Generalized Decision Feedback Equalization

5.1 Generalized Channel Partitioning ................................................... 803
  5.1.1 Discrete-Modulator Decomposition ........................................... 805
    5.1.1.1 Discrete-Modulator Construction via channel singular vectors .......... 807
    5.1.1.2 Input Singularity Elimination ........................................... 812
  5.1.2 Finding $A$, the GDFE's Discrete Modulator .................................. 817
  5.1.3 Generalized Decision Feedback Development ..................................... 820
    5.1.3.1 The Generalized Linear Equalizer - GLE .................................. 829
  5.1.4 The Precoded GDFE ............................................................... 830
  5.1.5 Single-sided Receiver GDFEs .................................................. 831
  5.1.6 Single-sided Transmitter GDFE ............................................... 832
5.2 Special GDFEs: VC, Triangular DFE, and Massive MIMO ........................................ 833
  5.2.1 Vector-Coding as the GDFE without feedback .................................... 834
  5.2.2 The triangular GDFE ............................................................ 835
  5.2.3 The Circulant DFE (CDFE) ...................................................... 838
  5.2.4 Diagonal Dominance and Massive MIMO Linear GDFE Simplification ............ 841
    5.2.4.1 Generalized Cholesky Algorithm ........................................... 843
  5.2.5 The Tonal GDFE ....................................................................... 845
    5.2.5.1 Tonal GDFE Calculation: using the correct noise-whitened channel equivalent ............................................................... 846
5.3 GDFE Transmit Optimization ............................................................ 850
  5.3.1 The channel-dependent optimized input .......................................... 850
    5.3.1.1 nonzero Gap Inputs .......................................................... 851
  5.3.2 GDFE SNR Maximization over $R_{xx}$ ........................................... 852
  5.3.3 Triangular GDFE $R_{xx}$ Optimization .......................................... 853
    5.3.3.1 Reordering singular-autocorrelation dimensions ......................... 853
  5.3.4 CDFE with Causal Triangular input ............................................. 854
    5.3.4.1 IFFT matrices ..................................................................... 856
  5.3.5 Some Simplifying ZF/MMSE-Equivalence Relationships .......................... 873
    5.3.5.1 Water-fill Simplifications .................................................... 873
    5.3.5.2 Worst-case noise simplifications: .......................................... 875
  5.3.6 Asymptotic Stationary Convergence of certain GDFEs ............................. 881
    5.3.6.1 Stationary Channels ............................................................ 881
    5.3.6.2 Canonical Factorization for finite- and infinite-length stationary sequences 882
    5.3.6.3 Convergence of the Canonical Channel Models and the CDFE .............. 883
5.4 The Gaussian MAC and GDFE ............................................................. 887
  5.4.1 The GDFE's Relationship to the MMSE MAC ...................................... 887
    5.4.1.1 MACs with intersymbol interference ........................................ 890
  5.4.2 Vector DMT for the MAC ............................................................ 894
    5.4.2.1 Tonal GDFE Specification ..................................................... 898
Generalized Decision Feedback Equalization (GDFE) theory is a powerful transceiver-design approach that jointly addresses space, time, and frequency dimensionality. GDFE’s were introduced by this text’s author and G. David Forney in [1] (1994). GDFE’s fundamentally allow a common approach to many MIMO systems (later appearing under various names) and to most matrix-AWGN mutliuser designs. This chapter then completes this text’s modulation-coding methods, while later chapters address supporting systems like timing-synchronization (Chapter 6), channel identification and adaptive tracking (Chapter 7), decoder algorithms and architecture (Chapter 8), and code designs in Chapters 9 - 10. This chapter thus attempts to “put it all together” and create an insightful design methodology that fundamentally addresses most all linear communication systems with additive Gaussian noise.

Chapter 4 introduced symbols with large temporal dimensionality, specifically a finite $N < \infty$ is large. This chapter further develops such large symbols to a class of canonical GDFE transmission designs, of which vector coding, and its special cyclic-prefixed case DMT/C-OFDM, are special cases. This approach further expands into $\mathcal{N}$ sets of $L = \min(L_x, L_y)$-dimensional GDFEs for each (temporal) index $n = 1, ..., \mathcal{N}$. Some of Chapter 2’s MAC and BC systems are also GDFE special cases, and indeed Chapter 3’s MMSE-DFE is a certain GDFE’s asymptotic limiting case. This chapter then ties together the results for better understanding and extrapolates to more complete designs beyond earlier chapters’ scope, particularly for Chapter 2’s multi-user designs. In general, this matrix channel is $\mathcal{N}_y \times \mathcal{N}_x$, where $\mathcal{N}_x$ is the total number of temporal input sub-symbol dimensions, and $\mathcal{N}_y$ is the number of output sub-symbol dimensions, with then $N_x = \mathcal{N}_x \cdot \mathcal{N}_y$.

A code may often have codewords that concatenate several sub-symbols. These sub-symbols become $\mathcal{N}_x \cdot L_x$-dimensional sub-symbols when there is MIMO. An outer AEP-sense codeword (See Chapter 2, Section 3) is tacit, particularly when results with $\Gamma = 0$ dB codes appear. Codewords consist of $\mathcal{N}_x \cdot L_x$-dimensional sub-symbols when there is MIMO. This chapter thus focuses on a large dimensional sub-symbol and its canonical “GDFE” processing, presuming tacitly that some outer larger concatenated code is also (typically) present, and that outer code’s codewords span many such sub-symbols. Indeed, Chapter 4’s separation theorem allows good AWGN-channel codes to be independently and tacitly applied outside the GDFE analysis so that any good code can be used with any good GDFE transceiver system. It will be convenient to call the $\mathcal{N}_x \cdot L_x$-dimensional concatenation a codeword, and consequently any AEP-sense longer collection of symbols a codeword. Real (1D) or Complex (2D) elements within the symbols are the sub-symbols. Chapter 4’s Separation Theorem continues to apply so that good (simple scalar) AWGN codes equally well apply to linear matrix AWGN channels, although now to many GDFE-designed structures that all have the same geometric SNR and the mutual information.

Section 5.1 on channel partitioning investigates a general matrix-AWGN channel that leads to canonical transmission systems, introducing the GDFE and proper single-user singularity handling. Section 5.2 then shows that existing known optimal or canonical channel-partitioning structures like Chapter 4’s vector coding, discrete multi-tone, coded orthogonal-division-frequency-multiplexed, and vectored combinations are all special GDFE cases. Section 5.2 then finds very special case of the circulant DFE that re-uses DMT/OFDM’s cyclic prefix and asymptotically converges to (a minimal set of) Chapter 3’s MMSE-DFE(s). Section 5.2 also investigates conditions, sometimes known as diagonally dominant or “massive MIMO,” that allow linear simplification with small performance loss. Also a linear-receiver loss procedure addresses situations where the GEF’s nonlinear signal processing does not occur. Section 5.3 generalizes Chapters 3’s and 4’s transmit optimization for the single-user GDFE-based canonical design, finding that these often suboptimal detector structures reliably achieve capacity, or the highest data rates, and so are canonical.

Section 5.4 revisits Chapter 2’s Gaussian MAC as a GDFE and addresses the $R_{xx}$ optimization either to achieve a given data-rate vector $b$ with a (possibly weighted) minimum amount of energy or to achieve a best (possibly weighted or prioritized) $b$ with a given energy. Chapter 2’s systems had either a specific $R_{xx}$ or suggested search over all possibilities to trace a rate region - instead Section 5.4 introduces the concept of admissibility (whether a certain rate vector is possible to achieve directly without knowing the capacity region) and then how to design to achieve it with a GDFE receiver design for the given physically separated inputs. Expansion to design of a vectored DMT system with separate spatial tonal GDFE processing on each dimension/tone also appears in Section 5.4.
Section 5.5 similarly progresses the Gaussian BC channel, with both some direct GDFE examples as well as use of a duality concept that arises naturally from GDFE concepts and achieves all Gaussian BC capacity-region points through a series of appropriately precoded GDFE’s, on for each physically separated output. This dual-GDFE-set approach then circumvents need to use Chapter 2’s more complex WCN-based designs. Section 5.6 then progresses to the Gaussian IC channel, again use GDFE concepts in every user connection, but also spectrum optimization concepts across the entire set of user channels. Software analysis/design programs appear throughout this chapter to assist designers in evaluating the various design options.

Section 5.7 broadens analysis to statistically characterized channels.
5.1 Generalized Channel Partitioning

This section finds and describes the many equivalent matrix-AWGN channel modulation choices that result in sets of independent simple scalar-AWGN subchannels, all with the same geometric-average SNR and bits/symbol. Chapter 4 scalar-channel sets arose from discrete Fourier transforms and/or singular value decomposition. Alternately, Chapter 3’s minimal set of infinite-length MMSE-DFE’s is another equivalent set that can sometimes match Chapter 2’s (and 4’s) MT transmission design; and indeed both achieve the same best-coded data rate of $I(X(D); Y(D))$, for a given common power spectral density $R_x(e^{-j\omega T})$ and with gap $\Gamma = 0$ dB. MT systems partition the channel into an infinite number of scalar-AWGN subchannels that each carry a different information per subsymbol dimension, while by contrast each (of the set of) MMSE-DFE system(s) partition the channel into an infinite set of AWGN subchannels that each carry the same amount of information. This later system is also the limit of C-OFDM with a single good code’s constellation used on the same set dimensions with the same $R_{xx}$ and associated performance, as another instance of Chapter 4’s Separation Theorem.

Discrete Matrix Channel Models: This section re-investigates Chapter 2’s (and 4’s) vector-coding (VC) partitioning for the matrix additive-Gaussian-noise channel:

$$y = H \cdot x + n.$$  (5.1)

A purely space-time MIMO channel is $L_y \times L_x$ with $N_x = N_y = 1$ and up to $L_x$ and/or $L_y > 1$ dimensions. The SISO channel may also have, in frequency-time, $N_x = N + \nu > 1$ dimensions per subsymbol. The subsymbol nominal output dimensionality is $N = 1$ for real, and $N = 2$ for complex with Chapter 4’s Vector DMT, and then $N_x = L_x \cdot (N + \nu)$. This chapter’s developments also hold for any general $H$ of any dimensionality. This more general $N_y \times N_x$ matrix $H$ allows channel models that may expand across time/frequency and space - for instance multiple electromagnetic wave modes on a single (carrier) wavelength in waveguides (fiber or copper), angular orbital momentum dimensions, or even quantum-communication dimensions. A Toeplitz $H$ structure only applies for stationary time/frequency systems that lead to the IFFT/FFT’s use for DMT/OFDM partitioning, but such Toeplitz structure does not necessarily hold more generally, particularly in space-time. One interesting non-Toeplitz $H$ choice models time-varying convolution where the successive zeros in rows of $H$ shift as with convolution, but each row’s shifted nonzero entries vary with respect to the previous row. Such time-varying models may be useful in wireless mobile transmission systems. Other $H$ choices model crosstalk between different copper wires in a cable. Chapter 4’s VC with singular value decomposition (SVD) also did not restrict $H$ to have any structure.

As in Chapters 2 and 4, a geometric product SNR’s with singular values $\lambda_n$, and each with equal-variance $\frac{\lambda_n^2}{2}$ additive white Gaussian noise, is (for $\Gamma = 0$ dB):

$$\text{SNR}_{vc,u} + 1 = \left[ \prod_{n=1}^{\infty} \left( 1 + \frac{\lambda_n \cdot \lambda_n^*}{N_2} \right) \right]^{\frac{1}{N_2}} = \left\{ \frac{|R_{yy}|}{|R_{nn}|} \right\}^{\frac{1}{N_x}} = 2^{\frac{I(x; y)}{N}}. \quad (5.2)$$

Since Chapter 4’s VC uses the optimum finite-length partitioning for any input autocorrelation matrix $R_{xx}$, VC attains the highest achievable rate bound $I(x; y)$ for this input. Any one-to-one transformation of either $x$ or $y$ to another random vector cannot change the mutual information as per Section 2.3’s Preservation of Information Lemma. This $I$-preservation admits discovery of other equivalent structures and sets of parallel channels that perform the same for the given input autocorrelation function $R_{xx}$.

1See sections 3.11 and 3.12.
2Of the random process $X(D)$ and codes with gap $\Gamma = 0$ dB, where $X(D)$ is the discrete transform $(D \rightarrow e^{-j\omega T}$ for Fourier Transform) such that $X(D) = \sum_{k=-\infty}^{\infty} x_k \cdot D^k$ with $R_x(D)$ defined in a limiting sense such that $R_x(D) = \lim_{K \rightarrow \infty} \frac{1}{2\pi T} \sum_{k=-\infty}^{\infty} r_{xx,k} \cdot D^k$ where the stationary $x_k$ has $r_{xx,k} \equiv E[x_k \cdot x_n - k]$.
3 with correct decisions implied by $\Gamma = 0$ dB.
4For instance, Stanford Prof. Joseph Kahn and others have modeled intermodal crosstalk in optical fibers with such a matrix channel.
Channel-Partitioning Characterization: Figure 5.1 overviews partitioning. There are infinitely many channel partitions that all can achieve the same aggregate data rate $b = I(x; y)$ over each its corresponding set of parallel independent simple scalar-AWGN subchannels. Generalized Decision Feedback Equalization, or the “GDFE,” describes and implements all these partitions. VC (and its special case DMT with cyclic prefixes) are the simplest and most natural partitions for a temporal channel, but many exist. They are also linear and have optimal (ML) detectors. Nonlinear detectors use interim decisions (or modulo precoders, as in Chapter 2) and reliably achieve best data rates, despite non-ML detectors. An interesting nonlinear system for Chapter 3’s ISI channels is Section 5.2’s Circulant Decision Feedback, or CDFE. This CDFE, under an assumption of channel stationarity over all sampling periods, converges (with care to satisfy Appendix D’s Paley Weiner Criterion or PWC) to Chapter 3’s canonical minimal-size set of MMSE-DFEs. VC and DMT, under the same stationarity assumption, converge to Chapters 2 and 4’s MT.

Subsection 5.1.1 describes channel-input decomposition into a channel-dependent deterministic vector space and a random input-dependent (Hilbert) space corresponding to the autocorrelation matrix $R_{xx}$. Subsection 5.1.2 then shows that this decomposition extracts a finite-length canonical nonsingular replacement channel that is nonsingular and has the original-channel’s mutual information. With this equivalent canonical channel, the GDFE then easily follows through canonical/Cholesky factorization of the information-bearing input component in Subsection 5.1.3. Subsection 5.1.4 updates Chapter 2’s lossless precoders precoders, as well as those of Section 3.8, for the GDFE. Subsections 5.1.5 and 5.1.6 provide some interesting one-sided results for special channel structures that generalize those earlier found for Chapter 2’s MAC and BC, respectively.
5.1.1 Discrete-Modulator Decomposition

Figure 5.2 illustrates the division of the $N_x$-dimensional input vector space $C^{N_x}$ into two mutually orthogonal subspaces: a pass space $P_H$ containing input vectors that pass through the channel so that $H \cdot x \neq 0$ if $x \in P_H$ and a null space $N_H$ containing input vectors that the channel nulls so that $H \cdot x = 0$ if $x \in N_H$.

The null space and pass space are mutually orthogonal. In general, an input vector can have nonzero components in both the pass and the null space, but those components in the null space clearly waste transmit energy because none pass to the channel output. Thus, only pass-space components carry information through the channel. A vector that has no null-space component is Figure 5.2’s $\tilde{x}$ (such a vector fails Appendix D’s PWC’s finite-length vector equivalent if the null space has dimension 1 or larger, which means that the input autocorrelation matrix is singular, $|R_{xx}| = 0$).

Discrete-matrix modulator and pass-space projection: Figure 5.2’s matrix-AWGN-channel transmitter has discrete-modulator

$$x = \sum_{n=1}^{N_x} u_n \cdot c_n = C \cdot u ,$$

where the vectors $c_n$, $n = 1, ..., N_x$ need not be an orthonormal set, and where the random-message components multiplying these vectors, $u_n$, need not be independent. The discrete-modulator vectors $\{c_n\}_{n=1..N_x}$ project into the channel’s rank-$\rho_H$ pass space $P_H$ to obtain a new set of $\rho_H \leq N_x$ linearly independent vectors $\{\tilde{c}_n\}_{n=1..\rho_H}$, that characterize $\tilde{x} \in P$ by

$$\tilde{x} = \sum_{n=1}^{\rho_H} \tilde{u}_n \cdot \tilde{c}_n = \tilde{C} \cdot \tilde{u} .$$

---

5There are $2(N_x)$ real dimensions when the input vector is complex.

6Said failure is not usually a problem for finite-length realizations with delay, but it’s limiting case creates non-realizable filters.

7A special case is when the input components $u_n = X_n$ are independent and the vectors $c_n = m_n$ are the channel-dependent, SVD-derived, orthonormal basis of vector coding; but this need not be the case in general.
Possible \( \tilde{x} \) values span only the \( q_H \)-dimensional pass subspace of the \( N_H \)-dimensional complex vectors \( \mathbb{C}^{N_H} \) and are then, by construction, in one-to-one correspondence with \( \tilde{u} \) values (because of the linear independence of the vectors \( \{ \tilde{c}_n \}_{n=1}^{q_H} \) in Equation (5.4)). Example 5.1.1.1 provides a method to compute both \( \tilde{u} \) and \( \tilde{C} \) from the channel matrix \( H \) and from an arbitrary \( u \) and \( C \). The vector \( \tilde{x} \) is the finite-length discrete-time-vector equivalent of a continuous-time modulated input signal \( x(t) \) having no spectral component in the channel’s dead-zones, i.e., those frequency ranges for which \( H(f) = 0 \) (or also those that fail Appendix D’s PWC, as in Chapter 3). Figure 5.3 illustrates \( x \)'s projection into \( \mathcal{P}_H \).

![Figure 5.3: Projection illustration using singular vectors.](image)

**Mutual-Information Preservation:** A general input may have nonzero null-space components, so

\[
R_{xx} \neq R_{\tilde{x}\tilde{x}}.
\]  
(5.5)

However, any lost \( x \) components (not in \( \tilde{x} \)) carry zero information through channel \( H \). The (Gaussian) input has entropy \(^8\)

\[
\mathcal{H}_x = \mathcal{H}_{x|\mathcal{N}_H} + \mathcal{H}_{x|\mathcal{P}_H} \geq \mathcal{H}_{\tilde{x}}.
\]  
(5.6)

However, basic MMSE estimation relationships with entropy (Gaussian \( R_{nn} = I \)) then also lead to \(^9\)

\[
\mathcal{H}_{x|y} = \mathcal{H}_{x|\mathcal{N}_H} + \mathcal{H}_{x|\mathcal{P}_H} \geq \mathcal{H}_{\tilde{x}|y}
\]  
(5.7)

and thus

\[
I(x; y) = \mathcal{H}_x - \mathcal{H}_{x|y}
\]  
(5.8)

\[
= \mathcal{H}_{x|\mathcal{N}_H} + \mathcal{H}_{x|\mathcal{P}_H} - \mathcal{H}_{x|\mathcal{N}_H} - \mathcal{H}_{x|\mathcal{P}_H} - \mathcal{H}_{\tilde{x}|y}
\]  
(5.9)

\[
= \mathcal{H}_{\tilde{x}} - \mathcal{H}_{\tilde{x}|y}
\]  
(5.10)

\[
I(\tilde{u}; y) = I(\tilde{x}; y) = I(x; y)
\]  
(5.11)

so that mutual information remains the same, an intuitively gratifying result – coding/transmitting into the channel’s null space does not increase the achievable data rate. Subsection 5.1.1.1 illustrates one method for determining \( \mathcal{N}_H \) and \( \mathcal{P}_H \) and for then finding the information-equivalent vector \( \tilde{x} \) with components \( \tilde{u} \) only in the channel’s pass space. Further, since \( \tilde{u} \) is in one-to-one correspondence with possible \( \tilde{x} \) values, then

\[
I(\tilde{u}, y) = I(\tilde{x}; y) = I(x; y)
\]  
(5.12)

\(^8\)This equation follows by viewing entropy as the sum of scalar terms based on \( R_{xx} \)'s eigendecomposition, which may have nonzero component in \( \mathcal{N}_H \). Section 2.3 discusses entropy.

\(^9\)Recall the null space does not pass to \( y \) so \( H_{x|\mathcal{N}_H|y} = H_{x|\mathcal{N}_H} \).
5.1.1.1 Discrete-Modulator Construction via channel singular vectors

Channel Singularity Elimination: Figure 5.4 illustrates an algorithm that eliminates \( x \) components in \( \mathcal{N}_H \). The noise-equivalent channel matrix \( \tilde{H} = R_{\bar{N}x}^{-1/2} \cdot H \) has SVD \( \tilde{H} = F \cdot \Lambda \cdot M^* \) or

\[
\tilde{H} = \sum_{n=1}^{\bar{H}} \lambda_n \cdot f_n \cdot m_n^* \quad \text{(step 1 in Fig. 5.4),}
\]

(5.13)

where \( \bar{H} \leq \bar{N}_x \) is the channel rank and corresponds to the number of nonzero singular values. The singular values are non-negative real and in decreasing order \( \lambda_1 \geq \lambda_2 \geq ... \geq \lambda_{\bar{H}} \geq 0 \) without loss of generality. The pass space \( \mathcal{P}_H \) is then spanned by the orthonormal set of column vectors\(^{10}\) of the \( \bar{N}_x \times \bar{H} \) matrix

\[
\tilde{M} = [m_{\bar{H}} \ m_2 \ ... \ m_1] \quad \text{(step 1 in Fig. 5.4).}
\]

(5.14)

Then,

\[
\tilde{x} = \tilde{M} \cdot x
\]

(5.15)

where the Hermitian matrix \( \tilde{M}^* \) is a “projection matrix” in mathematics

\[
\tilde{P}_M = \tilde{M} \cdot \left( \tilde{M}^* \cdot \tilde{M} \right)^{-1} \cdot \tilde{M}^* = \tilde{M} \cdot \tilde{M}^* \quad \text{(step 2 in Fig. 5.4).}
\]

(5.16)

\( M \)'s remaining column vectors similarly span the null space \( \mathcal{N}_H \). The projection \( \mathcal{P}_H \) of each of the \( \bar{N}_x \)-dimensional modulation vectors \( \tilde{c}_n \) is

\[
\tilde{c}_n = \tilde{P}_M \cdot c_n
\]

(5.17)

may be compactly rewritten (possibly for easy matlab construction) as

\[
\tilde{C}_{temp} = \tilde{P}_M \cdot C \quad \text{(step 3 in Fig. 5.4).}
\]

(5.18)

Also,

\[
R_{\bar{N}_x} \tilde{x} = \tilde{P}_M \cdot R_{\bar{N}_x} x \cdot \tilde{M}
\]

(5.19)

The matrix \( C \) has no more than \( \bar{N}_x \), \( \bar{N}_x \)-dimensional, column vectors that correspond to the original modulator for \( x \). The matrix \( \tilde{C}_{temp} \subseteq C \) is a rank-\( \bar{H} \) submatrix of \( \bar{N}_x \)-dimensional column vectors. This \( \tilde{C}_{temp} \) matrix thus contains \( \bar{H} \) linearly independent columns, which reindex (the designer must record the reindexing in practice) so that

\[
\tilde{C}_{temp} = [\tilde{C} \mid \tilde{C}_1] \quad \text{(step 3 in Fig. 5.4).}
\]

(5.20)

The \( \bar{N}_x \times \bar{H} \) matrix \( \tilde{C} \) corresponds to the linearly independent vectors/columns of \( \tilde{C}_{temp} \), and the remaining dependent column vectors\(^{11}\) are in \( \tilde{C}_1 \). Then, similarly reindexing the \( u \) components into the \( \bar{H} \times 1 \) vector \( u_2 \) for \( \tilde{C} \) and the remaining components\(^{12}\) in vector \( u_1 \) for \( \tilde{C}_1 \), \( \tilde{x} \) becomes

---

\(^{10}\)Matlab places lowest index on right or bottom, while this text has that order reversed. To reverse order \( F \rightarrow F \cdot J_{\bar{N}_x} \), \( \Lambda \rightarrow J_{\bar{N}_x} \cdot \Lambda \cdot J_{\bar{N}_x} \), and \( M \rightarrow M \cdot J_{\bar{N}_x} \).

