N-1} X_n \cdot e^{-j\frac{2\pi}{N}nk} \quad , \tag{4.426}$$

which is a sum of a large number of random variables. Losely speaking with the basic Central Limit Theorem or probability, such a sum in (4.426) 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(14.2 dB)) has a probability of 10^{-7} and a PAR of 12.5 dB has probability 10^{-5}. When such a peak occurs, with high-probability large "clip" noise occurs if the conversion devices or other analog circuits cannot handle this large value. So, with probabilities on the order of the bit-error probability, entire DMT/OFDM symbols may have many bit errors internally unless the circhit design's PAR 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).

High PAR in good designs: 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 3) 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 2). Thus, any well-designed transmission system can have the same high PAR problem. Fortunately, good solutions exist for the DMT/OFDM systems.

Clip Entropy: It might initially seem prospects for high performance in practice are thus dim unless the designer can afford the higher dynamic range and power consumption of conversion and analog circuits. However, let a designer assume a PAR of say 10 dB is acceptable (even sinusoids in practice require 3-4 dB) so this reduction to 10 dB saves roughly one bit of precision range[80] more than a non-information-bearing signal would require. Such a peak in a Gaussian distribution occurs with probability 10^{-3}. Internal to the data transmission stream, a small amount of overhead data rate can be 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 slow 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. In effect, clips have low entropy. This is exactly the observation used in the three following approaches of this section.

4.10.2 Tone Reservation

Tone reservation adds a **peak annihilator** $\boldsymbol{c}$ to each transmitted symbol $\boldsymbol{x}$. The peak annihilator was first introduced independently by former student Dr. Jose Tellado [28] in his Stanford dissertation and jointly by former EE379 student Alan Gatherer (then at Texas Instruments) and M. Polley of Texas Instruments[29]. Thus, the transmitter sends $\boldsymbol{x} + \boldsymbol{c}$ after cyclically extension. The peak-annihilator attempts to add the negative of any peak so that the signal reduces in amplitude at the the peak's sampling instant. The peak-annihilation vector $\boldsymbol{c}$ forms through frequency-domain inputs on a few tones in set $\mathcal{I} = \{i_1, i_2, ..., i_r, ..., \overline{N} - i_r, ..., \overline{N} - i_1\}$. Set $\mathcal{I}$ does not carry data. Typically $r << \overline{N}/2$. The determination of this set appears at this subsection's end.

[80] Conversion devices basically provide 6 dB of dynamic range increase for each additional bit used to quantize signals.

Thus,

$$\boldsymbol{x} + \boldsymbol{c} = \mathcal{Q}^* \cdot [\boldsymbol{X} + \boldsymbol{C}] \tag{4.427}$$

and

$$C_n = \begin{cases} 0 & n \notin \mathcal{I} \\ \text{selected} & n \in \mathcal{I} \end{cases} . \tag{4.428}$$

Thus, $X_n \neq 0$ when $n \notin \mathcal{I}$, leaving $\frac{\overline{N}}{2} - r$ tones available for data transmission and the others reserved for selection and optimization of the peak annihilator.

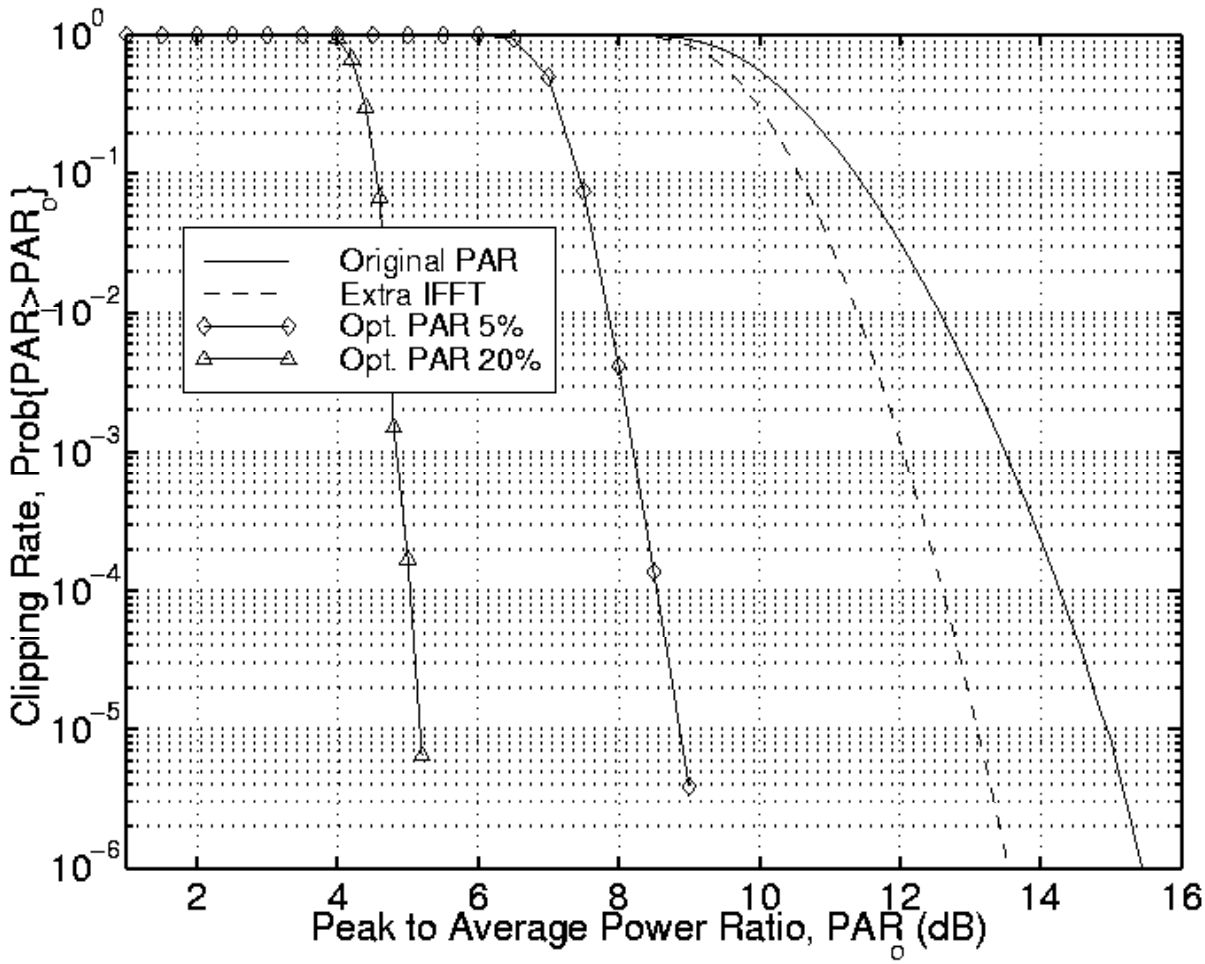


Figure 4.59: Probability of peak above specified PAR level for $N = 128$ OFDM/DMT systems with 5% and 20% of tones reserved. The "extra FFT" curve represents a double-pass of the transmitter IDFT where the phase of some signals has been selectively altered on the second pass to reduce PAR for comparison.

The peak annihilator forms as

$$\min_{\boldsymbol{C}} \max_{\boldsymbol{x}} |\boldsymbol{x} + \mathcal{Q}^* \cdot \boldsymbol{C}| \tag{4.429}$$

with the constraints that $C_n = 0, \ \forall \ n \in \mathcal{I}$ and $\|\boldsymbol{x} + \mathcal{Q}^* \boldsymbol{C}\|^2 \leq \overline{N} \cdot \bar{\mathcal{E}}_{\boldsymbol{x}}$. In this PAR context, the reader is reminded that $\tilde{N} = 1$ and inphase and quadrature are separately handled for peaks. Equation (4.429) is 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 4.59 illustrates the reduction in peak probability exceeding various PAR levels. For reservation of 5% of the tones, a 6 dB reduction (savings of one converter 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 appears next.