\(^{11}\)This separation is often obvious, but a general procedure follows the use of the command “qr” in matlab. By executing the command \( \{Q,R,J\}=qr(P_{\bar{M}} \cdot C)\),” then \( \bar{H} \) is obtained by the command \( \text{rank}(P_{\bar{M}} \cdot C)\),” and then \( \tilde{C} \) is the first \( \bar{H} \) columns of the matrix formed by \( Q \cdot R \cdot J \) in matlab and the last \( \bar{N}_x \) \( \bar{H} \) columns are then \( \tilde{C}_1 \). The rearrangement of inputs by \( J \) needs to be noted and used by the designer also. See also the Appendix E program eliminate.x, nullspace.m and lichols.m commands.

\(^{12}\)Index reversal with respect to largest at top/left here is temporary to match matlab commands, and this indexing is transitional only and eventually disappears.
\[
\tilde{H} = R_{nn}^{-1/2} \cdot H
\]

\[
M = \begin{bmatrix}
  m_{\tilde{N}_x} & \cdots & m_{Q_H+1} \\
  \vdots & \ddots & \vdots \\
  m_{Q_H} & \cdots & m_1
\end{bmatrix}_{N_H \times \tilde{M}}
\]

\[
\tilde{H} = R_{\tilde{M}} \cdot \tilde{M}^*
\]

\[
P_{\tilde{M}} = \tilde{M} \cdot \tilde{M}^*
\]

\[
\tilde{\text{temp}} = P_{\tilde{M}} \cdot C = \underbrace{\tilde{\mathcal{C}}}_{\text{ind}} \underbrace{\tilde{\mathcal{C}}_1}_{\text{dep}}
\]

\[
\mathcal{O} = \left[ \tilde{\mathcal{C}}^* \tilde{\mathcal{C}} \right]^{-1} (\tilde{\mathcal{C}}^* \tilde{\mathcal{C}}_1)
\]

\[
u = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix}
\]

\[
R_{uu} = \begin{bmatrix} R_{22} & R_{21} \\ R_{12} & R_{11} \end{bmatrix}
\]

\[
\tilde{x} = \tilde{P}_{\tilde{M}} \cdot x = \tilde{C} \cdot \tilde{u}
\]

\[
\tilde{R}_{\tilde{u}\tilde{u}} = R_{22} + \mathcal{O} \cdot R_{12} + \mathcal{O}^* \cdot R_{21} + \mathcal{O} \cdot R_{11} \cdot \mathcal{O}^*
\]

Figure 5.4: Algorithm for elimination of \( x \) components in channel null space.

\[
\tilde{x} = P_{\tilde{M}} \cdot x
\]

\[
= P_{\tilde{M}} \cdot \tilde{C} \cdot u
\]

\[
= \tilde{C}_{\text{temp}} \cdot u
\]

\[
= \begin{bmatrix} \tilde{C} & \tilde{C}_1 \end{bmatrix} \begin{bmatrix} u_2 \\ u_1 \end{bmatrix}
\]

\[
= \tilde{C} \cdot u_2 + \tilde{C}_1 \cdot u_1
\]

The new input autocorrelation matrix \( R_{uu} \) then can be partitioned according to the same labeling

\[
R_{uu} = \mathbb{E} \left\{ \begin{bmatrix} u_2 \\ u_1 \end{bmatrix} \begin{bmatrix} u_2^* & u_1^* \end{bmatrix} \right\} = \begin{bmatrix} R_{22} & R_{21} \\ R_{12} & R_{11} \end{bmatrix}
\]
The remaining matrix vectors/columns $\tilde{C}_1$ are linearly dependent on $\tilde{C}$ and have projections on $\tilde{C}$ as

$$\tilde{C}_1 = \tilde{C} \cdot [\tilde{C}^* \cdot \tilde{C}]^{-1} \cdot \tilde{C}^* \cdot \tilde{C}_1.$$  \hfill (5.27)

(5.27) defines a matrix $\tilde{O}$ by

$$\tilde{O} = \tilde{C} \cdot \left\{ [\tilde{C}^* \cdot \tilde{C}]^{-1} \cdot \tilde{C}^* \cdot \tilde{C}_1 \right\} = \tilde{C} \cdot \tilde{O} \quad \text{(step 4 in Fig. 5.4)},$$

essentially a matrix post-multiply by $\tilde{O}$ to compute $\tilde{C}_1$ from $\tilde{C}$ in (5.20). The matrix $\tilde{O}$ is more helpful than $\tilde{C}_1$ itself as follows: Step 5 in Figure 5.4 shows explicitly the decomposition of $u$ into the $\varrho_H$ components $u_1$ that multiply $\tilde{C}$ and the $N_x - \varrho_H$ components $u_2$ that multiply $\tilde{C}_1$. These two components combine into a single component $\tilde{u}$ according to

$$\tilde{x} = \tilde{C} \cdot u_2 + \tilde{C} \cdot \tilde{O} \cdot u_1 = \tilde{C} \cdot (u_2 + \tilde{O} \cdot u_1) = \tilde{C} \cdot \tilde{u} \quad \text{(step 6)}$$

So the $\varrho_H$-dimensional $\tilde{u}$ forms from $\tilde{O}$ alone by

$$\tilde{u} \triangleq u_2 + \tilde{O} \cdot u_1 \quad \text{(step 6)}.$$  \hfill (5.30)

Then, the designer simply discards $\tilde{C}_{\text{temp}}$ and uses instead $\tilde{C}$. The new (more useful) input’s autocorrelation matrix is

$$R_{\tilde{u}\tilde{u}} = R_{22} + \tilde{O} \cdot R_{12} + R_{21} \cdot \tilde{O}^* + \tilde{O} \cdot R_{11} \cdot \tilde{O}^* \quad \text{(Step 8 in Fig. 5.4).} \hfill (5.31)$$

**Channel Equivalences:** Figure 5.5 illustrates the original channel (with input $u$) and with channel null-space components removed (with input $\tilde{u}$). The outputs are exactly the same in both cases. The lower equivalent channel captures all necessary information about the original channel while eliminating any useless transmission components that the channel singularity zeros.

---

Figure 5.5: The original channel and its equivalent with channel singularity removed.
Appendix G's matlab program fixmod.m (and its called lichols.m) execute the commands in Figure 5.4 directly. The program comments are

```matlab
function [Ct, Ot, Ruutt] = fixmod(H_NW, Ruu, C, tol)

% fixmod removes the part of x that lies in H's nullspace
x->xt, u->ut
% Inputs: H_NW, Ruu, C, tol
% H_NW: Noise-whitened channel
% Ruu: Autocorrelation matrix of u
% C: discrete modulator matrix
% tol: used in licols to determine rank of matrix Ctemp
%
% Outputs: Ct, Ot, Ruutt
% Ct: new discrete modulator with components without H's nullspace components
% Ot: Projection matrix of C2 onto C
% Ruutt: new autocorrelation matrix for u
%
% This program calls licols
```

```matlab
function [Xsub,idx]=licols(X,tol,mode)

% Extract a linearly independent set of columns of a given matrix X

% Inputs:
% X: The given input matrix
% tol: A rank estimation tolerance. Default=1e-10
% mode: Rule for choice of columns
%   'desc': choose columns based on descending values of R
%   'LR': choose columns from left to right

% out:
% Xsub: The extracted columns of X
% idx: The indices (into X) of the extracted columns
```

**EXAMPLE 5.1.1 (1 + D Channel)** An example $2 \times 3$ matrix ($R_{nn} = I$, so $H = \tilde{H}$)

$$
H = \begin{bmatrix}
1 & 1 & 0 \\
0 & 1 & 1
\end{bmatrix}
$$

(5.32)

corresponds to Chapter 3's $1 + D$ channel with $N = 2$ and $\nu = 1$, so $N_x = 3; N_y = 2$. This channel's SVD is

$$
H = F \cdot \begin{bmatrix}
\sqrt{3} & 0 & 0 \\
0 & 1 & 0
\end{bmatrix} \begin{bmatrix}
\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\
\frac{1}{\sqrt{3}} & 0 & -\frac{1}{\sqrt{3}} \\
\frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}}
\end{bmatrix}^* 
$$

(5.33)

---

13Thank you to 2021 graduate student Ethan Liang for this program.
\[ \widetilde{M} = \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{6}} & 0 \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} \end{bmatrix} . \]  

(5.34)

Then
\[ P_{\widetilde{M}} = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} & -\frac{1}{3} \\ \frac{1}{3} & \frac{2}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{2}{3} \end{bmatrix} . \]  

(5.35)

Then with a white 3-dimensional PAM input, \( x = u \) or equivalently \( C = I \) so that
\[ \widetilde{x} = P_{\widetilde{M}} \cdot x , \]  

(5.36)

\( C_{\text{temp}} \) follows as
\[ C_{\text{temp}} = P_{\widetilde{M}} \cdot C = P_{\widetilde{M}} \cdot I = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} & -\frac{1}{3} \\ \frac{1}{3} & \frac{2}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{1}{3} & \frac{2}{3} \end{bmatrix} . \]  

(5.37)

Thus
\[ \tilde{C} = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{2}{3} \\ -\frac{1}{3} & \frac{1}{3} \end{bmatrix} \quad \text{and} \quad \tilde{C}_1 = \begin{bmatrix} -\frac{1}{3} \\ \frac{1}{3} \\ \frac{2}{3} \end{bmatrix} . \]  

(5.38)

The matrix \( \tilde{O} \) is then
\[ \tilde{O} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} . \]  

(5.39)

The new two-dimensional input \( \tilde{u} \) is then
\[ \tilde{u} = \begin{bmatrix} \tilde{u}_2 \\ \tilde{u}_1 \end{bmatrix} = u_2 + \tilde{O} \cdot u_1 = \begin{bmatrix} u_3 \\ u_2 \end{bmatrix} + \begin{bmatrix} -1 \\ 1 \end{bmatrix} u_1 = \begin{bmatrix} u_3 - u_1 \\ u_2 + u_1 \end{bmatrix} . \]  

(5.40)

The unchanged channel output is
\[ y = \tilde{H} \cdot P_{\widetilde{M}} \cdot C \cdot u + n = \tilde{H} \cdot \tilde{x} + n \]  

(5.41)

\[ = \tilde{H} \cdot \tilde{C} \cdot \tilde{u} + n \]  

(5.42)

\[ = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{u}_2 \\ \tilde{u}_1 \end{bmatrix} + n . \]  

(5.43)

This final channel-output model is, in terms of just those components of \( x \), \( \tilde{u}_2 = u_3 - u_1 = x_3 - x_1 \) and \( \tilde{u}_1 = u_2 + u_1 = x_2 + x_1 \), that “pass” through \( \tilde{H} \). Furthermore, it is clear that any input component with nonzero projection on the vector
\[ m_1 = \begin{bmatrix} \frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \end{bmatrix} , \]  

(5.44)

for instance the PAM input
\[ \begin{bmatrix} -1 \\ -1 \\ 1 \end{bmatrix} , \]  

(5.45)

will not pass to the channel output. Thus, the two codewords +,-,+ and -,+,- that nominally look quite separated and good for coding with a minimum distance of \( +2 \sqrt{3} \) could not be
distinguished at the channel output. Any codewords containing nonzero differences of this type simply waste energy as far as passage through the channel. For this channel, better input design than PAM would let only \( \tilde{E}_{u_1} \) and \( \tilde{E}_{u_2} \) be nonzero while avoiding any “oscillating” difference components (i.e., \(+, -, +\) or \(-, +, -\)) between codewords.

Using the fixmod command in matlab produces:

\[
\begin{align*}
&\text{>> H_NW=[1 1 0; 0 1 1];} \\
&\text{>> C=eye(3);} \\
&\text{>> Ruu=eye(3);} \\
&\text{>> [Ct, Ot, Ruutt] = fixmod(H_NW, Ruu, C)}
\end{align*}
\]

\[
\begin{align*}
Ct &= \\
&\begin{bmatrix}
0.6667 & 0.3333 \\
0.3333 & 0.6667 \\
-0.3333 & 0.3333
\end{bmatrix} \\
Ot &= \\
&\begin{bmatrix}
-1.0000 \\
1.0000
\end{bmatrix} \\
Ruutt &= \\
&\begin{bmatrix}
2.0000 & -1.0000 \\
-1.0000 & 2.0000
\end{bmatrix}
\end{align*}
\]

which indeed matches (5.38) and also provides \( \tilde{R}_{uu} \).

In general for noise-whitened channel \( \tilde{H} = R_{nn}^{-1/2} \cdot H \), modulator pass-space reduction so far uses \( C \)'s fixed discrete-modulator vectors and the deterministic channel \( \tilde{H} \), in particular through \( \tilde{M} \). The pass space \( \tilde{P}_H \) and the null space \( \tilde{N}_H \) depend only upon the channel matrix \( \tilde{H} \). However, the input vectors \( x \) are random, constructed through modulation of a random component \( u_n \) with each of the discrete-modulator vectors, \( u_n \cdot c_n \). Thus, the reduction so far to \( x \) depends only on input’s deterministic constituents, namely \( \tilde{C} \). The projected vectors \( \tilde{x} \) themselves span a random vector (“Hilbert”) space that depends (Gaussian case) on \( \tilde{R}_{xx} \). This random vector space rarely occurs by accident, and usually occurs by design for optimum or good \( \tilde{R}_{xx} \) (or \( \tilde{R}_{\tilde{x}\tilde{x}} \)); such optima may have lesser rank or dimensionality such that \( \rho_x < \rho_H \). The discrete equivalent of the PWC channel-input realization is that \( \tilde{R}_{\tilde{u}\tilde{u}} \) and also \( \tilde{R}_{\tilde{x}\tilde{x}} \) be nonsingular, equivalently \( \ln |\tilde{R}_{\tilde{x}\tilde{x}}| < \infty \), reminiscent of the PWC’s integral form (see Appendix D). With \( N_x > N_y \), the PWC is never satisfied for white inputs like PAM – so for instance, PAM is never the best-performing design on an ISI channel with finite-length symbols.

5.1.1.2 Input Singularity Elimination

A second singularity-elimination step occurs for the input autocorrelation \( \tilde{R}_{xx} \). Input singularity may not relate to the channel null space, and may be intentional by design, or by accident through uniformed design. Figure 5.6 illustrates the process that this subsection subsequently describes.

The input covariance matrix \( \tilde{R}_{\tilde{x}\tilde{x}} \) has an eigendecomposition (step 1 in Fig 5.6)

\[
\tilde{R}_{\tilde{x}\tilde{x}} = \sum_{n=1}^{N_x} \tilde{E}_{x,n} \cdot \tilde{q}_n \cdot \tilde{q}_n^* = \begin{bmatrix} q_{\tilde{x}_H} & \tilde{q}_2 & \ldots & \tilde{q}_1 \end{bmatrix} \begin{bmatrix}
0 & \ldots & 0 & \ldots & 0 \\
0 & \ldots & 0 & \ldots & 0 \\
0 & \ldots & 0 & \ldots & 0 \\
0 & \ldots & 0 & \ldots & \tilde{E}_{x,N_x} \\
0 & \ldots & 0 & \ldots & 0 \\
0 & \ldots & 0 & \ldots & 0 \\
0 & \ldots & 0 & \ldots & 0 \\
0 & \ldots & 0 & \ldots & \tilde{q}_1^*
\end{bmatrix} \begin{bmatrix}
\tilde{q}_{\tilde{x}_H}^* \\
\vdots \\
\tilde{q}_1^*
\end{bmatrix},
\] (5.46)
where $\tilde{e}_{x,n}$ is the $n^{th}$ eigenvalue, and $\tilde{q}_n$ is the corresponding eigenvector. In effect, $\varrho_x$ becomes $N^*$ for the designed input; the input rank is equal to the number of used dimensions. Input-singularity removal omits the input vectors for which $\tilde{e}_{x,n} = 0$ from $\tilde{x}$'s (or $x$'s) construction because $\tilde{x}$ has no information-bearing component in these dimensions. Define $\tilde{Q} \triangleq [\tilde{q}_{N^*}, ..., \tilde{q}_1]$ and $P_{\tilde{Q}} \triangleq \tilde{Q} \cdot \tilde{Q}^*$ (step 2 in Fig. 5.6).

Figure 5.6: Algorithm for elimination of zero-energy $x$ dimensional components in the input autocorrelation $R_{xx}$.

---

14. Again Matlab often orders with largest in top left and any zeros at lower right; the permutation matrix $J_{N_x}$ reverses this.

15. Recalling from Chapter 2 that singular dimensions of a random vector contribute nothing to its entropy.
where $\tilde{A}$ and $\mathbf{u}'$ remain to be determined.

Without loss of generality, eigenvalue indexing satisfies $\mathcal{E}_{x,n} = 0$ for $n > N^*$. This second “input” projection uses the $N^* \leq \varrho_H$ column vectors of the $N_x \times N^*$ matrix $\tilde{A}_{\text{temp}}$

$$\tilde{A}_{\text{temp}} = P_Q \cdot \tilde{C} = \begin{bmatrix} \tilde{A} & \tilde{A}_1 \end{bmatrix} \quad \text{(step 3 in Fig. 5.6)},$$

Then

$$\tilde{A}_1 = \tilde{A} \cdot O_A \quad \text{(5.51)},$$

where $O_A = (\tilde{A}^* \cdot \tilde{A})^{-1} \cdot \tilde{A}^* \cdot \tilde{A}_1$ (step 4 in Fig. 5.6). Further,

$$\tilde{\mathbf{u}} = \begin{bmatrix} \tilde{\mathbf{u}}_2 \\ \tilde{\mathbf{u}}_1 \end{bmatrix} \quad \text{(step 5 in Fig. 5.6)},$$

and

$$\mathbf{u}' = \tilde{\mathbf{u}}_2 + O_A \cdot \tilde{\mathbf{u}}_1 \quad \text{(step 6 in Fig. 5.6)}.$$

When $N^* = \varrho_H$, then the matrix $\tilde{Q}$ is a square unitary matrix $(\tilde{Q} \cdot \tilde{Q}^* = \tilde{Q}^* \cdot \tilde{Q} = I)$ and then $\tilde{A} = \tilde{C}$.

The entropy of $\mathbf{x}'$ remains the same as $\tilde{x}$, $\mathcal{H}_{\mathbf{x}'} = \mathcal{H}_{\tilde{x}}$, because removing zeroed eigenvalues from the Gaussian autocorrelation matrix does not change entropy nor the matrix itself. Thus, the mutual information remains the same as the original channel, or

$$\mathcal{I}(\mathbf{x}'; \mathbf{y}) = \mathcal{I}(\tilde{x}; \mathbf{y}) = \mathcal{I}(\mathbf{x}; \mathbf{y}) \quad \text{(5.54)}.$$

The final input decomposition into its nonsingular component is

$$\mathbf{x}' = P^{*}_{\tilde{Q}} \cdot P^{*}_{\tilde{M}} \cdot \mathbf{x} = \tilde{A} \cdot \mathbf{u}' \quad \text{(step 7 in Fig. 5.6)}.$$

The new input autocorrelation matrix satisfies

$$R_{\mathbf{x}' \mathbf{x}'} = \tilde{A} \cdot R_{\mathbf{u}' \mathbf{u}'} \cdot \tilde{A}^* = \tilde{A} \cdot \left( \tilde{R}_{22} + O_A \cdot \tilde{R}_{12} + \tilde{R}_{21} \cdot O_A^* + O_A \cdot \tilde{R}_{11} \cdot O_A^* \right) \cdot A^* \quad \text{(step 8, Fig. 5.6)},$$

where $R_{\mathbf{u}' \mathbf{u}'}$ is necessarily nonsingular with rank $\varrho_{\mathbf{u}'} = N^*$ and of course the matrix entries $\tilde{R}_{11}$, $\tilde{R}_{12}$, $\tilde{R}_{21}$ and $\tilde{R}_{22}$ correspond to $\tilde{\mathbf{u}}$, and hence then also derive from $\mathbf{u}$. For every value that the random vector $\mathbf{x}'$ takes with nonzero probability, there is a unique $\mathbf{u}'$ value that corresponds to it, so these two are in one-to-one correspondence, and thus

$$\mathcal{I}(\mathbf{u}'; \mathbf{y}) = \mathcal{I}(\mathbf{x}'; \mathbf{y}) = \mathcal{I}(\mathbf{x}; \mathbf{y}) \quad \text{(5.57)}.$$

Essentially, $\mathbf{u}'$ generates $\mathbf{x}'$ and (5.57) is thus the only mutual information of interest on channel $\tilde{H}$.

The input’s two-stage reduction to $\mathbf{u}'$, is the finite-length equivalent to Chapter 3’s finding only that transmit band over which there is nonzero channel-output energy (or which satisfies Appendix D’s PWC) and Chapter 3’s focusing attention on this band only for the infinite-length MMSE-DFE. Since the mutual information is the same as the original system, and indeed also the same as the known optimum VC for the input $R_{\mathbf{x} \mathbf{x}}$, a nonsingular equivalent system now exists.

Appendix G’s program “fixin.m” removes the input singularity\footnote{This separation is often obvious, but a general procedure follows the use of the command “qr” in matlab. By executing the command “[Q,R,J]=qr(\tilde{P}_Q \cdot \tilde{C})”, then $N^*$ is obtained by the command “rank(\tilde{P}_Q \cdot \tilde{C})”, and then $\tilde{A}$ is the first $N^*$ columns of the matrix formed by $Q^* R^* J$ in matlab and the last $\varrho_H - N^*$ columns are then $A_2$. The rearrangement of inputs by $J$ needs to be noted and used by the designer also.}.

\footnote{Thanks again to 2021 Student Ethan Liang for this program.}
function [At, OA, Ruupp] = fixin(Ruutt, Ct, tol)

xt->xp, ut->up (Reducing ranking of ut->up, possible b/c Rxtt singlar)
Input: Ruutt, Ct, tol
Ruutt: autocorrelation matrix of utilde
Ct: "nullspace of H"-adjusted discrete modulator matrix
tol: used in licols to determine rank of matrix Ctemp (default is 1e-10)
Output: At, OA, Ruupp
At: "nullspace of H & Rxx singularity"-adjusted discrete modulator matrix
OA: projection matrix of A2 onto A
Ruupp: autocorrelation matrix of uprime

[reducing a singular input for the 1 + D channel]  This example returns to Example 5.1.1's 1 + D AWGN channel with N + ν = 3 (N_x = 4): This channel's input originally had \( \varrho_x = 3 \) with \( R_{xx} = I \), the elimination of channel-null-space input components reduced the input to \( \varrho_u = 2 \); therefore there are no singular input components remaining.

\[
R_{xx} = \begin{bmatrix}
5/6 & -1/3 & 5/6 \\
-1/3 & 4/3 & -1/3 \\
5/6 & -1/3 & 5/6
\end{bmatrix},
\]  (5.58)

Thus, instead the example proceeds with the alternative input autocorrelation matrix and modulator:

\[
R_{xx} = \begin{bmatrix}
5/6 & -1/3 & 5/6 \\
-1/3 & 4/3 & -1/3 \\
5/6 & -1/3 & 5/6
\end{bmatrix},
\]  (5.59)

which forms from some \( \mathbf{u} \) as \( \mathbf{x} = \mathbf{C} \cdot \mathbf{u} \), where the input \( \mathbf{u} \) has 2 x 2 autocorrelation matrix

\[
R_{uu} = \begin{bmatrix}
1 & 0 & 1 \\
0 & 0 & 0 \\
1 & 0 & 2
\end{bmatrix},
\]  (5.59)
and\(^\text{18}\)

\[
C = \begin{bmatrix}
\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{3}} \\
\frac{1}{\sqrt{3}} & 0 & \frac{1}{\sqrt{3}} \\
\frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{3}}
\end{bmatrix}.
\]

(5.60)

The following matlab commands eliminate the channel singularity and then also eliminate channel input singularity for the above example.

\[
\begin{align*}
Rxx &= \begin{bmatrix}
0.8333 & -0.3333 & 0.8333 \\
-0.3333 & 1.3333 & -0.3333 \\
0.8333 & -0.3333 & 0.8333
\end{bmatrix}; \\
Ru = &[1 0 1 \\
0 0 0 \\
1 0 2]; \\
C &= \begin{bmatrix}
0.4082 & -0.7071 & -0.5774 \\
0.8165 & 0.0000 & 0.5774 \\
0.4082 & 0.7071 & -0.5774
\end{bmatrix}; \\
\begin{bmatrix} Ct, Ot, Ru \end{bmatrix} &= \text{fixmod}(H, Ruu, M) \\
Ct &= \begin{bmatrix}
0.4082 & -0.7071 \\
0.8165 & 0.0000 \\
0.4082 & 0.7071
\end{bmatrix}; \\
Ot &= 1.0e-15 * (Ot \text{ is zero}) \\
&= 0.1473 \\
&= 0.0196; \\
Ru &= \begin{bmatrix}
1.0000 & 0.0000 \\
0.0000 & 0.0000
\end{bmatrix};
\end{align*}
\]

\[
\begin{align*}
\begin{bmatrix} At, OA, Ru \end{bmatrix} &= \text{fixin}(Ru, Ct) \\
At &= \begin{bmatrix}
0.4082 \\
0.8165 \\
0.4082
\end{bmatrix}; \\
OA &= 9.9148e-16 \text{ (zero)}; \\
Ru &= 1.0000
\end{align*}
\]

First from \(R_{uu}\), it is clear that input dimension 2 has zero energy so the \(\tilde{C}\) is for inputs \(u_1\) and \(u_3\) (which were the linearly independent columns that were maintained):

\[
\begin{align*}
\tilde{x}_3 &= \frac{u_3}{\sqrt{6}} - \frac{u_1}{\sqrt{2}} \\
\tilde{x}_2 &= \frac{\sqrt{2} \cdot u_3}{\sqrt{3}} \\
\tilde{x}_1 &= \frac{u_3}{\sqrt{6}} + \frac{u_1}{\sqrt{2}}.
\end{align*}
\]

(5.61) \quad (5.62) \quad (5.63)

The input autocorrelation matrix \(R_{xx}\) is singular but the zero energy on one of the pass-space

\(^{18}\)This \(C\) was found by zeroing one of the two pass-space energies and putting its energy in the null space. There are many other ways to construct a modulator and \(R_{uu}\) to start this example; this is just one way.
dimensions is addressed above by the fixin.m command leaving

\[
x_3' = \frac{u_3}{\sqrt{6}} \tag{5.64}
\]

\[
x_2' = \frac{\sqrt{2} \cdot u_3}{\sqrt{3}} \tag{5.65}
\]

\[
x_1' = \frac{u_3}{\sqrt{6}} , \tag{5.66}
\]

which is a one-dimensional input after removal of all singularity.

Finally, the overall channel is

\[
\begin{align*}
\text{>> } & H=[1 \ 1 \ 0 \\
& 0 \ 1 \ 1]; \\
\text{>> } & H*At = \\
& \begin{bmatrix} 1.2247 \\
& 1.2247 \end{bmatrix} \\
\text{>> } & At'*H'*H*At = 3.0000
\end{align*}
\]

so the matched filter output is 3 times the input \( u_3 \).

### 5.1.2 Finding \( A \), the GDFE’s Discrete Modulator

Given an input \( u' \) after Subsection 5.1.1’s two-step singularity, this subsection and all ensuing chapters and sections rename this input simply to \( u \) to avoid the use of primes and tildes. (The primes and tildes are necessary in singularity elimination for notational bookkeeping purposes, but need no longer continue.)

#### Canonical Forward Channel:

Again, the GDFE development generalizes to Gaussian noise vectors \( n \) that do not necessarily have a diagonal autocorrelation matrix through

\[
R_{nn} = R_{nn}^{1/2} \cdot R_{nn}^{*1/2} , \tag{5.67}
\]

where \( R_{nn}^{1/2} \) is any (matrix square roots\(^\text{19}\) are not usually unique) square root of \( R_{nn} \). The equivalent “white-noise” channel \( H \) forms through noise whitening, \( H = R_{nn}^{1/2} \cdot H \). An additional step is possible because \( R_{uu} \) is now nonsingular, through any \( R_{uu} \) square root \( \Phi \) such that

\[
R_{uu} = \Phi \cdot \Phi^* , \tag{5.68}
\]

where the input \( u \) now has relationship

\[ u = \Phi \cdot v \tag{5.69} \]

with some white input \( v \) where \( R_{vv} = I \). Then the channel becomes

\[
\bar{y} = R_{nn}^{-1/2} \cdot y = R_{nn}^{-1/2} \cdot (H \cdot \bar{A} \cdot \Phi \cdot v + \bar{n}) = \bar{H} \cdot v + \bar{n} , \tag{5.70}
\]

where \( \bar{H} = R_{nn}^{-1/2} \cdot H \cdot \bar{A} \cdot \Phi \), and \( \bar{n} \) is white noise with unit variance per (possibly complex) dimension. Figure 5.4 illustrates this noise whitening, and the consequent noise-whitened channel matrix \( \bar{H} \) can undergo all singularity elimination (channel and input).

---
\(^{19}\) The matlab command sqrtm finds a symmetric-matrix square root such that \( R^{1/2} \cdot R^{1/2} = R \). The matlab chol command finds a triangular square root such that \( R^{1/2} \cdot R^{*1/2} = R \).
**Definition 5.1.1** [The GDFE discrete modulator] A GDFE discrete modulator satisfies

\[ A = \tilde{A} \cdot \Phi, \tag{5.71} \]

where \( \tilde{A} \) is any set of linearly independent columns of any valid symmetric decomposition of \( R_{\tilde{x}\tilde{x}} = \tilde{A} \cdot R_{\tilde{u}\tilde{u}} \tilde{A} \), and \( \Phi \) is any valid decomposition of the resultant \( R_{\tilde{u}\tilde{u}} = \Phi \cdot \Phi^*. \)

The input to the GDFE discrete modulator is where designers place codeword subsymbols for a good AWGN code. The filtered noise \( \tilde{n} \) remains independent of the input \( u \). By the sufficiency of matched filtering\(^{20}\), the receiver may process the received signal by a set of matched-filter vectors (the rows of \( \tilde{H}^* \)) without information loss to obtain Appendix D’s MMSE canonical forward channel model:

\[ z = \tilde{H}^* \cdot \tilde{H} \cdot v + \tilde{H}^* \cdot \tilde{n} = R_f \cdot v + n', \tag{5.72} \]

where \( R_f \) is the canonical forward-channel matrix \( R_f = \tilde{H}^* \cdot \tilde{H} \) and \( R_{n'v'} = R_f \). This forward channel is canonical (as per Appendix D) because the channel shaping \( R_f \) is the same as the noise autocorrelation or shaping, also \( R_f \). Because the noise vector \( n \) is uncorrelated with the input vector \( u \), a right triangle can be drawn to illustrate Equation (5.72) as in Figure 5.7 (on the left). This triangle satisfies a “Pythagorean Theorem”

\[ R_{zz} = R_f \cdot R_{vv} \cdot R_f + R_{n'n'} = R_f^2 + R_f, \tag{5.73} \]

observing that \( R_f^* = R_f \). Also, because canonical channel noise \( n' \) is uncorrelated with the channel input \( u \), then (via Appendix D’s Orthogonality Principle)

\[ \hat{z} = R_f \cdot v, \tag{5.74} \]

is the MMSE vector estimate of \( z \) given \( v \). Thus, another expression for \( R_f \) directly invoking Appendix D’s Orthogonality Principle, is

\[ R_f = R_{zv} \cdot R_{zz}^{-1} = R_{zv} \tag{5.75} \]

The MMSE matrix for this forward-channel estimation problem is then clearly \( R_{n'n'} = R_f \). (This also implies that \( R_{zv} = R_{vz} \), or \( R_{zv} \) is conjugate symmetric.)

**Canonical Backward Channel:** Good design desires a canonical MMSE estimate of \( v \) from \( z \), where \( u = \Phi \cdot v \); this MMSE estimate arises through the the canonical backward channel model:

\[ v = R_b \cdot z + e, \tag{5.76} \]

where \( R_b \) is the MMSE matrix estimator (equalizer)

\[ R_b = R_{vz} \cdot R_{zz}^{-1}. \tag{5.77} \]

The vector \( e \) is the MMSE error vector for the MMSE estimate of \( v \) given \( z \). This backward channel is a matrix linear equalizer. The dual Pythagorean relationship on Figure 5.7’s right is

\[ R_{ee} = R_{vv} - R_{vz} \cdot R_{zz}^{-1} \cdot R_{zv} = I - R_b \cdot R_{zz} \cdot R_b. \tag{5.78} \]

A related error vector is the MMSE estimate of \( u \) or \( e' \triangleq \Phi \cdot e \) by linearity of MMSE and

\[ R_{ee'} = \Phi \cdot R_{ee} \cdot \Phi^*. \tag{5.79} \]

\(^{20}\)This matched-filter matrix is a one-to-one mapping between the signal components of \( x' \) and \( z \). The noise after filtering is white and so any components on other dimensions are irrelevant.
A very useful alternative expression for $R_b$ is (recalling that $R_{uz}$ is square nonsingular because of the earlier singularity removal).

\[
R_b = R_{uz} \cdot R_{zz}^{-1} \hspace{1cm} (5.80)
\]

\[
R_b^{-1} = R_{zz} \cdot R_{zu}^{-1} \hspace{1cm} (5.81)
\]

\[
R_b^{-1} \cdot R_{vv} = R_{zz} \cdot R_{zu}^{-1} \cdot R_{vv} \hspace{1cm} (5.82)
\]

\[
R_b^{-1} = R_{zz} \cdot R_f^{-1} \hspace{1cm} (5.83)
\]

\[
R_b^{-1} = (R_f^2 + R_f) \cdot R_f^{-1} \hspace{1cm} (5.84)
\]

\[
R_b^{-1} = R_f + I \hspace{1cm} (5.85)
\]

which allows computation of $R_b$ from the forward channel $R_f = \tilde{H}^* \cdot \tilde{H}$ and the given transmit autocorrelation matrix $R_{vv} = I$. Equivalently,

\[
I = R_b \cdot R_f + R_b = R_f \cdot R_b + \cdot R_b \hspace{1cm} (5.86)
\]

\[
R_b = I - R_f \cdot R_b \hspace{1cm} (5.87)
\]

\[
= I - R_b \cdot R_f \hspace{1cm} (5.88)
\]

or equivalently $R_b$ and $R_f$ commute, since $I$ is symmetric. These MMSE-based $R_b$ and $R_f$ are the same as Chapter 2’s versions for the MAC and BC user inputs because those were also the MMSE estimators.

**Canonical Triangles:** Using this convenient result (5.88) in Equation 5.78

\[
R_{ee} = R_{vv} - R_{uz} \cdot R_{zz}^{-1} \cdot R_{zu} = I - R_{uz} \cdot R_{zz}^{-1} \cdot R_{zu} \hspace{1cm} (5.89)
\]

\[
= I - R_{vv}^{-1} \cdot R_{uz} \cdot R_{zz}^{-1} \cdot R_{zu} \hspace{1cm} (5.90)
\]

\[
= I - R_b \cdot R_f \hspace{1cm} (5.91)
\]

\[
= R_b (\text{uses Equation } (5.88)) \hspace{1cm} (5.92)
\]

\[
| R_{ee} | = | I - R_b \cdot R_f | = | R_{ee} | \hspace{1cm} (5.93)
\]
an inverse signal-to-MMSE ratio. The use of the two triangles allows derivation of any backward relation from any forward relation by swapping $R_f$ and $R_b$ as well as replacing $R_{ee}$ by $R_{n'n'}$ (essentially then $z \leftrightarrow u$, $n' \rightarrow e$). The following relations use Figure 5.7’s forward and backward channels’ duality: \(\text{(with } R_{vv} = I \text{ ONLY)}\)

\[
R_b^{-1} = R_f + R_{vv}^{-1} = R_f + I \quad (5.94)
\]

\[
R_f^{-1} = R_b + R_{zz}^{-1} \quad (5.95)
\]

\[
R_{ee} = R_b \quad (5.96)
\]

\[
R_{n'n'} = R_f \quad (5.97)
\]

\[
R_{zz} = R_f \cdot R_{vv} \cdot R_f + R_f = R_f \cdot R_b^{-1} \quad (5.98)
\]

\[
R_{vv} = R_b \cdot R_{zz} \cdot R_b + R_b = R_b \cdot R_{zz} \cdot R_f^{-1} \quad (5.99)
\]

\[
R_{zv} = R_f \cdot R_{vv} = R_f \quad (5.100)
\]

\[
R_{vz} = R_b \cdot R_{zz} \quad (5.101)
\]

Because the signal part of $z$ is in one-to-one correspondence with $v$ as a result of the careful elimination of unnecessary dimensionality through the matrix $A$,

\[
I(v; z) = I(v; y) = I(x; y) \quad (5.102)
\]

so that the new channel between $v$ and $z$ carries all the pertinent information between channel input and output. The backward and forward canonical channels both have the same mutual information. Transmission from $v$ to $z$ has the same performance as $z$ to $v$, and as from $x$ to $y$. However, the backward channel next leads to a canonical and simpler receiver for $b \leq I(x; y)$, the GDFE.

### 5.1.3 Generalized Decision Feedback Development

The GDFE exploits the backward canonical model’s structure. The backward canonical model is

\[
v = R_b \cdot z + e \quad (5.103)
\]

where $R_b$ is now nonsingular. The inverse MMSE $R_b^{-1}$ has an lower-diagonal-upper Cholesky factorization\(^{\text{21}}\)

\[
R_b^{-1} = G^* \cdot S_0 \cdot G \quad (5.104)
\]

where $G$ is upper triangular and monic (ones along the diagonal), and $S_0$ is a diagonal matrix of positive Cholesky factors. Then factoring $R_{ee} = R_b$:

\[
R_b = G^{-1} \cdot S_0^{-1} \cdot G^{-*} \quad (5.105)
\]

Premultiplication of $v$ in (5.103) by $G$ produces

\[
v' = G \cdot v = S_0^{-1} \cdot G^{-*} \cdot z + e' = z' + e' \quad (5.106)
\]

where $e' = G \cdot e$ has diagonal autocorrelation matrix $S_0$, and $z'$ is the matrix feedforward-filter output $S_0^{-1} \cdot G^{-*} \cdot z$. Again, mutual information is maintained because the triangular matrices $G$ and $S_0$ are one-to-one, so

\[
I(v; z') = I(x; y) \quad (5.107)
\]

**Back Substitution:** The receiver recovers the channel input vector $v$ from

\[
z' = G \cdot v - e' \quad (5.108)
\]

\(^{\text{21}}\)See Appendix D: Cholesky Factorization of an autocorrelation matrix $R_{xx}$ is the matrix or finite-length-packet equivalent of spectral factorization of the infinite-length autocorrelation function $R_x(D)$. 

820
by a process known as back substitution, as also used by Chapter 2’s MAC: A detailed matrix description
of (5.108) is
\[
\begin{bmatrix}
\vdots \\
\vdots \\
\vdots \\
z'_0
\end{bmatrix}
= \begin{bmatrix}
1 & g_{N^*-1,N^*-2} & \ldots & g_{N^*-1,0} \\
0 & 1 & \ldots & g_{N^*-2,0} \\
\vdots & \vdots & \ddots & \vdots \\
0 & \ldots & 0 & 1
\end{bmatrix}
\begin{bmatrix}
\hat{v}_{N^*-1} \\
\hat{v}_{N^*-2} \\
\vdots \\
\hat{v}_0
\end{bmatrix}
- \begin{bmatrix}
0 \\
0 \\
\vdots \\
\hat{e}_0
\end{bmatrix},
\]
(5.109)
where each symbol’s first time dimension is at the bottom, following Chapters 2 and 4. By starting at
the bottom of (5.109) and ascending, back-substitution solution of (5.109) obtains the decision-feedback
equations:
\[
\hat{v}_0 = \text{dec}_0(z'_0)
\]
(5.110)
\[
\hat{v}_1 = \text{dec}_1(z'_1 - g_{1,0} \cdot \hat{v}_0)
\]
(5.111)
\[
\hat{v}_2 = \text{dec}_2(z'_2 - g_{2,1} \cdot \hat{v}_1 - g_{2,0} \cdot \hat{v}_0)
\]
(5.112)
\[
\vdots = \vdots
\]
(5.113)
\[
\hat{v}_{N^*-1} = \text{dec}_{N^*-1}(z'_{N^*-1} - \sum_{i=0}^{N^*-2} g_{N^*-1,i} \cdot \hat{v}_i)
\]
(5.114)
The back-substitution solution is not a maximum-likelihood detector: but because \(R_{vv} = I\) is diagonal,
it can reliably detect outputs at \(b = I(x;y)\) as long as error propagation for an actual systems dimen-
sional detectors is negligible. With \(\Gamma = 0\) dB gap, it may be that many (infinite) successive symbols for
dimension \(n = 0\) must decode (“dec”) first before any of the results can be used for dimension \(n = 1\),
and so on. This can introduce several subsymbol delays in decoding. In practice, Chapter 4’s separation
theorem with the same code and constellation on all dimensions reduces this delay.

ComputeGDFE Program: Appendix G’s computeGDFE.m program can be used, but computes a
SNR according to the size of the input square-root matrix \(A\), unless a last optional input appears that
is an alternative symbol size.

```
function [snrGDFEu, GU, WU, S0, MSWMFU, b, bbar, snrGLEu] = computeGDFE(H, A, cb, Lx)
```

Inputs
H: noise-whitened channel, Ly x Lx (one tone)
A: any Lx x Lx square root of input autocorrelation matrix
   This can be generalized non-square square-root
cb: =1 if \(H\) is complex baseband; =2 if \(H\) is real baseband
Lx: optional input of Lx when not equal to right-size of A
   this is used to compute bits/dimension properly

Outputs
G: feedback matrix
GU: unbiased feedback matrix
W: feedfoward linear equalizer
WU: unbiased feedforward linear equalizer
S0: sub-channel channel gains
MSWMFU: unbiased mean-squared whitened matched filter
b: bit distribution vector
bbar: number of bits/dimension
snrGDFEu - unbiased SNR in dB; assumes size of R_sqrt input

---

22This should not be confused with the singularity-elimination process that simply described blocks of vectors within a symbol
the user should recompute SNR if there is a cyclic prefix
b = bit distribution over symbol dimensions (real cb=2; complex cb=1)
bbar is sum(b)/Lx so total bits/(real cb=2; cplx cb=1) dimension
snrGLEu is the linear GLE SNR
RecU is the entire unbiased receiver before feedback section

Note: The R_sqrt need not be square non-singular, as long as
R_sqrt*R_sqrt’ = input autocorrelation matrix Rxx.

Thanks to Ethan Liang, corrected/updated J. Cioffi

--------------------------------------------------------------------------
See upcoming Example 5.1.2 for example use. The last optional argument snrGLEu is for a MMSE LE
(so no feedback allowed) best SNR. It is never larger than the GDFE SNR and usually smaller.