Gradient-descent approximation: The alternative method makes use of the function

$$clip_A(x_k) = \begin{cases} x_k & |x_k| < A \\ A \cdot \text{sgn}(x_k) & |x_k| \geq A \end{cases} \quad , \tag{4.430}$$

where $A > 0$, and $x_k \in \mathbb{R}$ and $A \in \mathbb{R}$. The "clip noise" for a symbol vector $\boldsymbol{x}$ is then

$$\delta^2_{\boldsymbol{x}} = \|\boldsymbol{x} - clip_A(\boldsymbol{x})\|^2 = \sum_{k=0}^{N-1} [x_k - clip_A(x_k)]^2 \quad . \tag{4.431}$$

With peak-annihilation in use

$$\delta^2_{\boldsymbol{x}+\boldsymbol{c}} = \|\boldsymbol{x} - clip_A(\boldsymbol{x} + \mathcal{Q}^* \cdot \boldsymbol{C})\|^2 \quad . \tag{4.432}$$

The gradient (with respect to $\boldsymbol{c}$) of the "clip-noise" is:

$$\nabla_{\boldsymbol{c}}\, \delta^2_{\boldsymbol{x}+\boldsymbol{c}} = 2 \cdot \sum_{k \ni |x_k + c_k| > A \& i \in \mathcal{I}} \text{sgn}(x_k + c_k) \cdot [|x_k + c_k| - A] \cdot \mathcal{Q}^*_{\mathcal{I}} \cdot \bar{\boldsymbol{q}}^*_k \tag{4.433}$$

where $\mathcal{Q}^*_{\mathcal{I}}$ is the $\overline{N} \times 2r$ 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 kth row of $\mathcal{Q}^*_{\mathcal{I}}$. The $\overline{N} \times 1$ vectors

$$\boldsymbol{p}_k = \mathcal{Q}^*_{\mathcal{I}} \bar{\boldsymbol{q}}_k \tag{4.434}$$

can be precomputed as they are a function only of the IFFT matrix and and the reserved tone indices.[81] These vectors are circular shifts of a single vector. They will have a "peak" in the kth 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

$$\epsilon_k = \text{sgn}(x_k + c_k) \cdot [|x_k + c_k| - A] \;\; \forall k \ni |x_k + c_k| > A \tag{4.435}$$

as a type of successive error signal, then the gradient algorithm becomes

$$\boldsymbol{c}_{j+1} = \boldsymbol{c}_j - \mu \cdot \sum_{k \ni |x_k + c_k| > A} \epsilon_k(j) \cdot \boldsymbol{p}_k \quad , \tag{4.436}$$

where j is an algorithm index reflecting how many iterations or gradients have been evaluated. Each of these iterations is a circular shift and add. This algorithm converges to close to the same result as the optimum linear-programming solution.

After approximately 10 iterations in Figure 4.60 (dashed line), 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

[81] 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 $\boldsymbol{c}$.

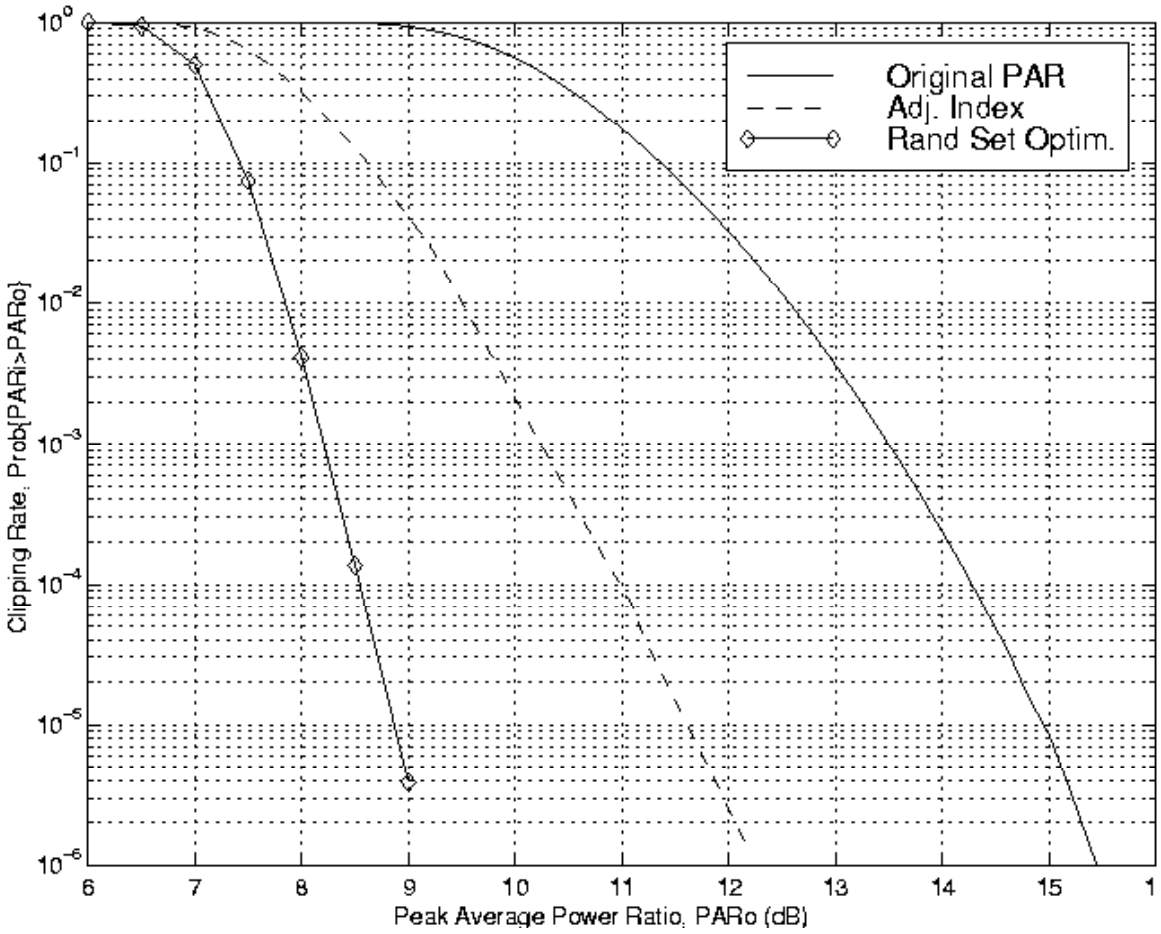


Figure 4.60: Probability of peak above specified PAR level for successive iterations of the gradient algorithm. The dashed line is 10 interations for $N = 128$ with 3 reserved tones.

calculations were for $N = 128$ and average over a variety of data vectors (and of course peaks only occur for a small fraction of those randomly chosen 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 of 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 [28] offers methods to optimize 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.)

Tone Reservation's requires no coordination between transmitter and receiver (the transmitter spoofs the receiver to off-load the tones in $\mathcal{I}$ by ensuring it measures poor channel gain on those tones. Thus, there is no need to specify the method in DMT. 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. (A code can recover the lost data.)

In continuous time, peaks can occur at time instants other than those corresponding to the DMT sampling times. 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. Nonetheless, this internal DSP cost could be negligible compared to the savings in conversion devices' dynamic range, power, and cost.

4.10.3 Tone Injection

A reserved-tone fraction that exceeds 5% considerably exceeds the minimum theoretical bandwidth loss that substantially reduces PAR. Tellado [28], and independently Kschichang, Narula, and Eyuboglu of Motorola [30], found means of adding peak-annihilation signals to the existing data-carrying tones.

Tone injection reserves no tones 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. Figure 4.61 illustrates the basic concept.

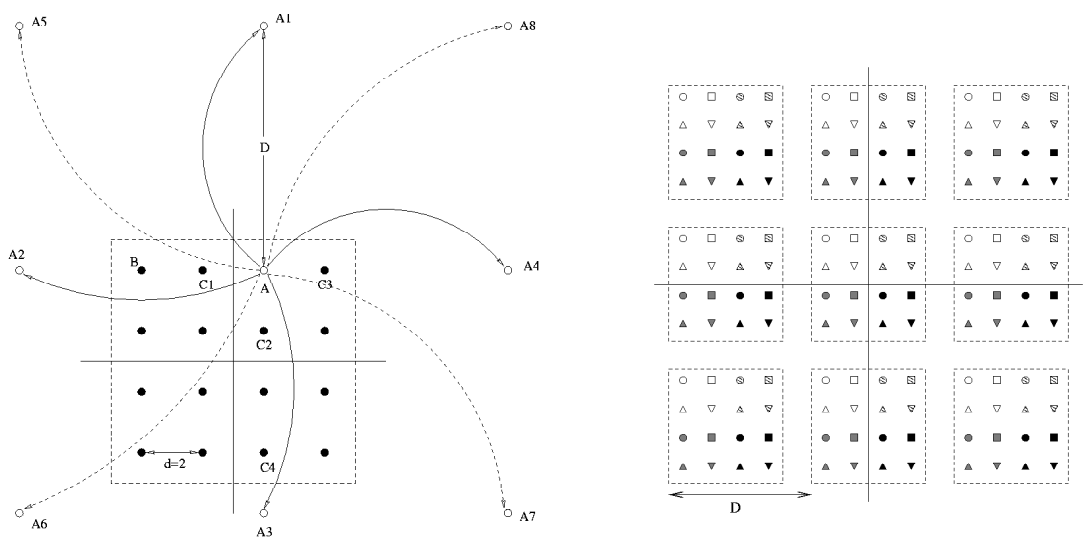


Figure 4.61: Illustration of constellation equivalent points for 16QAM and Tone Injection.