Suboptimal Detection:  The “decn” operation can be simple symbol-by-symbol “uncoded” detection
for the constellation Cn on the nth transmitted subsymbol un,n = 0, ..., N∗−1 (in which case Γ > 0 dB).
Any error made likely causes other subsequent decn errors, but the error propagation arrests when n
attains N∗ (unlike Chapter 3’s MMSE-DFE potential infinite-length error bursts). The feedback “filters”
are a function of the dimensional index n and represent a periodic dimension-variable feedback section.
The period is the symbol period. Similarly, the feedforward section is also periodically dimension-varying
because it is a triangular matrix multiply for which the rows are not simple shifts of each other. This
receiver is called the “Generalized DFE” or GDFE, and a GDFE block diagram appears in Figure 5.8.
The nth tone’s MSE is
\[ E[|e'_n|^2] = S_{0,n}^{-1} . \]  (5.115)

The GDFE creates a set of parallel AWGNs with (biased) SNR’s equal to
\[ SNR_{bias,n} = 1 + \frac{E[|v_n|^2]}{E[|e'_n|^2]} = 1 \cdot S_{0,n} \]  (5.116)
where this expression is similar to Chapter 3’s expression
\[ SNR_{MMSE-DFE} = \gamma_0 \cdot \frac{\|h\|^2}{N_0} \]
which derives from Chapter 3’s canonical factorization when the channel, noise, and input processes are
stationary. The product of these biased SNR’s is then
\[ SNR_{GDFE} = \left( \prod_{n=1}^{N^*} SNR_{bias,n} \right)^\frac{1}{N^*} = \frac{|R_{vv}|^{1/N_x}}{|R_{ee'}|^{1/N_x}} = \frac{|R_{ee'}|^{-\frac{1}{N_x}}}{2^{2\frac{\gamma}{\gamma_0}}} = 2^{2\frac{\gamma}{\gamma_0}}(x') \]
(5.117) is a finite-length CDEF result (see Chapter 3 for the infinite-length version). A suboptimal
detector, namely the GDFE can exhibit an SNR on an equivalent set of independent AWGN channels
that is equal to the best achievable for such a set for independent input-dimensional components. This
surprising result occurs only for the backward canonical channel model. For this symbol-length, the
same good codes that achieve capacity on the ISI-free channel can also do so reliably with the GDFE
system; further the same-gap codes can nearly achieve the same gap-reduced data rate with the GDFE.
If R_x'x' is the same as that generated by water-filling on the N* discrete subchannels of H that vector
coding would use, then the GDFE’s reliably decoded data rate approaches C, just as closely as with the
Chapter 4’s VC, even though the detector is suboptimum. Only a “white input” (diagonal R_{vv} = I)
attains this canonical performance level with the suboptimal detector. Furthermore, following Chapter
2’s arguments on shaping gain, GDFE use on good codes built from square constellations (like PAM or
QAM subsymbols) so that Γ → Γ − γ_s = Γ − 1.53 dB will retain their goodness and just lose the 1.53
dB of shaping that is possible with best codes.

---

23Or more generally, periodically “index-varying.” The period is the symbol period.
**For Matrix ISI Channel Models \( \tilde{H} \):** The GDFE replaces Chapter 3’s MMSE-DFE’s linear filters by matrix multiplies, indeed triangular matrix multiplies \((I - G)\) and \(S_0 \cdot G^{-*}\) correspond to a causal feedback section and a non-causal feed-forward section, respectively, when \(H\) is Toeplitz (stationary channel). The matched filter is the matrix \(R_{nn}^{-1/2} \cdot H \cdot A^*\). These matrix multiplies can be interpreted as periodically dimensionally varying filters - a different set of coefficients for each index \(n = 1, ..., N^*\) within the packet. The GDFE filters are thus fundamentally different from Chapter 3’s finite-length filters in general.

The GDFE’s diagonal-matrix SNR is

\[
\text{SNR}_{\text{GDFE}} \triangleq R_{vv} \cdot R_{e^*e}^{-1} = R_{e^*e}^* = \text{SNR}_{\text{GDFE, unb}} + I ,
\]

and its determinant is \(\text{SNR}_{\text{GDFE}} = |\text{SNR}_{\text{GDFE}}|\). Thus

\[
|\text{SNR}_{\text{GDFE}}| = |\text{SNR}_{\text{GDFE, unb}} + I| = \prod_{i=0}^{N^*-1} (1 + \text{SNR}_{U,i}) .
\]

Figure 5.8 illustrates the system.

---

**Bias:** The GDFE has bias like all MMSE estimators, as per Appendix D. Each GDFE output dimension has a MMSE estimate with \(S_0,n = \text{SNR}_n + 1\). The matrix \(\text{SNR}_{\text{GDFE}}\) corresponds to a biased estimate. That is

\[
\mathbb{E}[\hat{v}_n/v_n] = \left(\frac{S_0,n - 1}{S_0,n}\right) \cdot v_n \neq v_n .
\]
Bias can be removed individually after feedback on the subchannel dimensions or by scaling each feedforward-matrix output dimension by $\frac{S_{0,n}}{S_{0,n-1}}$. Also

$$G_{unb} = I + S_0 \cdot (S_0 - I)^{-1} \cdot [G - I] \quad \text{(5.121)}$$
$$W_{unb} = (S_0 - I)^{-1} \cdot G^{*-} \quad \text{(5.122)}$$
$$\text{MSWMF}_{unb} = (S_0 - I)^{-1} \cdot G^{*-} \cdot A^* \cdot H^* \cdot R_{nn}^{-1/2} \quad \text{(5.123)}$$

**Reconstructing $\hat{u}$:** After the GDFE estimates $\hat{v}$, then the corresponding MMSE estimate of $u$ is $\hat{u} = \Phi \cdot \hat{v}$, and in turn $\hat{x}' = A \cdot \hat{v}$, because MMSE estimates are linear. Any part of $x$ not contained in $x'$ is lost in transmission. When $x = x'$, there is no such loss and $u$ and $v$ are in 1-to-1 correspondence (as long as the GDFE symbol-by-symbol detectors function without significant error in estimation). Thus $A = A \cdot \Phi$ is a canonical matrix transmit filter or discrete modulator. Chapter 4’s separation theorem still applies if $\Gamma = 0$ dB good codes use a (large) constellation of constant size, and loading simplifies to determining constellation size and code rate (with no error propagation since $\Gamma = 0$ dB).

Section 5.3 investigates two specific structures for the matrix $\Phi$ and finds that a triangular factorization can lead to a MMSE-DFE finite-length equivalent. The other structure, an eigen-decomposition, leads to VC. There can be many such transformations and each corresponds to a different GDFE, but all attain the same performance. The following example illustrates one such case:

**EXAMPLE 5.1.2 [GDFE for 1 + .9D^{-1} Channel]** This example revisits Chapters 3’s and 4’s 1 + .9 · D^{-1} channel with input energy $\bar{E}_x = 1$ ($R_{xx} = I = R_{uu}$) and $\sigma^2 = .181$. This current-example instance investigates a GDFE for this channel with $N = 2$ and $\nu = 1$, so $N_x = 3$. The matrix $R_{nn}^{-1/2} \cdot H$ for this channel is

$$\frac{1}{\sigma} \cdot H = \frac{1}{\sqrt{.181}} \cdot \begin{bmatrix} .9 & 1 & 0 \\ 0 & .9 & 1 \end{bmatrix} = \begin{bmatrix} 2.1155 & 2.3505 & 0 \\ 0 & 2.1155 & 2.3505 \end{bmatrix} \quad \text{(5.124)}$$

with singular value decomposition:

$$\frac{1}{\sqrt{2}} \cdot \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 3.8694 & 0 & 0 \\ 0 & 2.2422 & 0 \end{bmatrix} \begin{bmatrix} .3866 & .6671 & .6368 \\ .8161 & .0741 & -.5731 \\ .4295 & -.7412 & .5158 \end{bmatrix}^* \quad \text{(5.125)}$$

The first two columns of $M$ span $\tilde{H}$’s pass space $\mathcal{P}_{\tilde{H}}$. The original PAM modulator corresponds to $C = I$, so

$$\tilde{C}_{\text{temp}} = P_{\tilde{M}} \cdot I = \begin{bmatrix} .5945 & .3649 & -.3285 \\ .3649 & .6715 & .2956 \\ -.3285 & .2956 & .7340 \end{bmatrix} \quad \text{(5.126)}$$

Then $\tilde{C}$ is the leftmost 2 columns of $\tilde{C}_{\text{temp}}$. This special case of white PAM modulation on $x$ corresponds to a rank-2

$$R_{\tilde{x}x} = P_{\tilde{M}} \cdot I \cdot P_{\tilde{M}} \quad \text{(5.127)}$$

Since $\sigma_x = 2$, then $\bar{E}_x = \bar{E}_H$ and no further singularity elimination is necessary. However, $R_{uu}$ is not diagonal. By using the steps in Figure 5.4,

$$R_{uu} = \begin{bmatrix} 2.5242 & -1.3717 \\ -1.3717 & 2.2346 \end{bmatrix} \quad \text{(5.128)}$$

and

$$\hat{u}_1 = x_1 - 1.235 \cdot x_3 \quad \text{(5.129)}$$
$$\hat{u}_2 = x_2 + 1.111 \cdot x_3 \quad \text{(5.130)}$$

**In matlab, the commands are:**

**EIIMINATE CHANNEL SINGULARITY:**

824
The $R_{uu}$ is not white. To find a white input, the designer can use several square root forms for the $2 \times 2$ matrix $R_{uu}$. This $R_{uu}$ is already nonsingular, the fixin.m program essentially does not further improve, but we show it anyway here:

$$
\begin{bmatrix}
0.5945 & 0.3649 \\
0.3649 & 0.6715 \\
-0.3285 & 0.2956
\end{bmatrix}
$$

$$
\begin{bmatrix}
-1.2346 \\
1.1111
\end{bmatrix}
$$

Specifically, $R_{xx}^\prime R_{xx}^\prime = R_{uu}$ and $\hat{C} = A$ in this situation. Design can proceed with a choice of square root for $R_{uu}$, so a first example is Cholesky Factorization of $R_{uu}$

$$
Gxbar = lohc(Ruupp);
$$

$$
Gx = Gxbar * inv(diag(diag(Gxbar)));
$$

$$
Xmit = A * Gxbar;
$$

$$
[snrGDFEu, GU, WU, S0, MSWMFU, b, bbar] = computeGDFE(H, Xmit, 2, 3);
$$

$$
\begin{bmatrix}
1.0000 & 0.5731 \\
0 & 1.0000
\end{bmatrix}
$$

$$
\begin{bmatrix}
0.1328 & 0 \\
-0.0492 & 0.0972
\end{bmatrix}
$$

$$
\begin{bmatrix}
8.5275 & 0 \\
0 & 11.2899
\end{bmatrix}
$$

$$
\begin{bmatrix}
0.3645 & 0.0000 \\
0.0179 & 0.3073
\end{bmatrix}
$$

$$
\begin{bmatrix}
1.5461 \\
1.7485
\end{bmatrix}
$$

$$
1.0982 \text{ bits/dimension (real for this example because } cb = 2)$$
The SNR is invariant at 5.5427 dB.
Instead, a symmetric square root might be of interest:

```
>> Gxbar=sqrtm(Ruutt)
>> Gxbar =
   1.5186  -0.4668
  -0.4668   1.4201
>> Xmit=A*Gxbar;
>> [snrGDFEu, GU, WU, S0, MSWMFU, b, bbar] = computeGDFE(H, Xmit,2,3);
>> snrGDFEu = 5.5427 dB
```

```
>> GU =
    1.0000  0.3680
    0     1.0000
>> MSWMFU =
    0.3881 -0.1812
  -0.1215   0.2378
>> b' =  1.3447  1.9499
>> bbar =  1.0982
```

The performance is the same, as is the total bits/symbol, but the bit distribution and the implementation are different. Yet another version uses the eigendecomposition of $R_{\tilde{u}\tilde{u}}$ directly, starting with the $A$ from the fixin program’s outputs $A$ and $Ruupp$.

```
>> [V,D]=eig(Ruutt);
>> Gxbar=V*sqrtm(D) =
  -0.6690  -1.4411
 -0.7433   1.2970
>> Xmit=A*Gxbar;
>> [snrGDFEu, GU, WU, S0, MSWMFU, b, bbar] = computeGDFE(H, Xmit,2,3);
>> snrGDFEu = 5.5427 dB
```

```
>> GU =
    1.0000  -0.3459
    0     1.0000
>> MSWMFU =
  -0.2535  -0.1261
 -0.1648   0.3645
>> b' =  1.8760  1.4186
>> bbar =  1.0982
```

A GDFE based on estimating $\tilde{u}_1$ and $\tilde{u}_2$ will therefore not achieve canonical performance because $R_{\tilde{u}\tilde{u}} \neq I$. However, any of the above $R_{vv} = I$ choices generates a valid GDFE with the same performance.

A GDFE based on the 3-dimensional input $R_{\tilde{x}\tilde{x}}$ also achieves canonical performance over 3 dimensions, but with same performance:

```
>> [Ct, Ot, Ruutt] = fixmod(H, eye(3), eye(3));
>> [A, OA, Ruupp] = fixin(Ruutt, Ct);
>> Rxxtt=A*Ruutt*A';
>> Gxbar=sqrtm(Rxxtt) =
```
Finally a white input or the input $R_{\tilde{x} \tilde{x}}$ wastes energy (1 unit of 3)

$$R_{\tilde{x} \tilde{x}} = \begin{bmatrix} 0.8918 & 0.5474 & -0.4927 \\ 0.5474 & 1.0073 & 0.4434 \\ -0.4927 & 0.4434 & 1.1009 \end{bmatrix}.$$  \hspace{1cm} (5.131)

and

$$\text{trace}(R_{\tilde{x} \tilde{x}}) = 3.$$  \hspace{1cm} (5.132)

A valid decomposition would use

$$R_{\tilde{x} \tilde{x}} = \Phi \cdot \Phi^*,$$  \hspace{1cm} (5.133)

or $\Phi \to \sqrt{1.5} \cdot \Phi$.

The corresponding GDFE now is:

>> [snrGDFEu, GU, WU, S0, MSWMFU, b, bbar] = computeGDFE(H, (1.5)*Rxxtt,2,3)

snrGDFEu = 8.1496 dB
>> GU =
    1.0000  1.1111  0.0000
       0       1.0000  0.9067
       0       0       1.0000
>> MSWMFU =
     0.4727  0.0000
     0.0783  0.3857
    -0.1923  0.4254
>> b' =  1.2264  1.3485  0.7196
>> bbar =   1.0982

or if the $R_{\tilde{x} \tilde{x}}$ energy is reset to the original 3 units, but now in the channel pass space.

>> [snrGDFEu, GU, WU, S0, MSWMFU, b, bbar] = computeGDFE(H, eye(3),2,3)

snrGDFEu = 5.5427 dB
>> GU =
    1.0000  1.1111  0.0000
       0       1.0000  0.9067
       0       0       1.0000
>> MSWMFU =
     0.4727  0.0000
     0.0783  0.3857
    -0.1923  0.4254
>> b' =  1.2264  1.3485  0.7196
>> bbar =   1.0982
>> trace(Rxxtt) =  2.0000 <  3
The SNR increases to 8.1dB and the feedforward filter and the combination of feedforward-matched filter complete the matlab commands above. The complexity remains at 1.667/dimension for the minimum 2-dimensional GDFEs above (it increases unnecessarily for the 3-dimensional versions). From Chapter 3, the finite-length MMSE-DFE required \( N_f = 2 \) and \( N_b = 1 \) to get this same performance, leading to a complexity of 3, almost twice that of the GDFE with altered input. Such is often the benefit of transmit optimization – not only does it work better, the receiver simplifies. The two-dimensional energy choice choice approximates water-filling. As \( N \) becomes large, the ultimate SNR is 8.4 dB for an i.i.d. input. Unfortunately, as \( N \) grows, the complexity for this GDFE choice will grow as \( N^2 \), which is unnecessarily well above the complexity of DMT (which achieves 8.8 dB when \( N = 16 \) and complexity is 3.8/dimension). If the design uses a water-fill input, then the large \( N \) GDFE SNR also approaches 8.8 dB.

Chain rule relationships and order: Chapter 2’s mutual-information chain rule

\[
I(x \ y) = I(x_0; y) + I(x_1 \ y/x_0) + \ldots + I(x_n \ y/[x_1 \ldots x_{n-1}]) + \ldots + I(x_{N-1} \ y/[x_1 \ldots x_{N-2}])
\]

relates directly to all GDFE’s. Each term in (5.134) corresponds to a MMSE-DFE estimate of \( x_n \) given the entire channel symbol output \( y \) and the previous values of \( x_1, \ldots x_n \). This corresponds to the GDFE’s series of MMSE estimates that use the channel output and past decisions – when the input sequence has diagonal autocorrelation \( R_{vv} = I \); an equivalent chain rule is then

\[
I(x \ y) = I(v_0; y) + I(v_1 \ y/v_0) + \ldots + I(v_n \ y/[v_1 \ldots v_{n-1}]) + \ldots + I(v_{N-1} \ y/[v_1 \ldots v_{N-2}])
\]

There are an infinite number of such possible transformations, leading to an infinite number of GDFEs, all with the same \( I \).

The gap applies to MSE, not just to noise. This MSE varies with dimension, so there can be variation when \( \Gamma > 0 \) dB when the gap-dependent SNR is

\[
SNR_{GDFE}(\Gamma) + 1 \Delta \prod_{n=1}^{N^*} \left( 1 + \frac{SNR_{bias,n} - 1}{\Gamma} \right)^{1/N_x}.
\]

The product is exactly equal to the original geometric SNR when \( \Gamma = 0 \) dB; otherwise

\[
\log_2 \left( 1 + \frac{SNR_{GDFE}(\Gamma)}{\Gamma} \right) \leq \log_2 \left( 1 + \frac{SNR_{GDFE}}{\Gamma} \right),
\]
so non-zero gap amplifies inaccuracy of energy distribution. This is analogous to Chapter 2’s situation where rate-sum became order dependent if gap increased above 0 dB.

**Zero-Forcing GDFE’s:** The zero-forcing GDFE sets zero noise in the MMSE solution and determines $G$ and $S_0$. Because there may be a multitude of equivalent MMSE structures that all have the same MMSE performance, caution needs to be exercised in assuming all ZF-GDFEs have the same performance with nonzero noise.

For any given GDFE, the ZF-equivalent uses the forward channel model to derive a ZF-GDFE. The forward channel output $z$ has lower-diagonal-upper Cholesky factorization of $R_f = G_f^* \cdot S_f^{} \cdot G_f^{}$; this corresponds to $R_b = R_f$ if $R_{nn} = 0$. Then $z$ is

$$z = G_f^* \cdot S_f^{} \cdot G_f^{} \cdot v + n' \quad \text{(5.138)}$$

The triangular $G_f$ also follows directly from QR factorization of the original channel, $\tilde{H} = S_f^{1/2} \cdot G_f^{} \cdot Q$ (ignore zeros to the left or below $G_f^{}$ to extract triangular part from the QR factorization if $\tilde{H}$ is not square and non-singular). The receiver can process $z$ with $S_f^{-1} \cdot G_f^{-*} \cdot z = G_f^{} \cdot v + n''$.

Ignoring the noise $n''$ as if it were zero produces a triangular set of equations (since $G_f$ is upper triangular) and letting “$\text{dec}$” represent a decoder for the code on each dimension?

$$\hat{v}_0 = \text{dec}_0(z'_0) \quad \text{(5.140)}$$

$$\hat{v}_1 = \text{dec}_1(z'_1 - G_{f,1,0} \cdot \hat{v}_0) \quad \text{(5.141)}$$

$$\hat{v}_2 = \text{dec}_2(z'_2 - G_{f,2,1} \cdot \hat{v}_1 - G_{f,2,0} \cdot \hat{v}_0) \quad \text{(5.142)}$$

$$\vdots = \vdots \quad \text{(5.143)}$$

$$\hat{v}_{N^*-1} = \text{dec}_{N^*-1}(z'_{N^*-1} - \sum_{i=0}^{N^*-2} G_{f,N^*-1,i} \cdot \hat{v}_i) \quad \text{(5.144)}$$

Equations (5.140) - (5.144) estimate the transmitted sequence, but the overall SNR (which is unbiased for ZF) is less than or equal to $2^{2I(x,y)} - 1$. This corresponding SNR product is (with 0 dB gap)

$$\text{SNR}_{zf-gdf} = \left[\prod_{n=1}^{N_x} (1 + S_{f,n})\right]^{1/N_x} - 1 \quad \text{(5.145)}$$

Equality to MMSE-GDFE holds under various conditions like the matrix $R_f$ is diagonal (that is $G_f = I$ and there is no feedback matrix) or certain types of noise (worst-case), as will be seen later in Section 5.3. Often, the ZF-GDFE, can have performance very close to the MMSE-GDFE, which is canonical and highest SNR level. The ZF design complexity is less because essentially just one Cholesky or QR factorization is necessary. An ML detector for the ZF-GDFE would achieve the full $I_i$ but would be very complex in most situations and is not necessarily equal to (5.140) - (5.144)’s back-substitution process, nor is ZF necessarily canonical.

**5.1.3.1 The Generalized Linear Equalizer - GLE**

GDFE implementation may impose additional constraints. One oft-encountered constraint is no feedback section (nor nonlinear precoder as in the upcoming Subsection 5.1.4, here called the Generalized Linear Equalizer (GLE). Such a linear-only discrete-modulation and/or discrete demodulation system retains canonical (and optimum) performance if $R_b$ is diagonal, or equivalently $G = I$. One example of such a system is Chapter 4’s vector coding, which the next Section 5.2 directly details further.

---

24If uncoded, then an instantaneous decision occurs for each subsymbol/dimension.
More generally, GLE performance will depend on the choice of $A$ and will not be the same for all the different square root choices of $R_{xx}$. Different $A$ choices for the same $R_{xx}$ can provide different GLE performance, unlike the GDFE where the performance is insensitive to the particular square-root choice of $A = R_{xx}^{1/2}$. The linear MMSE estimate of $v$ remains

$$\hat{v} = R_b \cdot z.$$  \hfill (5.146)

Indeed the MMSE matrix remains also $R_{ee} = R_b$ as in the canonical backward model. An ML detector acting on $R_b \cdot z$ can obtain optimum performance and highest data rate $I$, but can also be very complex. The GLE cannot use the GDFE’s feedback processing to simplify implementation but still achieve canonical performance. However the feedforward matrix $W$ minimizes MSE; further because this matrix has a separate row for each dimension output, it does indeed minimize also each dimensional MSE

$$MMSE_n \triangleq \mathbb{E} [ |e_n|^2 ].$$ \hfill (5.147)

The need for ML decoder complexity arises when $R_b$ is not diagonal. A decoder that ignores the correlation between error values on different dimensions is not optimum, nor is it canonical with non-diagonal $R_b$. However, it does have an SNR that applies to decoding if the error is (incorrectly) treated as AWGN:

$$SNR_{GLE,n} \triangleq R_{b,n}^{-1},$$ \hfill (5.148)

where $R_{b,n}$ is the $n^{th}$ diagonal term of $R_b$, which is only equal to $S_{0,n}$ if $R_b$ is diagonal. A GLE SNR forms as

$$SNR_{GLE} = \left[ \frac{N^*}{\prod_{n=1}^{N^*} R_{b,n}^{-1}} \right]^{1/N_x} = SNR_{GLE,u} + 1 .$$ \hfill (5.149)

Clearly $SNR_{GLE} \leq SNR_{GDFE}$ with equality when $R_b$ is diagonal. There is thus a GLE performance loss

$$0 \leq \gamma_{GLE} = \frac{SNR_{GLE} - 1}{SNR_{GDFE} - 1} \leq 1$$ \hfill (5.150)

with respect to using the feedback (or precoded) GDFE. The loss magnitude will depend on how close $R_b$ is to diagonal, which in turn will depend on the square-root choice $A$. Some choices lead to little or no loss, which Section 5.2 develops further.