Tone injection allows replaced of a selected constellation point A by any of 8 equivalent points. Each of these 8 points has greater energy and increases the transmitted energy for that particular tone when sent. Tone injection searches over points like the 8 shown for several tones to find a resultant symbol with reduced peak(s). Because peaks do not occur often and because the extra energy only occurs on a few tones, tone injection's energy increase is a very small, typically .1 dB or less. A knowledgeable receiver design decodes the original point for any of the 8 possible replacement points as the same input message (or component of message). The offset in either horizontal or vertical direction is $\bar{d}$ units. Thus, peaks in frequency domain trade for peaks in the time domain. The frequencey-domain peaks have little transmit energy increase, and reduce significantly the analog driver circuit's consumed power.

More formally, the time-domain passband DMT or OFDM signal[82] is

$$x(t) = \frac{2}{\sqrt{N}} \cdot \sum_{n=1}^{N/2-1} R_n \cdot \cos(\frac{2\pi}{NT'} nt) - I_n \cdot \sin(\frac{2\pi}{NT'} nt) \tag{4.437}$$

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

$$\tilde{x}_k = x_k \pm \frac{2 \cdot \bar{d}}{\sqrt{N}} \cos(\frac{2\pi}{NT'} nkT'') \pm \frac{2 \cdot \bar{d}}{\sqrt{N}} \sin(\frac{2\pi}{NT'} nkT'') \;\; \forall n \in \mathcal{I} \quad . \tag{4.438}$$

It is often not necessary to add both the cos and sin terms, as one is often sufficient to reduce the peak dramatically. Peak-injection algorithms find a small set of n indices at any particular symbol for which an injected peak signal is opposite in sign and roughly 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,

$$|\tilde{x}_{k_0}| = |x_{k_0}| - 2\bar{d}/\sqrt{N} = |x_{k_0}| - \frac{2}{\sqrt{N}} \cdot \sqrt{\frac{6 \cdot 2^{2\bar{b}_{n_0}}}{2^{2\bar{b}_{n_0}} - 1}} \quad . \tag{4.439}$$

On each such algorithm iteration, a peak is reduced at location k_i, $i = 0, ...$ until no more reduction is possible or a target PAR level has been achieved. Figure 4.62 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.

[82] We avoid DC and Nyquist, which are simple extensions if desired and letting $R_n \triangleq \Re\{X_n\|$ and $I_n \triangleq \Im\{X_n\|$.

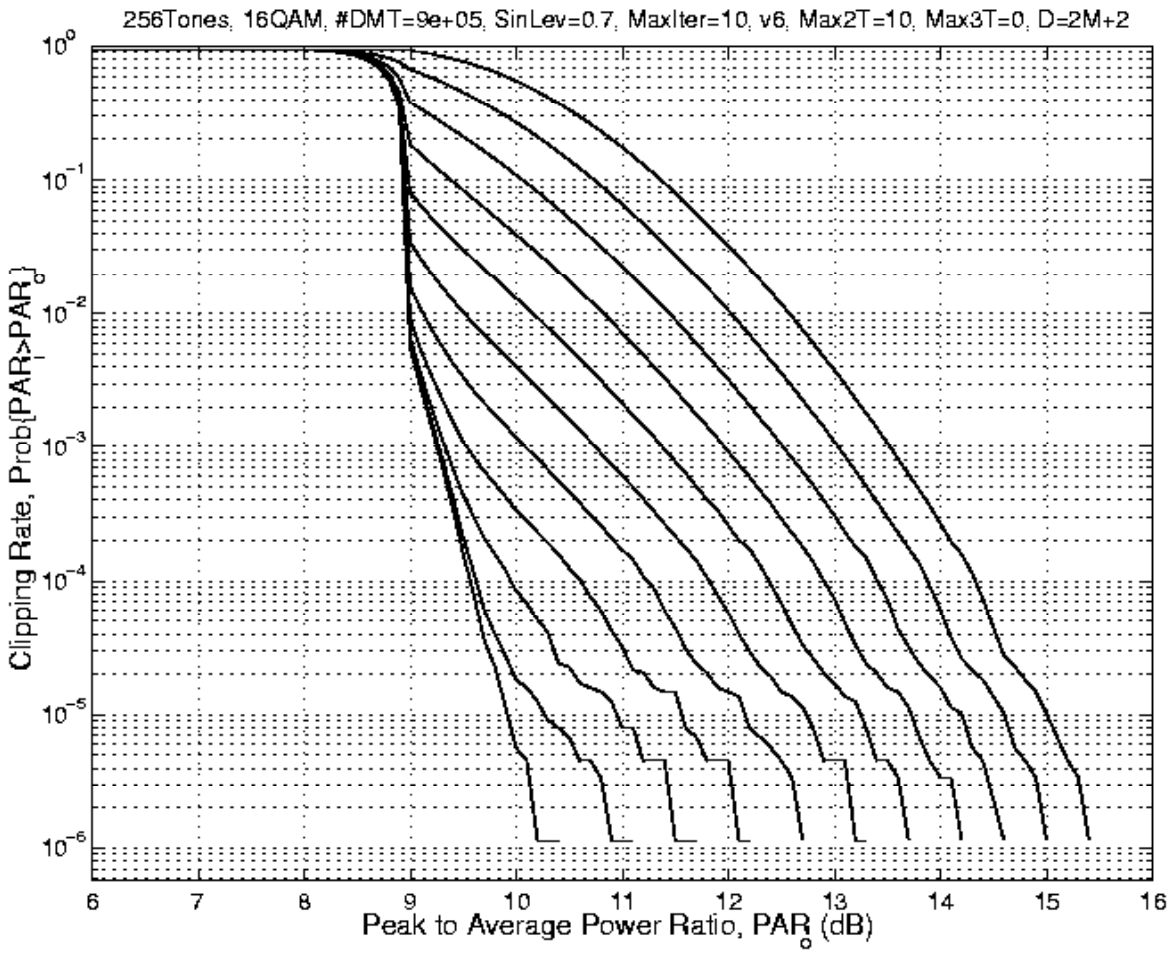


Figure 4.62: Illustration of successive iterations of tone injection.

4.10.4 Nonlinear Compensation of PAR

Tellado [28] also studied receivers that essentially store the knowledge of nonlinear mapping caused by peaks that exceed the channel's linear range in a look-up table. A maximum likelihood receiver then searches for most likely transmitted signal. This method is very complex, but has the advantage of nether bandwidth nor power loss, and exists only in receivers so need not be standardized per se, as would need to be the case for at least the silenced tones in tone reservation and the modulo-reception-rules for tone injection. Its complexity is highest. The receiver, also must "learn" the transmitters' nonlinearity, and this could require training periods that ultimately might need standardization.

Exercises - Chapter 4

4.1 *Short Questions (6 pts)*

a. (1 pts) Why is the X_0 subsymbol in Figure 4.2 chosen to be real?

b. (2 pts) Why is the $X_{\overline{N}}$ subsymbol in Figure 4.2 chosen to be one-dimensional? How would you interpret $\widetilde{N}$ on this channel?

c. (3 pts) Given ideal "sinc" transmit filters, $\varphi_n(t) = (1/\sqrt{T}) \cdot \text{sinc}(t/T)$, why are the channel output basis functions of multitone, $\varphi_{h,n}(t)$, orthogonal for all possible impulse responses $h_c(t)$?

4.2 *A Simple Multi-Tone Example (7 pts)*

a. (3 pts) Repeat Example 4.1 with the channel changed to $H(f) = 1 + 0.5 \cdot e^{j2\pi f}$ The symbol period is $T = N$ seconds. The matched-filter-bound has SNR $\text{SNR}_{\text{MFB}} = \frac{\bar{\mathcal{E}}_{\boldsymbol{x}} \cdot \|h\|^2}{\sigma^2} = 10$ dB, and the design-target Q-function-argument is 9 dB. The energy is $\bar{\mathcal{E}}_x = 1$, $N = 2\bar{N} = 8$. As in Example 4.1, the last sub-channel is not used.

b. (3 pts) Repeat your design in Part a for $N = 2\bar{N} = 16$.

c. (1 pt) Explain what is happening as N grows.

4.3 *GAP Analysis (7 pts)*
The reader may want to review Section 1.3 before proceeding with this question.

a. (1 pt) While the SNR gap may be fixed for often-used constellations, this gap is a function of two other design choices. What are these two choices?

b. (2 pts) Based on your answer to Part a, how can the gap be reduced?

c. (2 pts) For an AWGN channel with $SNR = 25$ dB 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 (5 pts)*

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 $1 + .9 \cdot D^{-1}$? Explain why.

c. (3 pts) Compare the difference of the SNR_{MFB} to the exact $SNR_{m,u}$ for the $1 + .5 \cdot D^{-1}$ and $1 + .9 \cdot D^{-1}$. Which one is closer? Why?

4.5 *Water-Filling Optimization (6 pts)*
Repeat Problem 4.2 using water-filling optimization. The given channel is $H(f) = 1 + .5 \cdot e^{j2\pi f}$, with $SNR_{MFB} = 10$ dB, and $\Gamma = 1$ (0 dB). (It is possible that all the dimensions 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, $\Gamma = 1$(0 dB). You may use the program waterfill.m at the course web site if so desired.

b. (2 pts) Calculate the gap for PAM/QAM that produces an argument of the Q-function equal to 9 dB. (The gap for $\bar{b} \geq 1$ is the difference between the SNR derived from capacity and the SNR in the argument of the Q-function for a particular probability of error.) What is that probability of error (that is the value of the Q-function at 9 dB)?

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 (6 pts)*

For the channel with $H(f) = 1 + 0.5 \cdot e^{j2\pi f}$, as in Problem 4.2 and 4.5, this problem will investigate the maximization of the margin using water-filling. (All the subchannels may be used in this problem). The block size is $N = 8$ for this problem. The waterfill program of Problem 4.6 may be used from class web site.

a. (2 pts) Is transmission of uncoded QAM/PAM at $P_e = 10^{-7}$ at data rate 1 possible? State the (uncoded) gap for $P_e = 10^{-7}$. Find the margin and see if it is ≥ 0, or the result of Problem 4.5(c) may be helpful.

b. (1 pt) For $\bar{b} = 1$, what gap provides zero margin? (A gap less than the gap in Part a means some level of coding is used.) If the design returns to uncoded PAM/QAM and does not use this code, the probability of error will be less than 10^{-7}. What $\bar{P}_e$ would correspond to uncoded PAM/QAM with the SNR that your first answer in this part?

c. (2 pts) What is the margin if $\Gamma = 1$(0 dB) is used instead of the code in Part b?

d. (1 pt) What is the margin if a code is used with the $\Gamma =$ margin in Part c (dB)?

4.7 *Using the the DMT Water-filling Programs (12 pts)*

Please use the DMTra and DMTma programs for this problem. Plots with only even values of N are acceptable. The sampling period is $T' = 1$.

a. (4 pts) Use a DMT-RA water-filling program on an FIR channel with real baseband pulse response given as a row vector ($h = [1\ 1 - 1 - 1]$ for $1 + D - D^2 - D^3$ or $1 + D^{-1} - D^{-2} - D^{-3}$ channel). Set the noise so that the matched-filter-bound on this channel is 17 dB. ν will be determined as the length of your FIR channel-input vector. The input parameters should be $\frac{\mathcal{N}_0}{2}$, $\bar{\mathcal{E}}_x = 1$, the number of dimensions (use 8 and 16), and the gap, $\Gamma = 0$ dB. (The programs implement the flow charts in Figures 4.8 - 4.10). Repeat this exercise for the channel $1 + D + D^2 + D^3$ and compare. Which channel seems better? Why?

b. (2 pts) Plot SNR_{dmt} vs. N for the channel $1 + D - D^2 - D^3$. ($\Gamma = 1$ (0 dB), $\bar{E}_x = 1$ and $\frac{\mathcal{N}_0}{2} = .1$)

c. (4 pts) Now use the DMTma program on the same two channels as Part a with $\bar{b} = 1$ What is the margin and the energy distribution for MA loading?

d. (2 pts) Plot margin vs. N for the channel $1 + D - D^2 - D^3$ with $\bar{b} = 1$ and $\Gamma = 0$ dB.

4.8 *Levin-Campello Algorithms (7 pts)*

This problem executes the Levin-Campello loading algorithm for the real baseband $H(D) = 1 + 0.5 \cdot D^{-1}$ channel with PAM/QAM, $\overline{Pe} = 10^{-6}$, $SNR_{MFB} = 10$ dB, $N = 8$, and $\bar{\mathcal{E}}_x = 1$. The loading granularity is $\beta = 1$ bit. Answers should show reasonable support.

a. (2 pts) Create a table of incremental energies $e(n)$ for loading with this channel. (This table need only include entries 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 $\bar{b}$ that can be supprted.

d. (1 pt) If the design were to reduce the total bits b by 2 bits from Part c, use the EF and BT algorithms to maximize the margin. What is this maximum margin?

4.9 *Using the DMT Levin-Campello Program (10 pts)*

Please use the DMTLCra and DMTLCma programs for this problem. Plots with only even values of N are acceptable. The sampling period is $T' = 1$.

a. (4 pts) Use a Levin Campello DMT-RA water-filling program on an FIR channel with the real baseband ($\widetilde{N} = 1$) pulse response given as a row vector ($h = [1\ 1 - 1 - 1]$ for $1 + D - D^2 - D^3$ or $1 + D^{-1} - D^{-2} - D^{-3}$ channel). Set the noise so that the matched-filter-bound on this channel is 17 dB. ν will be determined as the length of your FIR channel-input vector. The input parameters should be $\frac{\mathcal{N}_0}{2}$, $\bar{\mathcal{E}}_x = 1$, the number of dimensions (use 8 and 16), and the gap, $\Gamma = 0$ dB. Repeat this exercise for the channel $1 + D + D^2 + D^3$ and compare. Which channel seems better? Why?

b. (2 pts) Plot $\bar{b}$ vs. N for the channel $1 + D + D^2 + D^3$. ($\Gamma = 1$ (0 dB), $\bar{E}_x = 1$ and $\frac{\mathcal{N}_0}{2} = .1$)

c. (2 pts) Now use the DMTLCma program on the same two channels with the noise of Part b with $\bar{b} = 1$ What is the margin and the energy distribution for discrete MA loading? What happens as N increases?

d. (2 pts) Plot margin vs. N for the channel $1 + D - D^2 - D^3$ with $\bar{b} = 1$ and the noise of Part b and $\Gamma = 0$ dB. What does the maximum margin appear to be?

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 nth 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*

beyond this text's topics -= for the interested student: This problem uses 100 parallel channels(N=100) and evaluates different algorithm complexities:
Computation of a water-filling energy/bit distribution solves the water-filling equation 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 (6 pts)*

An ISI-channel with AWGN has a multitone transmission system that uses MA LC loading changes from real baseband $H(D) = 1+D-D^2-D^3$ to $H'(D) = 1+D-.1D^2-.1D^3$. 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 $P_e = 10^{-6}$. Remember to calculate a new SNR from the MALC program for each change in the channel.

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.) (Hint: A solution can find the energy and bit distributions for $N = 8$ and $N = 12$ before and after the channel change. The solution then shows which bits swapped and the corresponding incremental energies.)

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. The channel energy gain swaps are usually incremental to a previous value, ($G_i = \mathcal{E}_{new}/\mathcal{E}_{old}$) but if a channel gets a bit that didn't have one before just show the added incremental energy.

4.13 *Coded-OFDM Loading (10 pts)*

For the code of Example 4.4.2, the symbol rate is 250 kHz. Use $\overline{P}_e$ for all error-probability targets in this exercise.

a. (3 pts) What are the possible data rates that might be achieved in C-OFDM Loading with any possible SNR?

b. (2 pts) If the channel loss reduces (so improves the $SNR_{geo} = 14.5$ dB in Example 4.4.2) by 6.3 dB, what would be the best possible data rate now at the slightly increased $\bar{P}_e = 10^{-7}$?

c. (1 pt) If a margin of 3 dB were to be imposed on the results of Part b, how would the answer change?

d. (2 pts) Suppose the channel (in addition to the 6.3 dB in Part b) improved by 5.5 dB and that a constellation 256-QAM were allowed, what would be the date rate?

e. (2 pts) Suppose BPSK were allowed in a lowest safe mode. Compute the ratio between the new lowest data rate and the highest in Part a. What are three types of channel variation whose absence or presense that could each individually cause higher or lower speeds respectively to be best?

4.14 *Ergodic Water-fillilng and C-OFDM (10 pts)*

A channel has uniform discrete channel-SNR probability distribution $p_g = 0.1$ over 10 intervals with g interval endpoints given by:

$$\begin{aligned} g0 &= 0 \\ g1 &= 0.0105 \\ g2 &= 0.0223 \\ g3 &= 0.0357 \end{aligned}$$

$$
\begin{aligned}
g4 &= 0.0511 \\
g5 &= 0.0693 \\
g6 &= 0.0916 \\
g7 &= 0.1204 \\
g8 &= 0.1609 \\
g9 &= 0.2303 \\
g10 &= 0.4 \\
Pr\{g > g10\} &\approx 0
\end{aligned}
$$

The quantity $\bar{\mathcal{E}}_{\boldsymbol{x}}/\sigma^2$ for this channel is 43 dB and multiplies the individual gain to obtain SNR. Only the constellations 4-QAM, 16-QAM, and 64-QAM are permitted.

a. (3 pts) Find the average channel SNR, $< g >$, if the discrete g values are the centers of each of the 10 intervals. If input energy and channel noise variance were normalized so that $\bar{\mathcal{E}}_{\boldsymbol{x}}/\sigma^2 = 1$, what is the corresponding scale factor by which to multiply each value of g? Which rate-adaptive design appears to have the best date rate? Why might this be misleading? What is the safe data rate/choice?

b. (5 pts) Find the Margin Adaptive Ergodic water-filling solution for this channel and corresponding margin when Γ is both 9 dB (assume code rate 1) and 3 dB (assume code rate 1/2). What are the two data rates in bits/dimension?

c. (2 pts) Repeat Part b for RA Ergodic water-filling and find the corresponding data rates if the margin is non-negative. If you could increase the constellation size further, to what would you increase it?