An example use returns to the previous example and the $3 \times 3 R_{xx} = I$ input that a designer might try to implement:

```matlab
>> [snrGDFEu, GU, WU, S0, MSWMFU, b, bbar, snrLE] = computeGDFE(H, (1.5)*Rxxtt,2,3);
snrGDFEu = 8.1496 dB
snrLE = 4.6170 dB
```

Thus the nonlinear feedback gains roughly 4.5 dB in this example, even with the input roughly optimized.

### 5.1.4 The Precoded GDFE

Precoding averts error propagation by essentially moving the feedback section to the transmitter, where previous subsymbols are available without error. Section 3.8 introduced several types of suboptimal simple precoders, such as the Tomlinson and/or Laroia precoders. Similarly Chapter 2’s theoretical lossless precoders are the ideal limit of all good precoders. Practical precoder implementation can increase the transmit power slightly as in Section 3.8. The GDFE needs some straightforward modification for precoding.

Transmission design may use precoding to eliminate even error propagation within a GDFE symbol. The feedback section’s rows vary with subchannel index $n$ and

$$g_n \triangleq [1 \ g_{n,n-1} \ g_{n,n-2} \ldots \ g_{n,0}] \forall n = 0,\ldots,N^* - 1 .$$ \hfill (5.151)
The precoder again requires an unbiased feedback section:

\[ g_{U,n} = \left[ \frac{1}{\text{SNR}_n} g_{n,n-1} \ldots g_{n,0} \right] \]. \quad (5.152)

Each used dimension \( n = 0, \ldots, N^* - 1 \) has an associated \( \bar{b}_n = \frac{1}{2} \cdot \log_2 (1 + \varepsilon_n \cdot S_n) \). The lossless precoder then becomes periodic in the symbol period with operation

\[ v'_n = \left( v_n - \sum_{i=0}^{n-1} g_{U,n,i} \cdot v'_i \right) \varepsilon_n \], \quad (5.153)

as in Chapter 2’s BC. Each of the \( \tilde{N}^* \) dimensions of the feedforward matrix output should have a modulo operator prior to the decision element, except the first.

Section 3.8’s Tomlinson and Laroia precoders also easily follow with the periodic \( g_{U,n} \) vectors being used in the usual manner for each corresponding dimension; however, the Laroia precoder requires a cross-dimensional decoder processing (as in Section 3.8) after a first receiver decision element. So it would not be applicable, for instance, to Chapter 2’s broadcast channel).

5.1.5 Single-sided Receiver GDFEs

![Figure 5.9: The single-sided receiver GDFE.](image)

When the matrix \( H \) has rank \( \rho_H \) equal to the number of channel input dimensions, a special one-sided GDFE can occur. Such a \( H \) does not occur in the simple convolution case. However, examples of \( H \) that have such rank are:

**[Vector DMT]** The cyclic-block matrix \( H \) corresponding to use of a common symbol clock \( 1/T \) and \( \bar{b} \) and cyclic-prefix \( \nu \) at several different locations, each with their own IDFT discrete modulator with zero energy on all the other locations’ assigned energy - that is Chapter 4’s Vector DMT/OFDM with the additional constraint.

**[Multiple-Access Channel]** For instance, Chapter 2’s MAC has a matrix AWGN that corresponds to \( N \) dimensions independently exciting a common channel (for instance a MIMO transmission system with \( N \) independent transmit antennas and \( N \) or more receive antennae). Equivalentlky \( R_{xx} \) is diagonal.
The absence of an ability to coordinate the input dimensions with an $A$ or $C$ matrix modulator might correspond to physically distinct locations in which the transmitted signals entering the channel, like Chapter 2’s MAC. If the $H$ matrix has rank greater than or equal to the number of inputs, $A = C = I$, then no dimensional reduction is necessary. Sections 5.4 and 5.5 further address the low-rank situation on multi-user channels. Figure 5.9 illustrates the GDFE in the case of distributed input dimensions, which corresponds to Chapter 2’s MAC.

5.1.6 Single-sided Transmitter GDFE

A dual of Subsection 5.1.5’s situation in has channel-output dimensions in physically separated locations, and thus matrix receiver processing is not physically possible as with Chapter 2’s BC. The channel input dimensions are in a common location. If the matrix $H$ is square and nonsingular, THIS corresponds to Chapter 2’s BC having all primary users.

Figure 5.10 illustrates this situation. In this case, the matrix $\tilde{H}$ has QR-factorization $\tilde{H} = S_{f}^{1/2} \cdot G_{f} \cdot M^{*}$ where $S_{f}$ is the diagonal matrix from the ZF-GDFE, $G_{f}$ is the monic upper triangular matrix, and $M$ is a Hermitian matrix $(MM^{*} = M^{*}M = I)$. Then the transmit signal is $x' = M \cdot v$ and the receive signal undergoes $z' = S_{f}^{-1/2} \cdot z$, leaving

$$z' = G_{f} \cdot v$$  \hspace{1cm} (5.154)

A lossless precoder using $G_{f}$ to create the ZF-GDFE’s set of parallel independent subchannels. This choice leads to the overall SNR being

$$\text{SNR}_{zf-gdfe} = \prod_{n=0}^{N_{x}-1} S_{f,n} \cdot .$$  \hspace{1cm} (5.155)

When the noise is not white, then there may be additional performance loss with respect to GDFE levels, unless as first presented in Chapter 2, the noise has worst-case mutual-information-minimizing autocorrelation.
5.2 Special GDFEs: VC, Triangular DFE, and Massive MIMO

Figure 5.8’s GDFE for a matrix AWGN channel \((H,R_{nn})\) has performance, for Gaussian code with \(\Gamma = 0\) dB, that is a function of the input autocorrelation matrix \(R_{xx}\) with maximum data rate \(I(x;y) = I(v;y)\). Best design considers the GDFE input vector to be \(v\), with \(R_{vv} = I\) and any energy scaling absorbed into the channel matrix \(\tilde{H}\) through the discrete modulator \(A\). The GDFE receiver then directly detects the encoded symbol-vector components \(v_n, n = 0,\ldots,N^*-1\) in succession to achieve this canonical performance maximum of \(\text{SNR} = 2^{2(I(x;y) - 1)}\). (5.156)

The system can reliably transmit at \(\bar{b} \leq I(x;y)\) with any good (scalar) AWGN code with sufficiently small gap \(\Gamma\) on all energized/passed GDFE dimensions/subchannels \(v\) corresponding to the given input autocorrelation \(R_{xx}\) and retains this same gap to best data rate for that \(R_{xx}\) on matrix channel \(H\).

**Proof:** the development in Section 5.1. QED.

**Lemma 5.2.1 [Finite-Length CDEF Equivalent]** Any matrix-AWGN has a GDFE system with receiver input \(y\) and transmitter input \(v\) and input autocorrelation \(R_{vv} = I\) such that

\[
\text{SNR}_{GDFE,U} = 2^{2(I(x;y) - 1)}.
\]

Sufficiently small gap merits attention: The gap applies to the MMSE on each GDFE (scalar) output dimension, and when \(\Gamma > 0\) dB, there is an inaccuracy as per (5.137) and a dependency on GDFE dimensional order (no such inaccuracy nor dependency exists when \(\Gamma = 0\) dB). However, the useful result is that with a reasonably good code, this slight deviation is tolerable and the canonical GDFE transmission system is often substantially less complex than an ML detector for this matrix channel. Effectively, the GDFE decouples the demodulation optimization from the good scalar code’s decoding. Good codes for the AWGN also apply equally well when the subsymbol constellation is limited to a Voronoi region like a square. This additional limitation (with average energy remaining the same) loses 1.53 dB of shaping gain. Indeed good codes exist that capture the minimum distance properties without regard to the subsymbol constellation boundary. They are the same codes, as Chapter 10 illustrates on the separation of shaping codes from fundamental-gain codes, the latter of which are those used by designers. These can reliably achieve \(P_e \rightarrow 0\) for any rate less than \(\log_2(1 + \text{SNR}_{\gamma_{s,max}})\). The \(\gamma_{s,max} = 1.53\) when the constellation boundary is a square. Thus GDFE results also are canonical for such systems, or really any system where the loss is \(\gamma_s\) for the non-hyperspherical boundary. Thus, good code expands to mean any code that has essentially infinite fundamental coding gain, or more explicitly any code that with probability one satisfies the AEP constraint of only one codeword in each conditional typical set within the boundary.

Imperfect Codes with any boundary: More specifically, the GDFE generates a set of parallel AWGN subchannels that each have mutual information \(I_n\). These subchannels or dimensions have an energy allocation \(E_{v,n}\) that the matrix \(A\) absorbs. The dimensional codes use constant or variable constellations with corresponding appropriate decoder, as per Chapter 4’s Separation Theorem, with \(b_n \approx I_n\). There are several input energy distributions that correspond to different GDFE dimensional orders, each with different individual-dimensional mutual information, where the differences are among the dimensionally indexed

\[
\bar{I}_n = \frac{1}{2} \log_2 (S_{0,n}) . \tag{5.157}
\]

However, these add to the same total \(\bar{I} = \sum_n \bar{I}_n\) for all these energy distributions; each such energy distribution corresponds to a different discrete modulator \(A\). With nonzero gap code the loaded data
rate (bits/dimension) is \( \bar{b}_n = \frac{1}{2} \log_2 \left( 1 + \frac{S_{0,n} - 1}{\Gamma} \right) \). (5.158)

Equation (5.158) loses accuracy as \( \Gamma \) increases, similar to Chapter 4's Separation Theorem accuracy loss (as per (5.158)) because the code is not near capacity. Separation of coding and modulation works best with good codes.

When the input \( x \)'s dimensions, have a water-filling energy allocation and a good code, they thereby create an \( R_{xx} \) that maximizes \( I(x;y) \), and thereby achieve the highest possible data rate, or capacity. So, nothing is lost by reusing a good-scalar-AWGN-designed code on this GDFE-processed matrix-AWGN channel. This section finds ways to construct such a best \( R_{xx} \) input and a corresponding GDFE. Similarly, as \( \Gamma \) increases, Chapter 4’s gap-dependent water-fill energy (recall \( K = E_n + \Gamma/g_n \)) narrows the water-fill band (which can reduce the number of used dimensions).

**GDFE basic types:** There exist many \( R_{vv} = I \) inputs that achieve the canonical level, corresponding to different feedback sections (and feedforward sections), but the same \( \text{SNR}_{GDFE,U} \). This section investigates two specific forms:

1. **Vector Coding** - which corresponds to the matrix \( A = \Phi \) resulting from eigendecomposition of \( R_{uu} \) to obtain \( \Phi \) as the eigenvector matrix that scales by the diagonal square-root-eigenvalue matrix.

2. **Triangular GDFE** - which corresponds to \( R_{uu} \)'s Cholesky factorization \( R_{vv} = \Phi \cdot \Phi^* \) to obtain triangular (causal) \( \Phi \) that scales by the diagonal square-root matrix of Cholesky factors, and includes the important special case of cyclic GDFE or the CDFE when \( H \) (or \( \tilde{H} \)) is cyclic, only the latter of which corresponds to Chapter 3’s original MMSE-DFE.

As \( N \to \infty \) (or equivalently as \( N_x \to \infty \) with \( L_x = L_y = 1 \) for each spatial transfer) for stationary (Toeplitz) \( R_{uu} \), then VC \( \to \) MT and the GDFE \( \to \) MMSE-DFE set. This holds if the \( R_{xx} \) is the same (and each method uses scalar AWGN codes with the same gap on all dimensions), even if that \( R_{xx} \) is not a water-fill solution. If the water-fill \( R_{xx} \) is used in all, then all attain the highest performance level for the given gap. Section 5.3 considers the construction of the optimum water-filling or any other desired \( R_{xx} \) for the CDFE and GDFE to avoid violation of the PWC (which will effectively result in a minimum number of parallel CDFE’s that correspond to Chapter 3’s minimum-size set of MMSE-DFEs).

### 5.2.1 Vector-Coding as the GDFE without feedback

Vector coding is a special GDFE case. First, noise-whitening replaces \( H \) by \( R_{nn}^{-1/2} \cdot H \) and applies SVD

\[
R_{nn}^{-1/2} \cdot H = F \cdot \Lambda \cdot M^*.
\] (5.159)

Vector coding’s input vectors are this channel’s right-singular-vectors, \( M \),

\[
x = \sum_{n=1}^{N_x} X_n \cdot m_n = M' \cdot X'.
\] (5.160)

The choice \( R_{uu} = M \cdot R_{xx} \cdot M^* \) thus characterizes vector-coding – any other \( R_{uu} \) is not vector coding, but could correspond to other GDFEs. Channel-pass-space reduction removes the column vectors \( m_n, n > \rho \tilde{H} \) from the \( M \) (the singular values are ordered from largest to smallest) so to obtain \( \tilde{M} \)

\[
x' = \sum_{n=1}^{\rho \tilde{H}} X_n \cdot m_n = \tilde{M} \cdot X'.
\] (5.161)

\(^{25}\) recall the unbiased SNR is \( SNR = SNR_{bias} - 1 \) for any gap
Vector coding with water-filling may also zero some pass-space dimensions, which then creates a singular $R_{xx}$. The corresponding input autocorrelation matrix, $R_{xx}x'$, is then (with $M''$ as the first $\varrho_x$ columns of $M'$)

$$x'' = \sum_{n=1}^{N''=\varrho_x} X_n \cdot m_n = M'' \cdot X'' .$$

(5.162)

and finally then

$$v_n = \frac{X_n''}{\sqrt{\mathbb{E}[X_n'' \cdot X_n'']}} n = 1, \ldots, \varrho_x ,$$

(5.163)

with then $A = M \cdot \text{Diag}(\sqrt{\mathbb{E}_n})_{n=1,\ldots,\varrho_x}$. The forward canonical matrix $R_f$ is then diagonal

$$R_f = \hat{H}^* \cdot \hat{H} = \text{diag}\left\{ \mathcal{E}_\varphi \cdot \lambda_{\varrho_x}^2 , \mathcal{E}_{\varrho_x-1} \cdot \lambda_{\varrho_x-1}^2 , \ldots , \mathcal{E}_1 \cdot \lambda_1^2 \right\} ,$$

(5.164)

and thus $R_f^{-1} = R_f + I$ is also diagonal. Consequently, $G = I$. The feedback section thus simplifies to no feedback at all (which avoids the error-propagation concern with suboptimal dimension-by-dimension detection), and the feedforward section simply becomes $S_0^{-1} = R_b$, an independent (diagonal) scaling of each received matched-filter-output dimension.

**Lemma 5.2.2 [Vector Coding as the Optimum and Canonical Transmission System]** Vector coding is both canonical and optimum as a transmission method for all $\Gamma \geq 0$ dB.

**Proof:** Since the GDFE provides $\text{SNR}_{d_{gfe}} = 2^{2\bar{\mathcal{I}}(x;y)}$, VC is canonical. Since there is zero GDFE feedback section, $G = I$, there is no concern for suboptimal detection, and an independent ML decoder on each subchannel (which is a simple slicer for uncoded transmission, but would be the ML decoder for the code applied to that dimension more generally) is optimum. QED.

The VC-based GDFE’s joint optimality and canonical nature is a special discrete modulator with modulating vectors equal to the channel’s right singular vectors. With bias removed, VC further has a MMSE equal to the (scaled) noise, so the gap only multiplies such noise. While the effect in (5.137) remains, the amplification of other dimensions’ signal components is also absent, so any gap-approximation inaccuracies are less than with a nonzero-feedback GDFE. The detector in this one VC-GDFE case is indeed ML. No other\(^{26}\) structure produces a diagonal channel with the consequence of $G = I$. Any other $G \neq I$ may have canonical SNR ($2^{2\bar{\mathcal{I}}(x;y)}$) but assumes correct decisions and thus is not ML nor optimum as a detector. Thus, VC is truly the best performing GDFE structure for transmission on the channel $(H, R_{nn})$. A receiver could implement an ML detector for $G \neq I$, but such a full ML detector with $G \neq I$ is likely much more complex than VC.

### 5.2.2 The triangular GDFE

The GDFE can also use Cholesky factorization to relate a nonsingular input autocorrelation matrix $R_{uu}$ to a diagonal input autocorrelation matrix $R_{vv}$:

$$R_{uu} = \Phi \cdot \Phi^* = G_\phi \cdot S_u \cdot G_\phi^* ,$$

(5.165)

\(^{26}\)A part from trivial phase differences; a factorization that allows complex entries for $\Lambda$ is essentially the same as one with a positive real $\Lambda$. 835
where $\Phi$ is upper-triangular.\(^{27}\)

![Figure 5.11: Triangular DFE with white input $v$.](image)

Furthermore,

$$u = \Phi \cdot v$$  \hspace{1cm} (5.166)

The data-message’s subsymbols are the $\bar{N}^*$ components of $v$, and

$$x' = \tilde{A} \cdot u = \left( \tilde{A} \cdot \Phi \right) \cdot v$$  \hspace{1cm} (5.167)

The corresponding GDFE has design for

$$\bar{H} = R_{nn}^{-1/2} \cdot H \cdot \tilde{A} \cdot \Phi = R_{nn}^{-1/2} \cdot H \cdot A$$  \hspace{1cm} (5.168)

and otherwise follows exactly as in Section 5.1 with input $v_n$, $n = 0, ..., \bar{N}^* - 1$ as input data and also as the GDFE’s feedback-section input. This structure appears in Figure 5.11.

The GDFE’s feedforward matrix and feedback matrices remain nonzero triangular (and not necessarily diagonal), leading to a higher implementation complexity and suboptimal detection; however, because the input has diagonal $R_{vv} = I$ by construction, this GDFE is still canonical. The $\text{SNR}_{gdf}$ performance still approaches capacity just like the VC system. This GDFE’s most interesting temporal-channel quality may be the triangular transmission filter, which suggests a causally implemented filter when the input $R_{xx}$ can be made nonsingular as $N \to \infty$. Causal filters satisfy the PWC. Upper-triangular filters can be implemented essentially without delay in this special case. In general, however, the discrete modulator $\tilde{A}$ can cause $A = \tilde{A} \cdot \Phi$ to be non triangular and there is then no causal-like implementation.

Even though the GDFE input $v$ is not the same as VC’s modal input $v$ and the decision-based detector is not optimum, the system-input vector $v$ carries the same information that leads to the same GDFE SNR in the absence of any previous-decision errors. The parallel-subchannel set is also different, even though the overall equivalent AWGN has the same SNR. Both operate reliably at capacity. Figure 5.12 illustrates the linear prediction relationships for the triangular GDFE transmitter. With respect to Chapter 3’s infinite-length MMSE-DFE, $\Phi$ replaces $G_x(D)$ in the finite-length symbol case.

\(^{27}\)The upper triangular matrix correctly appears first here, unlike the factorization of $R_b^{-1}$ where the upper triangular matrix appeared last. The upper-lower matlab Cholesky should use the chol(JRuuJ) command where $J$ has all ones down the anti-diagonal and zeros elsewhere, and then \textbf{pre- and post-} multiply the resultant matrix by $J$, and then take conjugate transpose to obtain Gxbar, or in matlab Gxbar=(J*chol(J*rxx*J))*J', and then finally remove diagonal terms by post multiplying as Gx=Gxbar*inv(diag(diag(Gxbar))), to obtain $G_x$. $R_{vv}$ is diag(diag(Gxbar))*diag(diag(Gxbar)). For those with access to Appendix E’s programs, the matlab program lohc.m directly computes this same Gx=lohc(rxx).
EXAMPLE 5.2.1 [1 + 0.9D⁻¹ Triangular GDFE] This example revisits Example 5.1.2, but this time uses the vector-coded modulation vectors directly. The following matlab commands should be self explanatory and work the complete process from channel to final design for the triangular GDFE on this channel with \( N = 2 \) and \( \nu = 1 \) (\( \bar{N}_s = 3 \)).

Recalling
\[
H = \begin{bmatrix}
2.1155 & 2.3505 & 0 \\
0 & 2.1155 & 2.3505
\end{bmatrix}
\]

\[
\begin{bmatrix}
Ct & Ot & \text{Ruutt}
\end{bmatrix} = \text{fixmod}(H, \text{eye}(3), \text{eye}(3));
\]
\[
\begin{bmatrix}
A & OA & \text{Ruupp}
\end{bmatrix} = \text{fixin}(\text{Ruutt}, Ct);
\]
% previous square-root forms as
\[
\text{Gxbar} = \text{lohc}(\text{Ruupp});
\]
\[
\text{Gx} = \text{Gxbar} \ast \text{inv(diag(diag(Gxbar)))} ;
\]
\[
\text{Xmit} = A \ast \text{Gxbar};
\]

Continuing now with the square-root matrix Xmit that includes both \( A \) and the Cholesky square-root of \( R_{u'u} \)

\[
[\text{snrGDFEu}, \text{GU}, \text{WU}, \text{SO}, \text{MSWMFU}, \text{b}, \text{bbar}] = \text{computeGDFE}(H, \text{Xmit}, 2, 3);
\]
\[
\text{snrGDFEu} = 5.5427 \text{ dB}
\]
\[
\begin{bmatrix}
\text{GU} =
1.0000 & 0.5731 \\
0 & 1.0000
\end{bmatrix}
\]
\[
\begin{bmatrix}
\text{WU} =
0.1328 & 0 \\
0.0492 & 0.0972
\end{bmatrix}
\]
\[
\begin{bmatrix}
\text{SO} =
8.5275 & 0 \\
0 & 11.2899
\end{bmatrix}
\]
\[
\begin{bmatrix}
\text{MSWMFU} =
0.3645 & 0.0000 \\
0.0179 & 0.3073
\end{bmatrix}
\]
\[
\begin{bmatrix}
\text{b'} =
1.5461 & 1.7485
\end{bmatrix}
\]
\[
\begin{bmatrix}
\text{bbar} =
1.0982
\end{bmatrix}
\]
The receiver design needs $G_U$ (or GU above) and the forward combined filter MSWMFU above. The transmitter implements Xmit. Further there are two used (real) dimensions with the bit distributions shown on each.

5.2.3 The Circulant DFE (CDFE)

The $N \times (N + \nu)$, or $N_y \times N_x$ with $L_x = L_y = 1$, temporal convolution matrix $H$ can never have rank greater than $N$. Also, the square $N \times N$ matrix $H$ that arises from cyclic-prefix use (the same cyclic prefix used for DMT) always has rank $\rho_H \leq N$. The case $\rho_x = N^* = \rho_H = N$ is occurs when the input autocorrelation matrix $R_{xx}$ corresponds to a sampling rate where loading energizes all input dimensions. One situation where $\rho_H < N$ occurs when the channel has notches (zero gain) at the exact frequencies that would be considered “DMT subcarrier” frequencies\textsuperscript{28}. Best input design always zeroes energy on any such notch frequencies, which leads to multiple discontiguous bands as Section 5.3 addresses. When only a single band (subset) of carriers have nonzero energy, the CDFE sometimes has the name “Single-Carrier OFDM.”