4.15 *802.11 Loading (10 pts)*

A C-OFDM system with transmit power 15 dBm allows square constellations 4^n-QAM where $n = 1, 2, 3, 4$ and uses a punctured rate-1/2 code with rates and free distances respectively given by the ordered pairs $[\frac{1}{2}, 10], [\frac{2}{3}, 6], [\frac{3}{4}, 5], [\frac{5}{6}, 4], [\frac{7}{8}, 3]$. The average channel attenuation is 90 dB. Both the 20 MHz and 40 MHz bandwidth options of IEEE 802.11 are permitted on this single-antenna system. The receiver's front-end analog signal processing increases the thermal noise by a factor of 6 dB relative to the channel thermal noise, which is known as a **noise figure**, of 6 dB. Thermal noise[83] at room temperature has flat one-sided power spectral density of $\mathcal{N}_0 = -174$ dBm/Hz.

The channel-gain ($|h|^2/\sigma^2$) distribution $p_i \triangleq Pr\{g_i \le g < g_{i+1}\}$ for the 20MHz-wide-channel option is

$$
p_i = \begin{cases}
.25 & i = 0 & 0 \le g < 3\tilde{g} \\
.35 & i = 1 & 3\tilde{g} \le g < 6\tilde{g} \\
.20 & i = 2 & 6\tilde{g} \le g < 12\tilde{g} \\
.15 & i = 3 & 12\tilde{g} \le g < 24\tilde{g} \\
.05 & i = 4 & 24\tilde{g} \le g < \infty
\end{cases}
$$

There is no finer resolution on the lower bin as the receiver simply indicates which bin the current g value is in, and will always indicate erasure for index $i = 0$.

a. (2 pts) Find the noise power (after the front-end analog signal processing) in dBm into the remainder of the receiver for both of the bandwidth options. Also, find the corresponding power spectral density levels (one-sided, $\mathcal{N}_0$) for each of these bandwidth options.

b. (2 pts) Find the SNR at the same point as Part a for both bandwidth options.

c. (1 pt) What is the ratio of the data rates if the same code (same r and same $|C|$) is compared on the two bandwidth options (hint: remember to use the number of data-carrying tones only).

[83] kT_a noise where $k = 1.38064852 \times 10^{23}\ \frac{m^2 kg}{s^2 K}$ is the Boltzmann constant and T_a is the absolute temperature in degrees K.

d. (2 pts) Find $\tilde{g}$ assuming that the value representing each interval is the lowest value.

e. (2 pts) Find the best r and $|C|$ for RA loading.

f. (1 pt) Find the data rates corresponding to your solutions in Part e.

4.16 *Binning (15 pts)*

A C-OFDM system allows square constellations 4^n-QAM where $n = 1, 2, 3, 4$ and uses a punctured rate-1/2 code with rates and free distances respectively given by the ordered pairs $[\frac{1}{2}, 10], [\frac{2}{3}, 6], [\frac{3}{4}, 5], [\frac{5}{6}, 4], [\frac{7}{8}, 3]$. The transmit energy per dimension is normalized to $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$, and the system random-error target rate is $\bar{P}_e = 2 \times 10^{-5}$ where the number of nearest neighbors is assumed to be 2 per dimension. The system has 48 used tones and a symbol rate of 250 kHz.

The designer simply wants to use the g (channel SNR) values that would cause a change in constellation as the indices or "bins" for selection of the code rate r and constellation size $|C|$, and then "bin" to these values, as examined in this problem.

a. (2 pts) Write a formula for the $\bar{P}_e$ expression using the Q-function with an argument that reflects the channel gain g, free distance d_{free}, and the constellation size $|C|$. Solve this expression using the inverse Q function to find a relationship that relates g to $|C|$ and d_{free}.

b. (2 pts) Create a table of the 20 necessary SNR's (i.e., the g values) that just barely attain or better the target performance for each of the code choices (g value is the lowest that achieves performance with values above it presumably using the same choice until the next highest rate is achieved just barely by the next highest g value).

c. (2 pts) Create a corresponding table of data rates for the channel-SNR g choices in Part b. Do some rate/constellation choices with larger constellations have lower data rates than other rate choices with smaller constellations? Comment on what is not be considered here, but is important with a random g.

d. (4 pts) If g is random and has continuous exponential probability distribution exponential $p(g) = \frac{1}{10} \cdot e^{-\frac{g}{10}}$ (so $\mathcal{E}_g = 10$), find the corresponding discrete binned probability distribution for your answer in Part b. The Matlab command "reshape" may prove useful in converting the tables of Part b to a linear array.

e. (1 pt) Find the average data rate corresponding to the distribution in Part d if any (non-integer) size constellation is allowed and a very good code is used.

f. (2 pts) Generate 48 random samples from the exponential distribution in Part d, by using a uniform random value generator on the interval $[0, 1]$ in Matlab and the result that a non-uniform cumulative distribution function $F(x)$ for a random variable x can be generated by applying the function $x = F^{-1}(u)$ to samples from a uniformly distributed random variable u. Compare the bins of these samples to your answer in Part b. The Matlab command "histcounts" may prove very useful. Plot the distribution and this one-symbol binned distribution.

g. (2 pts) Repeat Part f for 20 symbols of data and draw conclusions.

4.17 *VC for the* $1 + 0.5D^{-1}$ *(6 pts)*

For the real baseband $1 + .5 \cdot D^{-1}$ 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?

4.18 *DMT for the* $1 + 0.5 \cdot D^{-1}$ *(7 pts)*

For the real baseband $1 + .5 \cdot D^{-1}$ channel, $N = 8$, $\bar{\mathcal{E}}_x = 1$, $\frac{\mathcal{N}_0}{2} = .125$, $T' = 1$, and $\Gamma = 3$ dB.

a. (2 pts) Solve for the DMT channel gains g_n and the optimal water-filling RA bit distribution. What is the data rate?

b. (2 pts) Compare the DMT gains g_n to the gains λ_n^2 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) Investigate the singular values of H for this channel as N increases from 8, 16, and 32. How does this change with respect to DMT as N increases?

4.19 *Loading (15 pts)*

A DMT design is used on the channel with real baseband discrete-time response $H(D) = 1 - D^2$ 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: If you execute this without the assistance of Matlab, 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, $|H(e^{-\jmath\omega T})|$ 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. Comment on why this design has an SNR that seems similar to the $1 + .9D^{-1}$ example in the notes. (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 $\beta = 1$. (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)

4.20 *Generalized Nqyuist (6 pts)*

a. (2 pts) For what $N \geq 1$ does multi-tone transmission satisfy the Generalized Nyquist Criterion at symbol rate $1/T$ on any linear ISI additive white Gaussian noise channel? Is there an exception case?

b. (2 pts) What could the receiver do to attempt to attempt to satisfy the Generalized Nyquist Criterion for finite N with a multi-tone transmitter?

c. (2 pts) How does the N necessary for a channel depend on the channel characteristics?

4.21 *Periodic Channels (6 pts)*

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 Multi-Tone 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. Both the input and output signals are restricted to the interval $[0, T)$, although the channel is infinite length.

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. (2 pts) 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?

4.22 *Discrete Loading on MIMO (15 pts)*

There are three parallel QAM channels for a transmission path with symbol rates and channel gains given by the following table:

Channel Index n	g_n	$\frac{1}{T_n}$
1	20	10 MHz
2	30	10 MHz
3	40	20 MHz

The total transmit power that can be used is 16 energy units per 100 ns. The systems can only use QAM square constellations of sizes 4, 16, 64. The system probability of symbol error target is $\bar{P}_e = 10^{-6}$ and analysis should use a $\Gamma = 9$ dB. Note that one of the channels has twice the bandwidth of the other two.

a. (8 pts) Provide an incremental energy table to be used in loading. For an example of this see the table in Example 4.3.5.

b. (4 pts) Find the maximum data rate that can be achieved in bits/second.

c. (3 pts) If the bit rate desired is 100 Mbps, what is the maximum margin?