In the simpler nonsingular-input case,

$$R_{uu} = R_{xx},$$

(5.169)

the input factorization trivially becomes Cholesky factorization, and

$$x = G_\phi \cdot v$$

(5.170)

with $G_\phi$ a upper-triangular $N \times N$ matrix. This situation can correspond to the circulant DFE or CDFE. The CDFE case (with cyclic prefix) has a stationary channel with a circulant Toeplitz $H$ and finite fixed $\nu$. As $N \to \infty$, this CDFE converges to Chapter 3’s MMSE-DFE as in Section 5.3. With input singularity corresponding to input design zeroing certain non-zero-measure frequency bands, or equivalently failing to satisfy the PWC, then the situation has instead multiple CDFEs each converging to its own corresponding MMSE-DFE in its own PWC-satisfying band (see Section 3.11).

Convergence: For finite large $N$, the transmit filter corresponds to any of $G_\phi$’s middle rows, while the MMSE-DFE’s feedback section corresponds to the middle rows of the CDFE’s $G$.\textsuperscript{29} The CDFE’s matched-filter matrix converges to a matched filter, and any feedforward matrix middle row converges to the MMSE-DFE feedforward filter. The CDFE exists only when $R_{xx}$ is nonsingular, which may not occur, especially with water-filling’s typical dimensional zeroing. Section 5.3.4 provides an adjustment for multiple disjoint bands, which is called “generalized Cholesky factorization.” Under the cyclic-channel-matrix restriction, DMT is clearly an optimum transmission method with best SNR and ML detection achieved because $G = I$, and there is no error propagation. The CDFE when its input exists and matches DMT, canonically obtains the same SNR and reliable data rate, but uses a suboptimal detector. An ML detector for this CDFE can be very complex, but would match canonically the simpler DMT performance. Chapter 4’s Separation Theorem now takes additional support in that basically the C-OFDM system and CDFE have the same performance if the simple AWGN code has $\Gamma = 0$ dB. So C-OFDM is also canonical if it uses the same $R_{xx}$ and a very low $\Gamma$ code. Chapter 4’s Separation-Theorem discussion notes that use of large-gap codes can erode the C-OFDM/DMT equivalence rapidly. Again, the good code that ignores only shaping gain exhibits all the same canonical performance, for an SNR reduced by the shaping loss.

EXAMPLE 5.2.2 [1 + .9 · $D^{-1}$ CDFE with PAM input] Returning to this text’s familiar $1 + .9 · D^{-1}$ example, the AWGN noise variance is $\frac{N_0}{2} = .181$ and $\bar{E}_x = 1$. This cyclic prefix is length $\nu = 1$ on a block size of $N = N = 8$ ($N_x = 9$). Thus, $R_{uv} = R_{uu} = I$. For this case, $A = I$ also because $R_{uv}$ is diagonal. Rather than repeat the analysis of previous developments mathematically here, this example simply inserts some descriptive matlab text with results:

\textsuperscript{28}Equivalently $H(D) = \prod_{n=1}^{N} (1 + aD)$ has at least one $n$ for which $|a_n| = 1$.

\textsuperscript{29}All these middle rows will have the nonzero coefficient values tending to the same fixed filter when $N$ is long enough.
>> H=toeplitz([.9 zeros(1,7)]',[.9 1 zeros(1,6)]);
>> H(8,1)=1
H =
0.9000 1.0000 0 0 0 0 0 0
0 0.9000 1.0000 0 0 0 0 0
0 0 0.9000 1.0000 0 0 0 0
0 0 0 0.9000 1.0000 0 0 0
0 0 0 0 0.9000 1.0000 0 0
0 0 0 0 0 0.9000 1.0000 0
0 0 0 0 0 0 0.9000 1.0000
1.0000 0 0 0 0 0 0 0.9000
>> [snrGDFEu, GU, WU, S0, MSWMFU, b, bbar] = computeGDFE(1/sqrt(.181)*H, eye(8), 2, 9)
>> snrGDFEu = 7.1666 dB
>> GU =
0 0 0 0 0 0 0 0.4972
1.0000 0.4972 0 0 0 0 0 0
0 1.0000 0.6414 0 0 0 0 -0.2899
0 0 1.0000 0.6930 0 0 0 0.1780
0 0 0 1.0000 0.7128 0 0 -0.1113
0 0 0 0 1.0000 0.7206 0 0 0.0702
0 0 0 0 0 1.0000 0.7237 -0.0444
0 0 0 0 0 0 1.0000 0.7531
0 0 0 0 0 0 0 1.0000
>> MSWMFU =
0.2115 0 0 0 0 0 0 0.2351
0.1798 0.2729 0 0 0 0 0 -0.1371
-0.1104 0.1601 0.2948 0 0 0 0 0.0841
0.0691 -0.1002 0.1525 0.3033 0 0 0 0.0526
-0.0435 0.0631 -0.0961 0.1495 0.3066 0 0 0.0332
0.0275 -0.0399 0.0608 -0.0945 0.1483 0.3079 0 -0.0210
-0.0174 0.0253 -0.0385 0.0598 -0.0939 0.1478 0.3084 0.0133
-0.0933 0.0418 0.0010 -0.0439 0.0961 -0.1687 0.2772 0.1647
>> b' =
1.7297 1.5648 1.5156 1.4978 1.4909 1.4882 1.4871 1.0792
>> bbar = 1.3170

The important G matrix clearly converges to what is known (from Chapter 3’s instance of this same example in multiple places) to be the (unbiased) MMSE-DFE feedback section of 1 + .75 · D in the middle rows of G_u. This behavior would happen for any sufficiently large N on this example and illustrates that the MMSE-DFE is simply the limit of the CDFE as N becomes large. That is, no symbol-blocking is necessary if the transmitted symbol sequence is just long enough. However, at the beginning of transmission, especially on this maximum phase channel, the filter is different during a transient. An MMSE-DFE designed according to Chapter 3 for finite or infinite-length filters would actually only achieve its 8.4 dB SNR asymptotically, because the assumptions made in that derivation were stationarity for all time, when in fact this example’s more realistic situation illustrates that transmission begins at one point in time and continues. Thus, the CDFE then processes the initial dimensions/samples better than any MMSE-DFE in Chapter 3. Indeed the 7.16 dB SNR here increases to 7.75 dB at N = 16 (real baseband N_x = N + 1), 8.26 dB at N = 100 and 8.3 dB at N = 200, which all illustrate the fastest or best convergence to the eventual 8.4 dB. Thus, use of a stationary-designed MMSE-DFE over finite block lengths does not attain the asymptotic performance. Best performance instead would use the finite-length CDFE with N equal to the packet size.

This example’s feedforward filter also converges, but clearly has significant terms for all di-
dimensions at least for \( N = 8 \). The convergence is evident in the middle rows to \([0.22 \ 0.36 \ 0.58 - 0.92 \ 0.145]\). The non-zero "end" dimension here is related to \( \nu = 1 \) and eventually just implies an implementation with Chapter 3’s finite-equalizer-causality delay parameter \( \Delta > 0 \) to handle the maximum-phase channel.

It is interesting to compare the CDFE with a conventional GDFE as in the design of Example 5.1.2 except that \( N \) is increased from 2 there to 8 here to produce the results with now energy fed into each of the dimensions avoiding the channel singularity on one dimension and transporting its energy instead on the other dimensions.

\[
H = \frac{1}{\sqrt{0.181}} \text{toeplitz}([0.9 \ 0 \ldots 0 \ 0 \ 0 \ldots 0], [0.9 \ 1 \ 0 \ldots 0 \ 0 \ldots 0])
\]

\[
[Ct, Ot, Ruutt] = \text{fixmod}(H, \text{eye}(9), \text{eye}(9));
\]

\[
Ct
\]

\[
0.7764 \quad 0.2012 \quad -0.1811 \quad 0.1630 \quad -0.1467 \quad 0.1320 \quad -0.1188 \quad 0.1069 \quad 0.1069
\]

\[
0.2012 \quad 0.8189 \quad 0.1630 \quad -0.1467 \quad 0.1320 \quad -0.1188 \quad 0.1069 \quad -0.0962 \quad 0.0866
\]

\[
-0.1811 \quad 0.1630 \quad 0.8533 \quad 0.1320 \quad -0.1188 \quad 0.1069 \quad -0.0962 \quad 0.0866 \quad -0.0779
\]

\[
0.1630 \quad -0.1467 \quad 0.1320 \quad 0.8812 \quad 0.1069 \quad -0.0962 \quad 0.0866 \quad -0.0779
\]

\[
-0.1467 \quad 0.1320 \quad -0.1188 \quad 0.1069 \quad 0.9038 \quad 0.0866 \quad -0.0779 \quad 0.0702
\]

\[
0.1320 \quad -0.1188 \quad 0.1069 \quad -0.0962 \quad 0.0866 \quad 0.0702 \quad -0.0631 \quad 0.0568 \quad 0.0460
\]

\[
-0.0962 \quad -0.0962 \quad 0.0866 \quad -0.0779 \quad 0.0702 \quad -0.0631 \quad 0.0568 \quad -0.0511
\]

\[
\]

\[
 Ruupp
\]

\[
6.3966 \quad -4.8569 \quad 4.3712 \quad -3.9341 \quad 3.5407 \quad -3.1866 \quad 2.8680 \quad -2.5812 \quad 2.3231
\]

\[
-4.8569 \quad 5.3712 \quad -3.9341 \quad 3.5407 \quad -3.1866 \quad 2.8680 \quad -2.5812 \quad 2.3231 \quad 2.3231
\]

\[
4.3712 \quad -3.9341 \quad 4.5407 \quad -3.1866 \quad 2.8680 \quad -2.5812 \quad 2.3231 \quad -2.0908 \quad 1.8817
\]

\[
-3.9341 \quad 3.5407 \quad -3.1866 \quad 3.8680 \quad -2.5812 \quad 2.3231 \quad -2.0908 \quad 1.8817 \quad 1.6935
\]

\[
3.5407 \quad -3.1866 \quad 2.8680 \quad -2.5812 \quad 3.3231 \quad -2.0908 \quad 1.8817 \quad -1.6935 \quad 2.2346
\]

\[
-3.1866 \quad 2.8680 \quad -2.5812 \quad 2.3231 \quad -2.0908 \quad 2.8817 \quad -1.6935 \quad 1.5242 \quad 2.5242 \quad -1.3717
\]

\[
2.8680 \quad -2.5812 \quad 2.3231 \quad -2.0908 \quad 1.8817 \quad -1.6935 \quad 2.5242 \quad -1.3717 \quad 2.2346
\]

\[
-2.5812 \quad 2.3231 \quad -2.0908 \quad 1.8817 \quad -1.6935 \quad 1.5242 \quad -1.3717 \quad 2.2346
\]

\[
\]

\[
 Gxbar=\text{lohc}(Ruupp);
 Gx=\text{Gxbar*inv(diag(diag(Gxbar)))} ;
 Xmit=\text{A*Gxbar}
\]

\[
Xmit
\]

\[
0.8812 \quad -0.0000 \quad 0.0000 \quad -0.0000 \quad 0.0000 \quad -0.0000 \quad 0.0000 \quad 0.0000
\]

\[
0.2283 \quad 0.8757 \quad 0.0000 \quad -0.0000 \quad 0.0000 \quad -0.0000 \quad 0.0000 \quad 0.0000
\]

\[
-0.2055 \quad 0.2397 \quad 0.8681 \quad -0.0000 \quad 0.0000 \quad -0.0000 \quad 0.0000 \quad 0.0000
\]

\[
0.1850 \quad -0.2157 \quad 0.2554 \quad 0.8574 \quad -0.0000 \quad 0.0000 \quad -0.0000 \quad 0.0000
\]

\[
-0.1665 \quad 0.1942 \quad -0.2299 \quad 0.2779 \quad 0.8416 \quad 0.0000 \quad -0.0000 \quad -0.0000
\]

\[
0.1948 \quad -0.1747 \quad 0.2069 \quad -0.2501 \quad 0.3120 \quad 0.8163 \quad 0.0000 \quad 0.0000
\]

\[
-0.1348 \quad 0.1573 \quad -0.1862 \quad 0.2251 \quad -0.2808 \quad 0.3678 \quad 0.7710 \quad -0.0000
\]
Continuing with this non-cyclic alternative transmit-filter matrix

\[
\begin{bmatrix}
0.1213 & -0.1415 & 0.1676 & -0.2026 & 0.2527 & -0.3310 & 0.4733 & 0.6690 \\
-0.1092 & 0.1274 & -0.1508 & 0.1823 & -0.2274 & 0.2979 & -0.4260 & 0.7433
\end{bmatrix}
\]

Because there is no cyclic-prefix energy loss, this solution is slightly better, however the filters are less stationary in this case and may not converge as quickly, but the performance is better (because no energy is wasted in a cyclic prefix or in the channel’s null space), and the SNR is 7.5 dB. Vector Coding with equal-energy transmission on the 8 best singular-value dimensions would achieve this same performance level. This nonstationary triangular GDFE more rapidly converges in SNR with increasing \(N\) to a point intermediate to the 8.4 dB of the MMSE-DFE (which effectively is the CDFE and inserts energy into a useless dimension) and less than the 8.8 dB maximum that only occurs with water-filling. (\(N_y = N, N_x = N + \nu\))

\[
8.4 \text{ dB } < \lim_{N \rightarrow \infty} \text{SNR}_{N_x/N_y,GDFE,u} \leq 8.8 \text{ dB }.
\]

### 5.2.4 Diagonal Dominance and Massive MIMO Linear GDFE Simplification

Chapter 2 defines massive MIMO as \(L_y >> U\) for the multiuser MAC and \(L_x >> U\) for the multiuser BC, where \(U\) is the number of users. In a more general context, \(U << L\) simply means that \(H\) matrix is very “tall” or very "fat," respectively; this leads to the term “massive.” GDFEs have a useful situation for a wide array of commonly encountered tall/fat \(H\) channel matrices that have low correlation between the linearly independent \(U\) columns or rows respectively; essentially the GDFE simplifies to a linear, but still canonical, structure where \(G \rightarrow I\), i.e., no feedback nor precoder is necessary.

---

30 Any infinite-length extension will deviate by a vanishingly small amount from cyclic as \(N \rightarrow \infty\).
MU-MIMO, Massive-MIMO, Multiple Access Channel: Typically convolution matrices do not have this structure, and usually are not tall/fat for reasonable frequency-time decompositions (closer to square if $\nu << N$, or more generally $N_x \approx N$). With MAC channel elements $\tilde{H} = [h_{i,k}]_{i=0,...,L_y;k=0,...,U}$, the correlation requirement is (for each tone in vector DMT)

$$
\mathbb{E} \left[ h_{i,\ell} \cdot \hat{h}_{k,\ell}^* \right] \approx |\hat{h}|^2 \cdot L_y \cdot \delta_{i,k}
$$

(5.172)

where the “$\mathbb{E}$” expectation has the dimensional spatial-sample-average approximation:

$$
(\hat{h}_{i,\ell} \cdot \hat{h}_{k,\ell}^*) = \frac{1}{L_y} \cdot \hat{h}_{i,\ell}^* \cdot \hat{h}_{k,\ell} = \frac{1}{L_y} \cdot \sum_{\ell=0}^{L_y-1} \hat{h}_{i,\ell}^* \cdot \hat{h}_{k,\ell} \approx |\hat{h}|^2 \cdot \delta_{i,k}.
$$

(5.173)

This means that $\hat{H}^* \cdot \hat{H}$ tends toward diagonal, or is diagonally dominant. When the system is MIMO (usually wireless), this means the spatial correlation between the different users’ signals arriving at the common receiver is low, and indeed the random-vector columns corresponding to the different users received signals are independent. A convolution matrix rarely has this structure. Crosstalking cables of wires have this structure also for some frequency bands. This sometimes also has the name “rich scattering” in that the different users’ MIMO spatial channels robustly differ from one another.

The GDFE has a crucial step of factoring the $U \times U$ (really $U^o \times U^o$ but all users are primary in the massive MIMO case)

$$
R_b^{-1} = R_f + I = G \cdot S_0 \cdot G^*. \quad (5.174)
$$

The MAC spatial inputs will have a diagonal $R_{xx}$, so a diagonal $R_b^{-1}$, equivalently diagonal $R_f$, implies that $G = I$ and no feedback is necessary, but retains canonical GDFE performance. The MAC also has $A = I$, so then

$$
R_f = \tilde{H}^* \cdot \tilde{H}. \quad (5.175)
$$

While $R_b$ and $R_f$ have the lower dimensionality $U$, the Massive MIMO matrix $\tilde{H}$ has tall columns, so that (5.173) will be true, particularly in the Law of Large Numbers (LLN, see Chapter 2) sense. Equivalently $R_b/L_y$ or $R_f/L_y$ will have very small or negligible off-diagonal terms, particularly when $L_y >> U$. Indeed in many wireless applications just $L_y \geq U$ is sufficient, so often linear feedforward-only receiver matrix filtering (basically the GDFE’s $W_{urb} \cdot \hat{H}^*$) is sufficient for canonical performance (and indeed no error propagation either). Thus, much literature actually only teaches the linear matrix filter for the MAC as sufficient. This is not true in general, even with arbitrary tall matrices $\tilde{H}$, but is true when the low spatial correlation in (5.173) holds. When this situation does hold, the MAC (or channel more generally) is diagonally dominant.

MU-MIMO, Massive-MIMO, Broadcast Channel: Again for each $\tilde{H}_n$, the BC is slightly more complex to analyze, but exhibits the same effect when $L_x >> U$. In this case, with any energy scaling from $R_{uu}$’s normalization absorbed into the special square-root transmitter matrix $A = R_{x,x}^{1/2}$

$$
R_b^{-1} = A^* \cdot H^* \cdot R_{wcn}^{-1} \cdot H \cdot A + I
$$

(5.176)

where $R_{wcn}$ is the $U \times U$ worst-case noise autocorrelation, whose presumed existence limits the BC achievable data rates. In this case the fat matrix is $H$ (equivalently also $\tilde{H} = R_{wcn}^{-1/2} \cdot H$ is fat) and $A$ is tall. If the BC’s rows - instead of columns – (including worst-case noise adjustment) satisfy the equivalent of (5.173), then any $A$-case that attempt to match-filter to the BC to increase energy transfer would have column $a_u$ such that

$$
(\hat{h}_{\ell,i} \cdot a_{\ell,k}^*) = \frac{1}{L_x} \cdot \sum_{\ell=0}^{L_x-1} \hat{h}_{\ell,i}^* \cdot a_{\ell,k} \approx \delta_{i,k}.
$$

(5.177)

Again by the LLN, this same effect of diagonal dominance causes $G = I$ when $L_x >> U$. Not all BC’s satisfy (5.177), even with large $L_y$. However in practice, $L_x \geq U$ is often sufficient for $R_b^{-1}$ to be diagonally dominant and the feedback (lossless precoder for the BC) becomes unnecessary to approach the GDFE’s canonical performance level.

842
5.2.4.1 Generalized Cholesky Algorithm

The Generalized Cholesky Algorithm follows and operates on the reordered input that occurs during the creation of $J_A$. Traditional Cholesky acts upon the nonsingular reordered autocorrelation matrix

$$R_{xx}(N^* - 1) = \Phi \cdot S_x \cdot \Phi^*.$$  \hfill (5.178)

Generalized Cholesky exploits this interpretation. Cholesky factorization produces the MMSE linear-prediction estimate of $x_n$ from $\{x_{N-1}...x_0\}$ through $\Phi^{-1}$ through the successively computed errors $v_n$ (which are temporary to this section and not elements of a normalized $v$), or equivalently an MMSE estimate direction from the values of $v_n$ with $\Phi$:

$$x_n = v_n + g_{n-1} \cdot v_{n-1} + \ldots + g_{n,0} \cdot v_0.$$  \hfill (5.179)

Continuing with the MMSE interpretation, the rightmost $N^*$ positions in $R_{xx}$’s upper $N_x - N^*$ rows are $R_{\bar{x} x}$. These rightmost positions essentially define a $x_\nu$ that depends on $x_\nu$ through a linear combination that Generalized Cholesky finds. This positions constitute the $(N_x - N^*) \times N^*$ cross-correlation matrix between $x_\nu$ and $x_{\bar{\nu}}$. This cross-correlation matrix helps construct MMSE estimates, namely $\hat{x}_\nu = R_{[x_\nu]v} \cdot S_x^{-1} \cdot v$. Then, the MMSE estimator (and discrete modulator) for the nonsingular case is

$$A = \begin{bmatrix} R_{x_\nu x_\nu} \cdot \Phi^{-*} \cdot S_x^{-1} \\ \Phi \\ G_x \end{bmatrix} = \begin{bmatrix} \hat{\Phi}_x \\ \Phi_x \\ G_x \end{bmatrix},$$  \hfill (5.180)

which is generalized triangular, so $x = A \cdot v$ and

$$R_{xx} = A \cdot S_x \cdot A^*.$$  \hfill (5.181)

The discrete modulator can generate $x$ “causally” with the last $N_x - N^*$ dimensions depending exclusively only on previous inputs $v$ as in Figure 5.13. The diagonal factor $S_x^{1/2}$ can subsequently be absorbed into $H$ upon return to GDFE design.

Figure 5.13 and this process presumes re-ordering has already occurred prior to defining time for causality. A recursive implementation is always possible even if non-causality is imposed by the re-ordering. The following example illustrates the basic procedure.

31The correct upper triangular matrix is not directly computed in Matlab. The Matlab command to compute this desired upper form is $G_{\bar{x}} = \text{chol}(J R_{u u} J)$, where $J$ has all ones down the anti-diagonal and zeros elsewhere. Pre- and post-multiplication of the resultant matrix by $J$, and then taking conjugate transpose produces intermediate upper triangular matrix $G_{\bar{x}}$. Finally, removal of diagonal terms by post multiplication generates $G_x = G_{\bar{x}} \cdot \text{inv}(\text{diag}(\text{diag}(G_{\bar{x}})))$ or thus obtains the $\Phi$ used in this text. $S_x$ is $\text{diag}(\text{diag}(G_{\bar{x}}))^2 \text{diag}(\text{diag}(G_{\bar{x}}))$. For matlab command examples, see Examples 5.2.3, 5.3.3, and 5.3.4.
EXAMPLE 5.2.3 [Generalized Cholesky and GDFE for $1 + .9D^{-1}$ Channel.] The singular $R_{xx'}$ for the channel's pass space with white input was previously found in Example 5.2.2 as

$$R_{xx'} = \begin{bmatrix} .5945 & .3649 & -.3285 \\ .3649 & .6715 & .2956 \\ -.3285 & .2956 & .7340 \end{bmatrix} . \quad (5.182)$$

This matrix becomes the desired input autocorrelation. Then Generalized Cholesky proceeds as follows:

\[
\begin{align*}
\text{>> } & H = (1/\sqrt{.181}) \cdot \begin{bmatrix} .9 & 1 & 0 \\ 0 & .9 & 1 \end{bmatrix} = 2.1155 \ 2.3505 \ 0 \\
& 0 \ 2.1155 \ 2.3505 \\
\text{>> } & [F, L, M] = svd(H) \\
F &= \begin{bmatrix} -0.7071 \\ -0.7071 \end{bmatrix} \quad L = \begin{bmatrix} 3.8694 & 0 & 0 \\ 0 & 2.4222 & 0 \end{bmatrix} \quad M = \begin{bmatrix} -0.3866 & -0.6671 & 0.6368 \\ -0.8161 & -0.0741 & -0.5731 \\ -0.4295 & 0.7412 & 0.5158 \end{bmatrix} \\
\text{>> } & M_1 = M(1:3, 1:2) = \\
& \begin{bmatrix} -0.3866 & 0.6671 \\ -0.8161 & 0.0741 \\ -0.4295 & -0.741 \end{bmatrix} \\
\text{>> } & rvxtt = M_1 \cdot M_1' = \\
& \begin{bmatrix} 0.5945 & 0.3649 & -0.3285 \\ 0.3649 & 0.6715 & 0.2956 \\ -0.3285 & 0.2956 & 0.7340 \end{bmatrix} \\
\text{>> } & \Phi = \text{lohc}(rvxtt(2:3, 2:3)) = \\
& \begin{bmatrix} 0.7433 & 0.3451 \\ 0 & 0.8567 \end{bmatrix} \\
\text{>> } & \Phi_1 = \Phi \cdot \text{inv(diag(diag(Phi)))} = \\
& \begin{bmatrix} 1.0000 & 0.4028 \\ 0 & 1.0000 \end{bmatrix} \\
\text{>> } & S_x = \text{diag(diag(Phi))}^2 = \\
& \begin{bmatrix} 0.5525 & 0 \\ 0 & 0.7340 \end{bmatrix} \\
\text{>> } & A = [rvxtt(1, 2:3) \cdot \text{inv(Phi')} \cdot \text{inv(S_x)} \\
& \Phi] = \\
& \begin{bmatrix} 0.9000 & -0.4475 \\ 1.0000 & 0.4028 \\ 0 & 1.0000 \end{bmatrix} \\
\text{>> } & A \cdot S_x \cdot A' \text{ (another check)} = \\
& \begin{bmatrix} 0.5945 & 0.3649 & -0.3285 \\ 0.3649 & 0.6715 & 0.2956 \\ -0.3285 & 0.2956 & 0.7340 \end{bmatrix} \\
\text{>> } & A = A \cdot \text{sqrtm(S_x)} = \\
& \begin{bmatrix} 0.6690 & -0.3834 \\ 0.7433 & 0.3451 \\ 0 & 0.8567 \end{bmatrix} \\
\text{>> } & [\text{snrGDFEu, GU, WU, S0, MSWMFU, b, bbar}] = \text{computeGDFE}(H, A, 2, 3) \\
\text{snrGDFEu} = 5.5427 \text{ dB} \\
\text{GU} = \\
& \begin{bmatrix} 1.0000 & 0.3459 \\ 0 & 1.0000 \end{bmatrix} \\
\text{WU} = 
\end{align*}
\]
This is the same value as obtained previously. The input matrix $A$ is generalized triangular.

### 5.2.5 The Tonal GDFE

Section 4.7’s Vectored DMT/OFDM becomes $N L_x$-dimensional GDFEs on each tone’s $L_x$ “spatial” dimensions, again presuming all GDFEs’ full synchronization to the same symbol and sampling clocks. Section 4.7 found that each (possibly complex) Vector-DMT/Output channel-output dimension has vector form

$$Y_n = \tilde{H}_n \cdot X_n + N_n, \quad n = 0, ..., N - 1.$$  

(5.183)

The transmission system decomposes into Figures’ 5.14 (transmitter) and 5.15’s (receiver) “tonal GDFE’s.” $H_n$ is not in general cyclic, nor nonsingular, but the general GDFE theory applies to it, independently for each $n$. 

### Figure 5.14: Tonal GDFE Transmitter.

...
Singularity can often be avoided by design (each tone’s exclusion of input dimensions in the channel model that don’t pass to the tone’s channel output, i.e. that are in the null-space \( \mathcal{N}_n \) for that tone \( n \)). Thus, a tonal GDFE is a set of up to \( \sqrt{N} \) separate VC GDFE’s, one for each tone. Tonal GDFE’s dominant complexity is \( L_x^2 \), while the FFT’s used are \( N \cdot \log_2(N) \) for each user. Thus total complexity per symbol is the order of \( \frac{(N \cdot L_x^2 + L_x \cdot \sqrt{N} \log_2(N))}{\sqrt{N}} = L_x^2 + \log_2(N) \), which remains low for up to say \( L_x \approx \log_2(N) \) spatial dimensions or less. This is usually a large complexity reduction from the general case of \( (N \cdot L_x)^2 \). Such tonal GDFE systems appear in “Massive MIMO” in wireless, as in Sections 2.7 and 2.8. Tonal GDFE’s also find use in wireline cables of crosstalking transmission lines.

For each tone, the designer has the choice of using either \( G = I \) (vector coding) or \( G \neq I \) (triangular GDFE). The triangular GDFE may be necessary because vector coding (with \( G = I \)) requires coordination (i.e., the vector-coding matrix \( M_n \)) on the transmit side, which may not be physically possible (for instance, Chapter 2’s MAC). Thus a triangular GDFE for the square \( H_n \) may be the only feasible canonical implementation.

### 5.2.5.1 Tonal GDFE Calculation: using the correct noise-whitened channel equivalent

The Gaussian MAC often has sampled-time-domain specification \( H(D) = h_0 + h_1 \cdot D + h_2 \cdot D^2 + \ldots + h_v \cdot D^v \). Negative powers on \( D \) mean “non-causal” (realized with delay), so no generality is lost by starting at time 0. The \( h_k \) are the \( L_y \times L_x \) transfer responses in sampled time, indexed by \( k \). The following two matlab commands synthesize the tonal channels

\[
\begin{align*}
h &= \text{cat}(3, h_0, h_1, \ldots, h_{nu});
H &= \text{fft}(h, N, 3)
\end{align*}
\]

The output \( H \) is a \( L_y \times L_x \times \sqrt{N} \) tensor where Matlab’s \( H(:,:,n) \) would correspond to \( \tilde{H}_n \) in (5.183). A GDFE exists for each such tonal channel.

**Why is there not a \( \sqrt{1/N} \) factor in front of matlab’s unnormalized FFT command?**

The \( 1/\sqrt{N} \) factor that would cause Matlab’s FFT command to correspond to a unitary (squared magnitude preserving) transform does not appear in the above matlab equations. This is correct for a noise-whitened channel, although the reason may not be immediately evident: With fixed sampling rate, \( 1/T' \), the symbol period increases with \( N \) as \( T = N \cdot T' \). Thus the signal and noise energy both increase by a factor of \( \sqrt{N} \) over the longer time period. A random process that has energy 1 per symbol will instead have energy increased by the factor of increase in \( N \), so if the original system at sampling rate had unit energy for the process, the longer-symbol system now has energy \( \sqrt{N} \). Program inputs should therefore increase input energy/symbol by \( \sqrt{N} \) to be correct (energy per sample inputs remain the same).
However, with noise-whitening, there is no explicit noise-energy input. There is a tacit discrete-time convolution of the sampled channel and the noise whitening filter. The product of normalized FFT’s then overnormalizes the channel and thus the output must be scaled up by $\sqrt{N}$. Matlab’s FFT does not need this scaling correct, but again is not unitary; the unitary transform is useful to communication designers who use it to preserve input energy/power before/after the modulator processing. This is why the Matlab FFT above is used without scaling because it convolution with it requires no correction when FFT’s are multiplied in succession to correspond to convolution. The following example helps illustrate this effect.

**EXAMPLE 5.2.4**  
*Computation of a Tonal GDFE Set*  
A complex baseband-channel matrix AWGN channel with $\sigma^2 = .01$ has

$$H(D) = \begin{bmatrix} 1 + D & -.5 - .4D \\ .9 - .3D & 1 - .9D \end{bmatrix}$$

has

$$h_0 = \begin{bmatrix} 1 & -.5 \\ .9 & 1 \end{bmatrix} \quad \text{and} \quad h_1 = \begin{bmatrix} 1 & -.4 \\ -.3 & -.9 \end{bmatrix} \quad (5.184)$$

With 8 tones, the matlab sequence would be

```matlab
>> h0=[ 1 -.5 
    .9 1];
>> h1=[1 -.4 
    -.3 -.9];
>> h= cat(3, h0 , h1)
>> h
h(:,:,1) =
    1.0000   -0.5000
    0.9000    1.0000
h(:,:,2) =
    1.0000   -0.4000
   -0.3000   -0.9000
>> N=8;
>> >> H=(10)*fft(h, N, 3)
H(:,:,1) =
    20.0000 + 0.0000i   -9.0000 + 0.0000i
    6.0000 + 0.0000i    1.0000 + 0.0000i
H(:,:,2) =
    17.0711 - 7.0711i   -7.8284 + 2.8284i
    6.8787 + 2.1213i   3.6360 + 6.3640i
H(:,:,3) =
    10.0000 -10.0000i   -5.0000 + 4.0000i
    9.0000 + 3.0000i   10.0000 + 9.0000i
H(:,:,4) =
    2.9289 - 7.0711i   -2.1716 + 2.8284i
  11.1213 + 2.1213i   16.3640 + 6.3640i
H(:,:,5) =
    0.0000 + 0.0000i    -1.0000 + 0.0000i
  12.0000 + 0.0000i   19.0000 + 0.0000i
H(:,:,6) =
    2.9289 + 7.0711i   -2.1716 - 2.8284i
  11.1213 - 2.1213i   16.3640 - 6.3640i
H(:,:,7) =
```

847
10.0000 +10.0000i -5.0000 - 4.0000i
9.0000 - 3.0000i 10.0000 - 9.0000i

H(:,:,8) =
17.0711 + 7.0711i -7.8284 - 2.8284i
6.8787 - 2.1213i 3.6360 - 6.3640i

The tonal channels are conjugate because the original (matrix) channel was real. Note \( n = 1 \) corresponds to DC because matlab has positive integer indices. \( n = 4 \) is the Nyquist tone.

GDFE designs would require a tonal discrete modulator which if it had one unit of energy/sample would reduce to \( N/(N + \nu) = 8/9 \) because of cyclic prefix. The commands:

\[
A = \text{zeros}(2,2,8);
\]
\[
\text{for } n=1:N
\]
\[
A(:,:,n) = \sqrt{8/9} \times \text{eye}(2);
\]
\[
\text{end}
\]

generate a spatially white input directly into each and every tone with energy/tone \( 8/9 \) instead of 1 because of the cyclic prefix energy loss. The computeGDFE function could be used to generate 8 GDFE's according to

\[
\text{cb}=1;
\]
\[
\text{Lx}=3;
\]
\[
\text{GU} = \text{zeros}(2,2,8);
\]
\[
\text{WU} = \text{zeros}(2,2,8);
\]
\[
\text{S0} = \text{zeros}(2,2,8);
\]
\[
\text{MSWMFU} = \text{zeros}(2,2,8);
\]
\[
\text{b} = \text{zeros}(2,1,8);
\]
\[
\text{bbar} = \text{zeros}(1,8);
\]
\[
\text{for } n=1:N
\]
\[
[\text{snrGDFEu}(1,n), \text{GU}(:,:,n), \text{WU}(:,:,n), \text{S0}(:,:,n), \text{MSWMFU}(:,:,n), b(:,:,n), bbar(n)] = \ldots
\]
\[
\text{computeGDFE}(H(:,:,n), A(:,:,n), \text{cb}, \text{Lx});
\]
\[
\text{end}
\]

\[
\text{>> snrGDFEu} =
\]
\[
\text{>> GU}
\]
\[
\text{GU}(:,:,1) =
1.0000 + 0.0000i -0.3991 + 0.0000i
0.0000 + 0.0000i 1.0000 + 0.0000i
\]
\[
\text{GU}(:,:,2) =
1.0000 + 0.0000i -0.2928 + 0.0737i
0.0000 + 0.0000i 1.0000 + 0.0000i
\]
\[
\text{GU}(:,:,3) =
1.0000 + 0.0000i 0.0931 + 0.1414i
0.0000 + 0.0000i 1.0000 + 0.0000i
\]
\[
\text{GU}(:,:,4) =
1.0000 + 0.0000i 0.9056 + 0.1552i
0.0000 + 0.0000i 1.0000 + 0.0000i
\]
\[
\text{GU}(:,:,5) =
1.0000 + 0.0000i 1.5833 + 0.0000i
0.0000 + 0.0000i 1.0000 + 0.0000i
\]
\[
\text{GU}(:,:,6) =
1.0000 + 0.0000i 0.9056 - 0.1552i
0.0000 + 0.0000i 1.0000 + 0.0000i
\]

848
GU(:,:,7) =
    1.0000 + 0.0000i  0.0931 - 0.1414i
    0.0000 + 0.0000i  1.0000 + 0.0000i

GU(:,:,8) =
    1.0000 + 0.0000i  -0.2928 - 0.0737i
    0.0000 + 0.0000i  1.0000 + 0.0000i

>> MSWMFU

MSWMFU(:,:,1) =
    0.0487 + 0.0000i  0.0146 + 0.0000i
   -0.0865 + 0.0000i  0.2821 + 0.0000i

MSWMFU(:,:,2) =
    0.0460 + 0.0191i  0.0186 - 0.0057i
   -0.0409 + 0.0060i  0.0705 - 0.0787i

MSWMFU(:,:,3) =
    0.0366 + 0.0366i  0.0329 - 0.0110i
   -0.0364 - 0.0175i  0.0476 - 0.0370i

MSWMFU(:,:,4) =
    0.0166 + 0.0402i  0.0632 - 0.0120i
   -0.0381 - 0.0564i  0.0431 - 0.0177i

MSWMFU(:,:,5) =
    0.0000 + 0.0000i  0.0884 + 0.0000i
   -0.2792 + 0.0000i  0.0411 + 0.0000i

MSWMFU(:,:,6) =
    0.0166 - 0.0402i  0.0632 + 0.0120i
   -0.0381 + 0.0564i  0.0431 + 0.0177i

MSWMFU(:,:,7) =
    0.0366 - 0.0366i  0.0329 + 0.0110i
   -0.0364 + 0.0175i  0.0476 + 0.0370i

MSWMFU(:,:,8) =
    0.0460 - 0.0191i  0.0186 + 0.0057i
   -0.0409 - 0.0060i  0.0705 + 0.0787i

The bit distribution and total bits for each tone are, and then total and per real dimension:

>> reshape(b,[2,8]) =

>> bbar =

>> sum(b,'all') = 111.2179
>> ans/18 = 6.1788

The overall geometric SNR is thus

>> 10*log10(2^(2*ans)-1) = 37.1991 dB

Note the factor of 2 × 9 to encompass the cyclic prefix of 1 sample (dimension) that is lost to implementation simplification. Also note the biased geometric SNR is the arithmetic average of the biased SNRgdir (not an output) in dB, but the unbiased is really the true measure and is not exactly the arithmetic mean in dB.
5.3 GDFE Transmit Optimization

The GDFE exists for any input autocorrelation matrix $R_{xx}$. There is one best $R_{xx}$ that maximizes the SNR_{GDFE}. This section revisits this best $R_{xx}$’s determination through finite-length symbol partitioning and subsequent water-fill loading. Subsection 5.3.1 returns to vector coding (VC) to compute the best $R_{xx}$ while also revisiting briefly VC’s parallel-channel set. Subsection 5.3.2 then constructs this best $R_{xx}$ using the GDFE’s triangular white input through Generalized Cholesky Factorization. Subsection 5.3.3 then specifically investigates GDFE loading and the inherent interpolation in the transmit signal construction.

Subsection 5.2.3’s Circulant DFE has the same performance as DMT when both have the same circulant $H$, $R_{nn}$, and $R_{xx}$. This CDFE can also converge to Chapter 3’s canonical MMSE-DFE set. Subsection 5.3.4 uses an easily designed DMT system to optimize CDFE performance through two methods – discrete-time sampling-rate interpolation and continuous-time sampling interpolation, finding both in best cases more complex than the original DMT system, but with the same performance at any block size $N$ and guard-period $\nu$. Chapter 3’s continuous-time symbol-rate and carrier-frequency optimization then transform into Subsection 5.3.3’s CDFE with that subsection’s generalized-Cholesky interpolation. This section largely considers only $L_x = L_y = 1$, although expansion to vector DMT or Chapter 3’s MIMO MMSE-DFEs is straightforward although notationally tedious.

5.3.1 The channel-dependent optimized input

Subsection 5.2.1 showed that VC is a special GDFE case where the set of $N^+$ parallel channels correspond to a zeroed feedback section, i.e., $G = I$. The forward and backward canonical channels always have the same mutual information, but since there is no feedback, both zero-forcing (forward) and MMSE (backward) have the same optimal performance on each of the scalar AWGNs generated (that is, ZF and MMSE are the same for VC because the subchannels are each independent scalar AWGNs for which zero-forcing and MMSE produce the same result). VC enables direct calculation of a water-filling spectrum, $R_{vv}$ must initially relax to be diagonal with variable energy/dimension, as in Chapter 4. The diagonal energy scaling then absorbs into the discrete matrix modulator $A$. With such a calculated $R_{xx}$, the design removes input singularity (and of course channel singularity, which does not depend on $R_{xx}$) and forms $R_{uu}$. The resultant nonsingular $R_{uu}$ then factors to $R_{uu} = F \cdot F^*$. From this $R_{xx}$, then $\tilde{H} = R_{nn}^{-1/2} \cdot H \cdot M \cdot S_1^{1/2}$ for vector coding, or from any $A$ such that $A \cdot A^* = M \cdot S_2 \cdot M^*$. The VC cases’ forward canonical channel has an input $v$ with components $\nu_n/\|\nu_n\|$ that has identity autocorrelation $R_{vv} = I$, $R_f = H^* \cdot H = M^* \cdot M \cdot A^* \cdot F \cdot F^* \cdot A \cdot M^* \cdot M = \lambda^2$ for any diagonal $R_{vv}$. The forward canonical channel model then is

$$z = R_f \cdot v + n' = \lambda^2 \cdot v + n' = \text{diag}(\lambda_n^2 \cdot \nu_n/\|\nu_n\| + n_n'),$$

(5.185)

where $n'$ has autocorrelation matrix $R_{nn} = M \cdot \lambda_n^2 \cdot M^*$. With $\mathcal{E}_n \overset{\Delta}{=} \mathbb{E} [\nu_n^2]$, each subchannel has $\text{SNR}_n = \lambda_n^2$, which tacitly contains the $\mathcal{E}_n$ energy distribution. Waterfilling thus finds a set of energies $\{\mathcal{E}_n\}_{n=0,...,2^{n-1}}$ from the original channel $R_{nn}^{-1/2} \cdot H$ and energy constraint $\mathcal{E}_n \leq \text{trace}(R_{xx})$. These energies are the diagonal elements of $S_x$ used to form $\tilde{H}$. The the new input has energy 1 unit per (real or complex) dimension. The backward channel can proceed then as always to be

$$R_b^{-1} = S_0 = R_f + I = \text{diag}(\text{SNR}_n + 1),$$

(5.186)

or

$$R_b = \mathcal{E} \cdot e = \text{diag}(\frac{1}{1 + \text{SNR}_n}) .$$

(5.187)

Equation (5.187) validates the relation concerning the (biased) SNR on each GDFE dimension

$$\text{SNR}_{GDFE,n} = R_b = \mathcal{E} \cdot e, \text{SNR}_{GDFE,n} = R_b \cdot e, \text{SNR} = 1 + \text{SNR}_n .$$

(5.188)

32Recalling that $\lambda_n$ includes the effect of the noise normalization in (5.159), the $g_n$ of Chapter 4’s loading discussion is $g_n = |\lambda_n|^2$ here. Further, there is a $\mathcal{E}_n^2$ in the signal energy gain, but $\mathcal{E}_n$ in the noise gain, leaving $\text{SNR}_n = \lambda_n^2$. 850
Each forward subchannel’s unbiased SNR in (5.185) is also the corresponding backward channel’s unbiased \( \text{SNR}_n \), which thus proves VC’s forward-and-backward channel equivalence. Thus, \( A = M’ \cdot S_{1/2} \), with \( M’ \) as the corresponding first \( N^* \) columns of \( M \). There is similarly an \( F’ \) that deletes any columns corresponding to zeroed singular values and/or energies.) The best VC structure thus exhibits a unique property within all possible GDFE designs: ZF and MMSE are the same in this very special case. Recall in Chapter 3 that this occurred with nonzero noise only when there is no ISI (\( Q(D) = 1 \)), while in VC it is occurring with ISI overall, but because the MMSE has no ISI component in the special case when channel singular vectors are the discrete modulator, \( A = M’ \).

The VC receiver, which does not have a feedback section, essentially just multiplies by \( \Lambda^{-1} (F’)^* \), which is equivalent to the combination of matched filtering by \( \tilde{H}^* \), feedforward filtering by \( S_{0^{-1}} \), and removing each dimension’s bias:

\[
\text{Diag} \left( \frac{\text{SNR}_n + 1}{\text{SNR}_n} \right) \cdot S_{0^{-1}} \cdot \tilde{H}^* = \text{(5.189)}
\]

\[
= \text{Diag} \left( \frac{\lambda^2_n + 1}{\lambda^2_n} \right) \cdot \text{Diag} \left( \frac{1}{\lambda^2_n + 1} \right) \cdot \Lambda \cdot F^* \text{(5.190)}
\]

\[
= \text{Diag} \left( \frac{1}{\lambda^2_n} \right) \cdot (\lambda_n) \cdot F^* \text{(5.191)}
\]

\[
= \Lambda^{-1} \cdot F^* . \text{(5.192)}
\]

This unbiased feedforward section then simply completes the parallelization into uncorrelated dimensions and scales each subchannel by \( 1/\lambda_n \) (the noise variance scales by \( 1/\lambda^2_n \) but the path from normalized input \( v \) to channel output has gain \( \lambda^4 \) so again \( \text{SNR} = \lambda^2 \)). VC’s subchannels usually carry different amounts of information: For any \( \mathcal{E}_n \) distribution, and of course including waterfill or any other set the designer so creates,

\[
\bar{b}_n = \frac{1}{2} \cdot \log_2 \left( 1 + \frac{\text{SNR}_n}{\Gamma} \right) \text{(5.193)}
\]

and \( \bar{b}_n = \bar{c}_n \) when \( \Gamma = 0 \) dB. Separation-Theorem invocation permits constant constellation size’s (not constant mutual information) use with a good code, as with C-OFDM and DMT. Exceptionally important is that in the DMT/VC case, the (unbiased MMSE) error on each subchannel contains no component of the signal, has no need of a feedback section, and ML detection simplifies greatly with only a linear modulator/demodulator.