4.23 *Vector Coding Theory (8 pts)*

This problem assumes real baseband ($\widetilde{N} = 1$) channel.

a. (1 pt) Show that vector coding's post-F noise vector $\boldsymbol{N}$ is indeed white (independent in all dimensions) 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. (2 pts) Prove that $F^* = \begin{bmatrix} 1+\frac{1}{SNR_1} & & & 0 & 0 \\ & 1+\frac{1}{SNR_2} & & & 0 \\ & & . & & . \\ 0 & & & . & \\ 0 & 0 & . & . & 1+\frac{1}{SNR_N} \end{bmatrix} \Lambda R_{\boldsymbol{Xy}} R_{\boldsymbol{yy}}^{-1}$, which is the unbiased MMSE Linear Equalizer for MIMO systems from Chapter 3.

4.24 *Grasping Vector Coding → DMT (8 pts)*

Assume the real-baseband channel is $1 + a \cdot D^{-1}$. 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. (3 pts) 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?

4.25 *Partitioning (15 pts)*

Figure 4.63 shows a wireless transmission system that uses uncoded QAM with gap $\Gamma = 8.8$ dB at $P_e = 10^{-6}$ on each of 3 transmit antennas with the same exact carrier but independent input data streams. Each has the same synchronized symbol rate $1/T = 100$ kHz, and the energy per dimension at that rate is unity ($\bar{\mathcal{E}}_x = 1$) for each antenna. There are also 3 receive antennas.

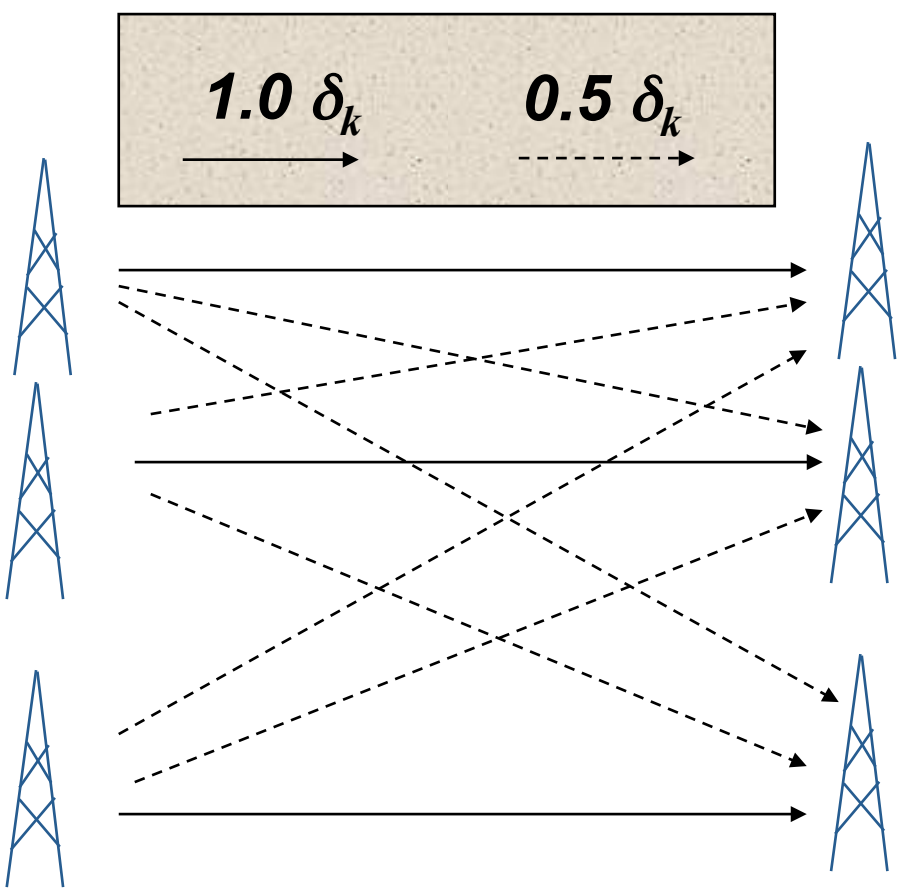


Figure 4.63: Wireless MIMO channel.

All 9 paths shown are AWGN channels with independent noise of two-sided power spectral density $\frac{\mathcal{N}_0}{2} = .01$ on each. Figure 4.63's legend provides the channel H values for each of these 9 paths. The "other" signals may be considered as noise for each of the 3 main signal paths from transmit antenna n to corresponding receive antenna directly to its right, or all 3 signals may be jointly transmitted and received by a single receiver that processes all 3 antennas' outputs. This problem investigates the difference in approaches.

a. (2 pts) If a receiver treats the "other" two signals as 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 power spectral density $\frac{\mathcal{N}_0}{2} = .01$.) (2 pts)

b. (3 pts) Find the 3×3 MIMO matrix channel H. Does it have any recognizable patterns?

c. (1 pt) How many real dimensions are there in this MIMO channel? How many complex dimensions?

d. (5 pts) Use Part b's answers 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/complex subsymbol for each of Part d's channels?

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?

4.26 *MIMO Partitioning Theory for* 2×2 *(15 pts)*

A discrete-time wireless channel has two input antennas and two output antennas with a transform matrix.

$$\begin{bmatrix} y_{2,k} \\ y_{1,k} \end{bmatrix} = \begin{bmatrix} 1 & a \\ a & 1 \end{bmatrix} \begin{bmatrix} x_{2,k} \\ x_{1,k} \end{bmatrix} + \begin{bmatrix} n_{2,k} \\ n_{1,k} \end{bmatrix} ,$$

where the stationary AWGN noise samples, $n_{1,k}$ and $n_{2,k}$, are independent with variance $E[|n_{1,k}|^2] = E[|n_{2,k}|^2] = \sigma^2$. There is no intersymbol interference, but there is crosstalk measured by the parameter a where $|a| \leq 1$. The synchronized channel inputs are QAM signals each with the same symbol rate $1/T$, and the parameter a is real. The sum of the transmitted energies is $\mathcal{E}_{\boldsymbol{x}} = \mathcal{E}_{x,1} + \mathcal{E}_{x,2}$. For this problem, the gap is 1 (0 dB).

a. (4 pts) Find a partitioning of this channel into two parallel independent channels that does not depend on a. Show the transmit matrix, receive matrix, and the channel gains.

b. (2 pts) What is the best assignment of energy for the antenna inputs if the two channels are subject only to a sum-of-energy constraint? Does this best energy depend on a?

c. (2 pts) What is the minimum SNR $= \frac{\mathcal{E}_{\boldsymbol{x}}}{2\sigma^2}$ as a function of a to ensure that there are two subchannels being used?

d. (3 pts) What is the maximum bit rate for a given a if the SNR is presumed above the minimum found in Part c? What is the maximum bit rate if the SNR is presumed below the minimum found in Part c?

e. (3 pts) Does this symmetric crosstalk between the antennas hurt or help the data rate? Provide an example where total energy is 10 units and $\sigma^2 = 1$, then try a few values of a between 0 and 1.

f. (1 pt) If Vector DMT was used, what would be the length of the smallest cyclic prefix needed?

4.27 *Hijacking the other industry's channel: DOCSIS 3.1 versus G.fast (G.9701), and wireless (11 pts)*

Many major internet service providers use coaxial distribution of internet signals on separate coaxial cable connections from a building basement to each residential living unit within the building. These connections are single-user point-to-point and thus do not need to multiplex (directly) multiple users.

On such connections, it becomes interesting to compare G.fast (originally intended for twisted pairs, which does not need all its "vector DMT" power for crosstalk cancellation) and DOCSIS 3.1 (which does not need all its profiling and multi-user multiplexing) for this single connection application.

a. (4 pts) Compare the maximum bandwidth used at baseband by G.fast with $N = 8192$ with a DOCSIS 3.1 system that uses $N = 4096$ both upstream and downstream:

   (i) (1 pt) What are the rough used bandwidths for each?

   (ii) (1 pt) Compare the Δf carrier/tone widths?

   (iii) (1 pt) Cyclic-extentions?

   (iv) (1 pt) Why is this a reasonable comparison to make?

b. (2 pts) Presuming DOCSIS 3.1's full use of the (maximum number of) lowest channels for upstream and G.fast's nominal 10% use of upstream, which would you expect to perform better? (Ignore loading-algorithm differences of DOCSIS and G.fast for this part).

c. (2 pts) Repeat Part b with the same assumptions for downstream. Which direction (down or up) would you expect to be more important in terms of speed and performance in terms of users' typical needs?

d. (1 pt) If loading is taken into account, how does this affect any comparisons made so far?

e. (1 pt) Suppose a latency of 1ms were desired, which system implementation is more likely to achieve this?

f. (1 pt) How would the use of a wireless transmission system like Wi-Fi or LTE perform on this channel if used at maximum symbol rate and bandwidth in comparison to either of DOCSIS 3.1 or G.fast?

4.28 *Simple Adaptive Computation of SVD in Vector Coding. (13 pts)*

The results of Problem 4.23 may be useful in this problem. Vector coding is used on the AWGN channel in Figure 4.64.

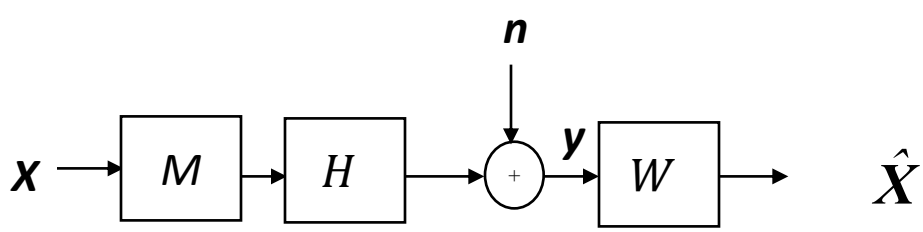


Figure 4.64: Illustration of channel for SVD computation.

The receiver uses $W = R_{\boldsymbol{X}\boldsymbol{y}} \cdot R_{\boldsymbol{y}\boldsymbol{y}}^{-1}$, a MMSE estimate for $\boldsymbol{X}$ given $\boldsymbol{y}$. Then

$$\boldsymbol{X} = W \cdot \boldsymbol{y} - \boldsymbol{U} \qquad \mathbb{E}\{\boldsymbol{U} \cdot \boldsymbol{U}^*\} = R_{\boldsymbol{U}\boldsymbol{U}}$$

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 subchannel? Why is this the same as minimizing MSE on each subchannel individually? (It can be assumed that the input $\boldsymbol{X}$ has non-zero energy on all dimensions for this computation.)

b. (1 pt) Show how the VC receiver W could be implemented adaptively, without explicit knowledge of H or F^* (pictorially and in few words). The implementation may still depend on M at this point.

Now, the hard part, implementing the transmitter without explicit knowledge of H or M.

c. (2 pts) Pretend $\boldsymbol{x}$ is a channel output and $\boldsymbol{Y} \triangleq W \cdot \boldsymbol{y}$ is a channel input for the MMSE value of W found previously in this problem. 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 $\boldsymbol{x}$? (While it may not be true on most channels, for this problem the matrix $R_{\boldsymbol{Y}\boldsymbol{Y}}$ should be considered non singular. Chapter 7 will deal with singularity and exploit the results of this homework problem.)

e. (2 pts) If the receiver could, during training, communicate $\boldsymbol{Y}$ or $\boldsymbol{y}$ (perhaps infrequently) 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)?

4.29 *Bias and Error Probability - 7 pts*

This problem illustrates that the best unbiased receiver has a lower P_e than the (biased) MMSE receiver using Figure 4.65's 3-point PAM constellation.

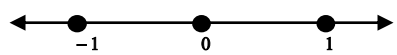


Figure 4.65: A three point PAM constellation.

The AWGN n_k has

$$\frac{\mathcal{N}_0}{2} = \sigma^2 = 0.1.$$

The inputs are independent and identically distributed uniformly over the three possible values. Also, n_k has zero mean and is independent of x_k.

a. (1 pt) Find the mean square error between $y_k = x_k + n_k$ and x_k. (hint: this is easy).

b. (1 pt) Find the exact P_e for the ML detector of the unbiased values y_k.

c. (2 pts) Find the scale factor α that will minimize the mean square error

$$\mathbb{E}[e_k^2] = \mathbb{E}[(x_k - \alpha \cdot y_k)^2].$$

Prove that this α does in fact provide a minimum by taking the appropriate second derivative.

d. (2 pts) Find $\mathbb{E}[e_k^2]$ and P_e for the scaled output $\alpha \cdot y_k$. When computing P_e, use the decision regions for the unbiased ML detector of Part b. Show that the biased receiver of Part c has a P_e that is higher than the (unbiased) ML detector of Part b, even though it has a smaller squared error.

e. (1 pt) Where are the optimal decision boundaries for detecting x_k from $\alpha \cdot y_k$ (for the MMSE α) in Part c? What is the error probability for these decision boundaries?

4.30 Problem 1 - Triple: multidimensional water-filling (13 pts):

Figure 4.66 illlustrates a complex baseband vector DMT system with $\mathcal{E}_x = 4$ energy units per symbol, and $\nu = 1$. All transmit messages are equally likely. Figure 4.66 also shows the additive white Gaussian noise spatial-autocorrelation matrices.

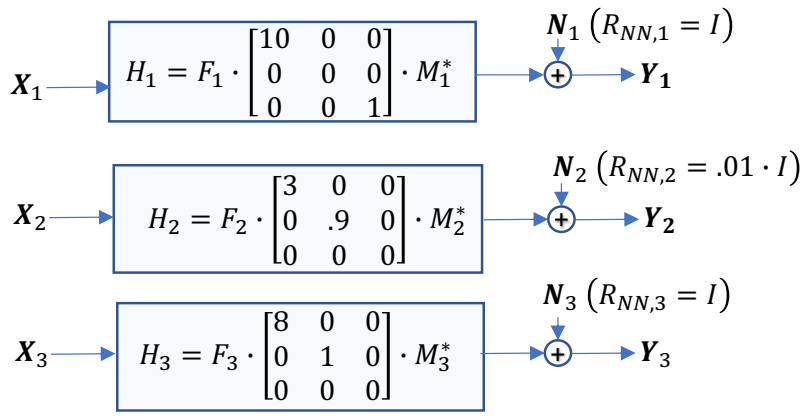


Figure 4.66: **Vector DMT Channel - Problem 1, 2023 Midterm**

a. (Total part a = 3 pts) From Figure 4.66, find

- the number of transmit antennas L_x =? (1/2 pt) ,
- the number of receiver antennas L_y =? (1/2 pt),
- and number of tones $\overline{N}$ =? (1/2 pt)
- What is the maximum number energy-carrying dimensions for **any** channel that uses your answers for L_x, L_y, and $\overline{N}$? (1/2 pt)
- Finally, for this specific channel how many dimensions can support energy transfer? (1 pt)

b. Describe the best discrete modulator and demodulator matrices. (1 pt)

c. Find and order the channel gains, g_n =?, largest to smallest. (2 pts)

d. Find the highest reliably achievable data rate and corresponding energy and information distributions. Also find $\bar{b}$. (4 pts)

e. If uncoded PAM/QAM must be used with a target $P_e = 10^{-6}$, now find the corresponding highest data rate. (2 pts)

f. Describe the MAP Decoder for part e, assuming that simple decoding is used - you may round bit distributions to closest integers and assume water-fill energies can be slightly adjusted (1 pts)

4.31 *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 $0 \le t \le T$ and is zero elsewhere generates the non-periodic DMT signal studied in Section 4.5. 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?

4.32 *Isaksson's Zipper*

For real baseband EPR6 channel $(1+D)^4 \cdot (1-D)$,

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?

4.33 *Kasturia's Block DFE - 15 pts*

Subsection 4.5.4 describes the so-called Block DFE for channel partitioning. This problem investigates the improvement for this guardband-less transmission method for the real-baseband $1 + .9 \cdot D^{-1}$ channel of the notes. Use all parameter values as always for this channel, except that we will assume this channel is causal and $1 + .9 \cdot D$. The gap will be 1 dB.

a. (2 pts) Find the 8×8 matrix $\hat{H}$ for $N = 8$ with the BDFE, and its associated SVD for vector coding.

b. (8 pts) Using water-filling for loading, what is the SNR for this system of parallel channels? Compare to VC and DMT.

c. (2 pts) Draw a picture of the receiver for the BDFE in this case and state where the extra complexity enters.

d. (3 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?

4.34 *TEQ Design for the* $1/(1+0.5 \cdot D)$ *Channel*

For the real-baseband $1/(1+.5D)$ 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 $\bar{b} = 1$.

4.35 *G.hn Home Network (10 pts)*

ITU Home Networking Standard G.9960 ("G.hn") uses DMT for transmission on various single or combinations of copper twisted pair, coaxial-cable, or power line wires. There are analog matching

circuits used between the different connections to allow impedance matching of electrical currents to flow smoothly from one segment to/from the next. G.hn can select Δf from the choices

$$\Delta f = 4.4140625 \cdot k \text{ kHz} \ , \ k = 2^l \ , \ l = 0, 1, ..., 6 \ .$$

FFT sizes are powers of 2 from 256 to 4096. The channels created are modulated by a separate carrier frequency in their entirety, and those carriers can be various select multiples of 25 MHz. The guard period is always 25% of the total number of non-prefix samples in a symbol. The maximum power transmitted is 15 dBm.

a. (1 pt) What is the sampling rate if $l = 4$ and $N = 4096$?

b. (1 pt) If flat energy were used on all channels, what would be the corresponding power spectral density (one-sided)?

c. (2 pts) What is the cyclic prefix length in samples and in absolute time units for your answer in Part a?

d. (2 pts) What is the symbol rate for your answer in Part a?

e. (1 pt) Suppose you know that twisted pairs have more significant attenuation increase with frequency than other copper channels. What is the likely value of l that you might then use?

f. (1 pt) What would the one-sided flat PSD be for your answer in part e?

g. (2 pts) Suppose circuit limitations for various purposes only allowed two choices of $\mathcal{E}_n$ consistent each within two contiguous bands of frequencies, where both bands have the same width in frequency and have been initially measured for SNRs with the same flat PSD. How might you design a loading algorithm for this situation?

4.36 *spatial equalizer* (12 pts)

This problem returns to the 3×3 MIMO spatial channel of Problem 4.25 and Figure 4.63. The input autocorrelation matrix is $\overline{R}_{\boldsymbol{xx}} = I$, while the noise autocorrelation is $\overline{R}_{\boldsymbol{nn}} = .01 \cdot I$, each normalized to the number of real dimensions.

a. What is the transmit energy per symbol for this system? (1 pt)

b. Find $\overline{R}_{\boldsymbol{xy}}$ and $\overline{R}_{\boldsymbol{yy}}$. (2 pts)

c. Find the MMSE estimator for $\boldsymbol{x}$ that uses the channel output $\boldsymbol{y}$ (1 pt)

d. Find the MMSE and $(SNR_{mmse})^{1/L_x}$. (2 pts)

e. If a threshold decision were made at each estimator (complex) dimension's output, and the MMSE error component on that dimension is viewed as the noise, what is the SNR_{mmse} for each such dimension? (3 pts)

f. Compare the answer for Part e to the corresponding answers in Problem 4.25, and explain any difference. (3 pts)

Bibliography

[1] Claude E. Shannon *A Mathematical Theory of Communication* Bell Systems Technical Journal, 1948. Vol. 27, No. 3, pp. 628-634.. See also *Communication in the Presence of Noise* Proceedings of the IRE, 1949 Vol. 37, No. 1, pp. 10-21.

[2] Geoffrey Li and Gordon Stuber, Editors. *"Orthogonal Frequency Division Multiplexing for Wireless Communications"*. See Chapter 3. Springer, USA, 2006.

[3] Thomas Starr, John M. Cioffi, and Peter. J. Silverman *"Understanding Digital Subscriber Line Technology"*. Prentice Hall, New Jersey, 1999.

[4] Stephen B. Weinstein *"The History of Orthogonal Frequency-Division Multiplexing "*. IEEE Communications Magazine. November 2009, pp. 26-35.

[5] L.H. Brandenburg and A.D. Wyner *"The Bell Systems Technical Journal"* Vol. 53, No. 5, May/June 1974, pp. 745 - 778.

[6] A. Paulraj and T. Kailath *"Increasing Capacity in Wireless Broadcast Systems Using Distributed Transmission/Directional Reception (DTDR)"* United States Patent 5,345,599. Priority Date 2/21/1992, Issue Date 9/6/1994.

[7] J. M. Cioffi and P. Voois *"Multidimensional Trellis Codes for Area Density Gains"* IBM Symposium on the Use of Signal Proccessing and Coding. Tucson, AZ, May 1985. See also Stanford Ph.D. Dissertation of Dr. Paul Voois, *"Two-dimensional data storage and retrieval"*, June 1994.

[8] *"Layered Space-Time Architecture for Wireless Communication in a Fading Environment When Using Multi-Element Antennas"* Bell Systems Technical Journal. Vol. 1, No. 2, pp. 41-59. Autumn, 1996.

[9] G. Ginis, W. Yu, C. Zeng, and J.M. Cioffi "Dynamic Digital Communication System Control" US patent 7,158,563 family (multiple patents issued) Filed June 8, 2001, Issued (first) 1/2/2007. See also "Vectored Transmission for Digital Subscriber Lines" IEEE Journal on Selected Areas in Communications Vol. 20 No. 5, June 2002, pp. 1085-1104.

[10] John M. Cioffi and G. David Forney, Jr. *"Generalized Decision-Feedback Equalization for Packet Transmission with ISI and Gaussian Noise"*. Communications, Computation, Control, and Signal Processing, a tribute to Thomas Kailath. Springer, Boston, MA ISBN 978-1-4615-6281-8. Editors: Paulraj A., Roychowdhury V., Schaper C.D. pp. 79-127.

[11] Haitham J. Taha and M.F.M. Saleh *" Multi-carrrier Transmission Techniques for Wireless Communication Systems: A Survey"* WSEAS Transactions on Communications, May 2009. Vol. 8, No. 5, pp. 457-472.

[12] James T. Aslanis, Jr. *"Coding for Communication Channels with Memory"* Ph.D. Dissertation, Stanford University, 1990.

[13] *"Bandwidth Optimized Digital Transmission Techniques for Spectrally Shaped Channels with Impulse Noise"* Ph.D. Dissertation, Stanford University, 1993.

[14] I. Bistritz and A. Leshem *"Game Theoretic dynamicl channel allocation for frequency selective interference channels"* IEEE Transactions on Information Theory. Vol 65., No 1, pp. 330-353, January 2019.

[15] Zhi-Quan Luo and Jong-Shi Pang "Analysis of Iterative Waterfilling Algorithm for Multiuser Power Control in Digital Subscriber Lines" EURASIP Journal on Applied Signal Processing. Vol 2006, Article ID 24012, p. 1-10, DOI 10.1155/ASP/2006/24012.

[16] Andrea Goldsmith "Wireless Communications " Cambridge University Press, NY, 2005.

[17] John M. Cioffi, Peter S. Chow, and Kenneth J. Kerpez "Optimum Equivalent Loading in Multi-Dimensional Transmission" IEEE Open Journal of the Communications Society. June 8, 2020, Vol. 1, pp. 681-699.

[18] Govanni Cherubini, Evangelos Eleftheriou, and Sedat Olcer "Filtered Multitone Modulation for Very High-Speed Digital Subscriber Lines" IEEE JSAC. Vol. 20, No. 5, June 2002. pp 1016-1028.

[19] Eldad Perahia and Robert Stacey "Nest Generation Wireless LANs: 802.11n and 802.,11ac" Cambridge University Press May 23, 2013.

[20] Vahid Tarokh and Hamid Jafakhani and A. Robert Calderbank. "Space-Time Block Codes from Orthogonal Designs" IEEE Transactions on Information Theory. Vol. 45, No. 5, Junly 1999. pp 1456-1467.

[21] Siavash Alamouti. "A Simple Transmitter Diversity Scheme for Wireless Communications" IEEE Journal on Selected Areas in Commnications. Vol. 16, October 1998. pp 1451-1458.

[22] Jacky S. Chow and John M. Cioffi and John A.C. Bingham. "Equalizer training algorithms for multicarrier modulation systems" Proceedings of the IEEE International Conference on Communications June 1993. pp 761- 765.

[23] Noafal Al-Dhahir and John M. Cioffi. "Optimum Finite-Length Equalization for Multicarrier Transceivers" IEEETransactions on Communications. Vol. 44, No. 1, January 1996, pp. 56-64.

[24] Guner Arslan and Brian Evans and Sayfe Kiaei. "Equalization For Discrete Multitone Transceivers To Maximize Bit Rate" IEEETransactions on Signal Processing. Vol. 49, No. 12, December 2001, pp. 3123-3135

[25] Sanjay Kasturia and John M. Cioffi. "Vector Coding with decision feedback equalization for partial-response channels" Proceedings Globecom 1988. Hollywood, FL, December 1988.

[26] K. Van Acker, and G. Leus and M. Moonen and T. Pollet. "Improved initialization for time domain equalization in ADSL" Signal Processing Magazine 2004. Vol 8, No. 10, pp.1895-1908.

[27] Kok-Wui Cheon and John M. Cioffi "Precoder for DMT with insufficient cyclic prefix" Proceedings of 1998 IEEE International Conference on Communications. June1998, Atlanta, GA., pp.339-343.

[28] Jose Tellado and John M. Cioffi "Efficient algorithms for reducing PAR in multicarrier systems" Proceedings. 1997 IEEE Symposium on Information Theory. Nov 1997, Cambridge, MA . pp.191. See also, PhD dissertation, Stanford University, "Multicarrier Modulation with Low Peak-to-Average Power," 1999.

[29] Alan Gatherer and Michael Polley "Controlling Clipping Probability in DMT Transmission". IEEE Asilomar Conference on Signals, Systems, and Computers. IEEE, November 2-5 1997, Asilomar, CA , pp. 578-584.

[30] A. Narula and Frank Kschischang, and Vedat Eyuboglu "A new approach to PAR control in DMT systems". IT Study Group 15, Question 4, contribution NF-083. May 1998.

Index