### 5.3.1.1 nonzero Gap Inputs

Any GDFE’s optimum input code has \( \Gamma = 0 \) dB. However, VC/DMT systems have an optimum decoder for codes of any gap \( \Gamma > 0 \) dB because each subchannel is exactly an AWGN. When \( G \neq I \), as in the triangular GDFE, then the GDFE error vector contains a (non-Gaussian, since \( \Gamma > 0 \) dB) channel-input-based component. Correspondingly these alternative GDFEs create different SNR sets even though all minimize MMSE for the same input autocorrelation matrix \( R_{xx} \), channel \( H \), and noise \( R_{nn} \) and thus have the same mutual information \( I(x; y) \). The gap-dependent GDFE overall geometric SNR is

\[
\text{SNR}_{GDFE}(\Gamma) = \Gamma \cdot \left\{ \prod_{i=1}^{N^*} \left( 1 + \frac{\text{SNR}_n}{\Gamma} \right) \right\}^{1/N^*} - 1 \text{(5.194)}
\]

Since different choices of the input realization lead to different \( \text{SNR}_n \) sets, a nonzero gap causes \( \text{SNR}_{gdf/e} \) to vary with the particular set of \( \text{SNR}_n \)’s. All GDFE’s only have the same \( \text{SNR}_{gdf/e,n} = 2^{2\mathcal{T}(x, y)} - 1 \) only when \( \Gamma = 0 \) dB (or again good code within any Voronoi shaping region with gap equal to the shaping gain loss). In fact when \( \Gamma > 0 \) dB, the \( \text{SNR}_{GDFE,U} \) in (5.194) is the largest with the \( \text{SNR}_n \)’s of VC/DMT.
Theorem 5.3.1 [VC/DMT Optimality at nonzero Gap] Vector coding (which has a special case DMT when cyclic prefixes are used) has the highest SNR among all GDFE’s for the same matrix AWGN channel $\tilde{H}$ and input when the gap is nonzero, $\Gamma > 0$ dB.

proof: Because VC has no signal-error component, division by the gap measures each subchannel’s performance, as intended with the gap. The overall gap works in exactly the same way for the overall channel as it does for each VC/DMT subchannel. However, any SNR that has a nonzero MMSE signal-dependent component will have that component also experience multiplication by $\Gamma > 0$, which increases the MSE’s signal component (but the signal does not increase because the transmit power remains the same). Thus, the overall computed SNR in (5.194) will be too low on each subchannel, and indeed then the product of such terms is also lower than the exact $\text{SNR}_{\text{gdf},u} = \Gamma \cdot \left(2^Z(x,y) - 1\right)$ of the VC/DMT system. Further, VC/DMT is exactly an ML detector while the GDFE is not, guaranteeing at least slightly higher VC/DMT performance with $\Gamma > 0$ dB. QED.

Theorem 5.3.1 suggests the non-VC GDFE’s computed SNR under-estimates its performance when $\Gamma > 0$ dB; acknowledging that the gap is an approximation. However as the proof notes, because the VC/DMT detector is maximum likelihood, and since the SNR accurately represents exactly what is happening on each VC dimension, then no other detector could have better performance. The $\text{SNR}_{\text{gdf}}$ directly relates (via the gap) to VC’s constant symbol-error probability. However, for other GDFE’s with non-ideal codes, the error is not exactly Gaussian for nonzero gaps, and thus the detector is thus no longer exactly ML, and its performance is worse than a VC/DMT detector unequivocally (even if the SNR calculation is somewhat pessimistic and underestimates non-VC/DMT GDFE’s performance).

Larger gap also means a larger deviation in computed SNR, causing the underperformance magnitude to increase with $\Gamma$.

EXAMPLE 5.3.1 Section 4.6’s $1 + 0.9 \cdot D^{-1}$ DMT example for $N = 8$ and $\Gamma = 0$ dB had $\text{SNR}_{\text{DMT}} = 7.6$ dB, which exactly equals the CDFE SNR for that same channel and input in Section 5.2. However, if those examples are reworked with $\Gamma = 8.8$ dB, the two SNR’s are then

$$\text{SNR}_{\text{dmt}}(\Gamma = 8.8) = 9.5 \text{ dB} > \text{SNR}_{\text{cdf},u}(\Gamma = 8.8) = 8.6 \text{ dB ,} \quad (5.195)$$

a difference of almost 1 dB. These SNR’s are higher than the true SNR of 7.6 dB but do not represent higher performance because a gap of 8.8 dB essentially means performance is as if the noise were 8.8 dB larger than with capacity-achieving codes (so performance of the nonzero-gap DMT system would be 9.5-8.8 or 0.7 dB, which is below the earlier example’s 7.6 dB when $\Gamma = 0$ dB). Thus (5.195)’s 8.6 dB may be lower than a precise calculation, but overall CDFE performance still cannot exceed DMT’s 9.5 dB with $\Gamma = 8.8$ dB.

Coded-OFDM also will not perform as well as the DMT system, because Chapter 4’s Separation Theorem loses accuracy as $\Gamma$ increases. C-OFDM’s discrete modulator may use the same $C$ (with good code) on all channels. The ML receiver must weight better dimensions more heavily in decoding, which tends also to confuse the nonzero gap’s applicability (an area that awaits motivated study as yet).

5.3.2 GDFE SNR Maximization over $R_{xx}$

$R_{xx}$ design first optimizes for VC where $A = M$ and subchannel gains are $g_n = \frac{|\lambda_n|^2}{\sigma^2}$. GDFE optimization then maximizes $\text{SNR}_{\text{GDF},U}$

$$\text{SNR}_{\text{GDF},U} = 2^Z(x,y) - 1 \quad . \quad (5.196)$$

which has maximum with water-filling solution

$$E_n + \frac{1}{g_n} = K \quad , \quad (5.197)$$

852
where \( K \) is the constant determined by either the data rate (MA problem) or energy constraint (RA) problem. In the RA case,\footnote{\( \text{SNR}_{\text{GDFE.U}} = 2^2 \bar{C} - 1 \).} \( \text{SNR}_{\text{GDFE.U}} = 2^2 \bar{C} - 1 \), \eqref{eq:5.198}

where \( \bar{C} \) is the capacity or maximum value for \( \overline{I}(x; y) \) The number of bits on each subchannel is\footnote{\( \bar{b}_n \leq \bar{C}_n = \frac{1}{2} \log_2 (1 + \text{SNR}_n) \).} \[ \bar{b}_n \leq \bar{C}_n = \frac{1}{2} \log_2 (1 + \text{SNR}_n) \] \eqref{eq:5.199}

Again, the water-filling procedure determines \( N_* \), the number of energized input dimensions. The GDFE design’s input-singularity removal eliminates zero-energy dimensions from the input when constructing the canonical forward and backward channels.

The optimum VC autocorrelation is then (with \( m \) determined from the SVD of \( F \cdot \Lambda \cdot M^* = R^{-1/2} H \) without the input scaling included)

\[ R_{xx}^{\text{(opt)}} = R_{xx'}^{\text{(opt)}} = \sum_{n=1}^{N_*} E_n \cdot m_n \cdot m_n^* , \] \eqref{eq:5.200}

which tacitly reorders indices so that the energized dimensions are \( n = 1, ..., N_* \).

5.3.3 Triangular GDFE \( R_{xx} \) Optimization

To optimize the triangular GDFE, the designer first constructs \( M \) through SVD of the channel matrix \( R^{-1/2} H \), and then computes the water-filling dimensional energies \( E_n \), and constructs the optimum \( R_{xx} \) as in \ref{eq:5.200}. Other VC input energy distributions may also find use, and this subsection’s procedure applies to any nonsingular \( R_{xx} \). A non-VC’s triangular \( G \) implementation can admit recursive causal transmitter implementation from \( v \) to best \( R_{xx} \). Such an implementation may be of interest for several reasons, including recursive time-series-like implementation or for Chapter 2’s BC.

The objective is a relationship \ref{eq:5.201} where \( R_{vv} = I \), and where \( A \) is generalized triangular:

\[ x = A \cdot v , \] \ref{eq:5.201}

A generalized triangular matrix essentially interpolates in a step-by-step causal-like manner from a set of \( N_* \) white-input dimensions \( v_n \) to a larger set of \( N + \nu \) discrete-modulator output dimensions.

5.3.3.1 Reordering singular-autocorrelation dimensions

A first \( R_{xx} \)-optimization step reorders \( R_{xx} \)’s first \( N_* \) indices that correspond to the nonsingular information-bearing input dimensions in \( v \). The permutation matrix \( J_{i,j} \) differs from an identity matrix only in that the 1’s that would have been in the \( i^{th} \) row and \( i^{th} \) column and in the \( j^{th} \) row and \( j^{th} \) column appear instead in the \( i^{th} \) row and \( j^{th} \) column, and in the \( j^{th} \) row and \( i^{th} \) column, respectively. The product \( J_{i,j} \cdot x \) switches the position of the \( i^{th} \) and \( j^{th} \) elements of the vector \( x \). For example, with a 4-dimensional input

\[ J_{2,3} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \] \eqref{eq:5.202}

and \( J_{2,3} \cdot [1234]^* = [1324]^* \). \( J_{i,j} \) is symmetric \( J_{i,j} = J_{j,i}^* = J_{j,i}^2 \), unitary, and \( J_{i,j} \cdot J_{i,j}^* = I \). Therefore, or trivially, \( |J_{i,j}| = 1 \). If \( x' = J_{i,j} \cdot x \), then \( R_{xx} \cdot x' = J_{i,j} \cdot R_{xx} \cdot J_{i,j}^* \), and \( |R_{xx}| = |R_{xx} \cdot x'| \).
The submatrix \( R_{xx}(i) \), \( i = 1, \ldots, \tilde{N}_x - 1 \) is the \((i+1) \times (i+1)\) lower-right-hand-corner of \( R_{xx} \). The following algorithm reorders \( x \) ’s input dimension index so that its first \( N^* \) dimensions are non-singular, that is \(|R_{xx}(i)| > 0\) for \( i = 0, \ldots, \tilde{N}_x - 1 \).

1. initialize \( i = 0, \delta = 1, J_A = I_{\tilde{N}_x} \).
2. while \(|R_{xx}(i)| > 0\)
   (a) \( i \leftarrow i + 1 \)
   (b) \( J_A \leftarrow J_A \)
3. If \( i = N^* \), exit this algorithm
4. \( J_A \leftarrow J_{i,\tilde{N}_x-\delta} \cdot J_A, R_{xx} \cdot x' = J_{i,\tilde{N}_x-\delta} \cdot R_{xx} \cdot J_{i,\tilde{N}_x-\delta} \)
5. \( \delta \leftarrow \delta + 1 \)
6. Go to step 2

This process interpolates the remaining dimensions’ entries from the nonzero inputs and constructs the desired covariance for the new \( R_{xx} \). Those last \( \tilde{N}_x - N^* \) dimensions are linearly dependent on the first \( N^* \) dimensions. The first \( N^* \) components are denoted \( x_v \) and the rest are \( x_{\bar{v}} \) so

\[
x = \begin{bmatrix} x_{\bar{v}} \\ x_v \end{bmatrix}.
\]

Any discrete modulator \( A \) for this newly ordered input must restore order in a final \( A \rightarrow J_A \cdot A \), that is the resulting order matrix \( J_A \) is absorbed into \( A \) for all further GDFE calculations,

\[
A \leftarrow J_A \cdot A.
\]

### 5.3.4 CDFE with Causal Triangular input

The optimum CDFE spectrum derives from water-filled DMT system with transmit IDFT matrix \( Q^* \) and tonal energy distribution \( E_n \). After sampling-rate adjustment (in each band if there are multiple bands) so that water-filling energizes all tones/dimensions, then this input becomes nonsingular. The designer may then proceed with direct application of Section 5.2.3’s triangular or circular GDFEs with this nonsingular water-fill input. Example 5.3.4 later in this subsection takes this approach. The resampling occurs independently for each discontiguous frequency band, which can require complex implementation. A simpler approach uses a digital-interpolation technique that separates the \( A \)-matrix “modulation” into a carrier-frequency-and-interpolation part (which is never triangular) and a “causal” (triangular) part. This method finds Chapter 3’s independent CDFE bands by essentially extracting those same bands from a DMT design of the same block-length. This subsection first addresses that digital-interpolation approach.

The optimum CDFE first finds a good \( N \times N \) autocorrelation matrix \( R_{xx} \) for a cyclic channel \( H \) from a DMT system,\(^{33}\)

\[
R_{xx} = Q_N^* \cdot R_{XX} \cdot Q_N,
\]

where \( R_{XX} \) is an \( \tilde{N} \times \tilde{N} \) diagonal matrix containing the DMT input energies on the diagonal. The matrix \( Q_N \) is \( \tilde{N} \times \tilde{N} \) and implements an FFT. For real-baseband DMT, then (5.205) has \( \tilde{N} \) replaced by \( N = 2\tilde{N} \) but the diagonal matrix \( R_{XX} \) is such that \( R_{XX}(i, i) = R_{XX}(N - i, N - i) \) to ensure real time-domain outputs. These vectors \( X \) and \( x \) reorder in frequency or time from smallest index 0 at the bottom to highest index \( \tilde{N} = \tilde{N}_x \) at the top. Because of the DMT system used in design and frequency-index ordering, there arise \( M \geq 1 \) bands that each has ALL nonzero entries, or equivalently

\(^{33}\) This procedure works for any DMT energy distribution, not just limited to, but of course including, water-filling. Also, \( \tilde{N} = \tilde{N}_x - \nu \) for the CDFE.
set of adjacent nonzero-energy frequency indices for $R_{XX}$. In fact, each such band has $\bar{N}_i$, $i = 1, \ldots, M$ nonzero input energies and

$$\bar{N}^* = \sum_{i=1}^{M} \bar{N}_i .$$  \hspace{1cm} (5.206)

The input’s information-bearing part contains nonzero tone inputs $\vec{X}$, which are interpolated from $\bar{N}^*$ used tones to $\bar{N}$ total tones according to

$$\vec{X} = J_g \cdot \vec{X} ,$$  \hspace{1cm} (5.207)

where $J_g$ has the following structures:

**Complex baseband $x$:** $J_g$ is an $\bar{N} \times \bar{N}^*$ permutation-and-decimation matrix\(^{35}\) with no more than one nonzero unit-value entry in each row (and all zeros in $\bar{N} - \bar{N}^*$ of the rows) with structure

$$J_g = \begin{bmatrix} 0_{\bar{N}_z, M+1 \times \bar{N}_M} & 0_{\bar{N}_z, M+1 \times \bar{N}_{M-1}} & \cdots & 0_{\bar{N}_z, M+1 \times \bar{N}_1} \\ I_{\bar{N}_M \times \bar{N}_M} & 0_{\bar{N}_M \times \bar{N}_{M-1}} & \cdots & 0_{\bar{N}_M \times \bar{N}_1} \\ 0_{\bar{N}_z, M \times \bar{N}_M} & 0_{\bar{N}_z, M \times \bar{N}_{M-1}} & \cdots & 0_{\bar{N}_z, M \times \bar{N}_1} \\ 0_{\bar{N}_z, M-1 \times \bar{N}_M} & I_{\bar{N}_{M-1} \times \bar{N}_{M-1}} & \cdots & 0_{\bar{N}_z, M-1 \times \bar{N}_1} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{\bar{N}_z, 1 \times \bar{N}_M} & 0_{\bar{N}_z, 1 \times \bar{N}_{M-1}} & \cdots & I_{\bar{N}_1 \times \bar{N}_1} \\ 0_{\bar{N}_z, 1 \times \bar{N}_M} & 0_{\bar{N}_z, 1 \times \bar{N}_{M-1}} & \cdots & 0_{\bar{N}_z, 1 \times \bar{N}_1} \end{bmatrix} ,$$  \hspace{1cm} (5.208)

**Real baseband case:** $J_g$ is a more complicated $N \times N^*$ permutation/decimation matrix. Some additional definitions are necessary. First, only band 1 can contain the single-real-dimensional DC frequency and only band $M$ can contain the single-dimensional Nyquist frequency. The remaining bands will all have $N_m$ nonzero real dimensions and $N_{z,m}$ zeroed real dimensions, $m = 2, \ldots, M-1$. Each of these bands has corresponding images for dimensions $n > N/2$ that have the same number of zeroed and nonzero real dimensions. However, band $m = 1$ will have $N^+_z$ nonzero real dimensions and $N^-_z$ nonzero real dimensions, one of which will include DC or $n = 0$. Thus, when DC carries no information (whence the superscript of + for “positive frequencies”), there will be $N^-_{z,1} = N^+_z - 1$ image zeros, and otherwise $N^-_{z,1} = N^+_z$ image-zeroed real dimensions. Similarly $N^-_{1,1}$ will be the number of nonzero image real dimensions and equal to $N^+_1 - 1$ when DC carries information and equal to $N^+_1$ when DC carries no information. Band $M$ can include Nyquist, either with zero or nonzero energy: this leads to $N^-_{z,M} = N^+_z + 1$ when Nyquist carries no information and $N^-_{z,M} = N^+_z$ otherwise. Similarly, $N^-_{M} = N^+_M + 1$ when Nyquist does carry information and $N^-_{z,M} = N^+_z$ otherwise. With these definitions $J_g$ for the real baseband case becomes

$$J_g = \begin{bmatrix} \begin{bmatrix} 0_{N^-_{z,1 \times N^-_{1,1}}} & 0_{N^-_{z,1 \times N^-_{2,1}}} & \cdots & 0_{N^-_{z,1 \times N^-_{M,1}}} \\ I_{N^-_{1,1 \times N^-_{1,1}}} & 0_{N^-_{1,1 \times N^-_{2,1}}} & \cdots & 0_{N^-_{1,1 \times N^-_{M,1}}} \\ 0_{N^-_{z,2 \times N^-_{1,1}}} & 0_{N^-_{z,2 \times N^-_{2,1}}} & \cdots & 0_{N^-_{z,2 \times N^-_{M,1}}} \\ 0_{N^-_{2,1 \times N^-_{1,1}}} & I_{N^-_{2,1 \times N^-_{2,1}}} & \cdots & 0_{N^-_{2,1 \times N^-_{M,1}}} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{N^-_{N^-_{z,M-1} \times N^-_{1,1}}} & 0_{N^-_{N^-_{z,M-1} \times N^-_{2,1}}} & \cdots & I_{N^-_{N^-_{z,M-1} \times N^-_{M,1}}} \\ 0_{N^-_{M-1 \times N^-_{1,1}}} & 0_{N^-_{M-1 \times N^-_{2,1}}} & \cdots & 0_{N^-_{M-1 \times N^-_{M,1}}} \end{bmatrix} & 0_{N \times (N^+_1 + N^+_M + \sum_{m=2}^{M-1} N_m)} \\ 0_{N \times (N^+_1 + N^+_M + \sum_{m=2}^{M-1} N_m)} & J_g^+ \end{bmatrix} ,$$  \hspace{1cm} (5.209)

where $J_g^-$ and $J_g^+$ are respectively defined by

$$J_g^- = \begin{bmatrix} 0_{N^-_{z,1 \times N^-_{1,1}}} & 0_{N^-_{z,1 \times N^-_{2,1}}} & \cdots & 0_{N^-_{z,1 \times N^-_{M,1}}} \\ I_{N^-_{1,1 \times N^-_{1,1}}} & 0_{N^-_{1,1 \times N^-_{2,1}}} & \cdots & 0_{N^-_{1,1 \times N^-_{M,1}}} \\ 0_{N^-_{z,2 \times N^-_{1,1}}} & 0_{N^-_{z,2 \times N^-_{2,1}}} & \cdots & 0_{N^-_{z,2 \times N^-_{M,1}}} \\ 0_{N^-_{2,1 \times N^-_{1,1}}} & I_{N^-_{2,1 \times N^-_{2,1}}} & \cdots & 0_{N^-_{2,1 \times N^-_{M,1}}} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{N^-_{N^-_{z,M-1} \times N^-_{1,1}}} & 0_{N^-_{N^-_{z,M-1} \times N^-_{2,1}}} & \cdots & I_{N^-_{N^-_{z,M-1} \times N^-_{M,1}}} \\ 0_{N^-_{N^-_{z,M-1} \times N^-_{1,1}}} & 0_{N^-_{N^-_{z,M-1} \times N^-_{2,1}}} & \cdots & 0_{N^-_{N^-_{z,M-1} \times N^-_{M,1}}} \end{bmatrix} ,$$  \hspace{1cm} (5.210)

\(^{34}\)the total number of real dimensions is, as always consistently, $N_i = 2\bar{N}_i$, twice the number of complex tones.  
\(^{35}\)J\(_{g}^\ast\) reorders and interpolates.
and

\[ J_g^+ = \begin{bmatrix}
0_{N_M^+ \times N_M^+} & 0_{N_M^+ \times N_{M-1}} & \ldots & 0_{N_M^+ \times N_1^+} \\
I_{N_M^+ \times N_M^+} & 0_{N_M^+ \times N_{M-1}} & \ldots & 0_{N_M^+ \times N_1^+} \\
0_{N_{M-1}^+ \times N_M^+} & 0_{N_{M-1}^+ \times N_{M-1}} & \ldots & 0_{N_{M-1}^+ \times N_1^+} \\
0_{N_{M-1}^+ \times N_M^+} & I_{N_{M-1} \times N_{M-1}} & \ldots & 0_{N_{M-1}^+ \times N_1^+} \\
\vdots & \vdots & \ddots & \vdots \\
0_{N_1^+ \times N_M^+} & 0_{N_1^+ \times N_{M-1}} & \ldots & I_{N_1^+ \times N_1^+} \\
0_{N_1^+ \times N_M^+} & 0_{N_1^+ \times N_{M-1}} & \ldots & 0_{N_1^+ \times N_1^+}
\end{bmatrix}. \tag{5.211}

5.3.4.1 IFFT matrices

Since the zero-energy tones between the bands carry no information, a nonsingular input \( u \) can be associated with only the information-carrying (nonzero input) tones in two cases as:

- **complex case** \( M \) size-\( \tilde{N}_i \) IFFT's \( Q_{\tilde{N}_i}^\ast \) or
- **real case** \( M \) size-\( \tilde{N}_i \) conjugate-symmetric IFFT’s.

A block FFT matrix operates on each of the non-singular input components individually:

**Complex DMT case:** The entire set of non-singular-part \( X \) components can be represented by

\[
\tilde{X} = \begin{bmatrix} X_M \\ X_{M-1} \\ \vdots \\ X_1 \end{bmatrix} = \begin{bmatrix} Q_{\tilde{N}_M}^\ast & 0 & \ldots & 0 \\
0 & Q_{\tilde{N}_{M-1}}^\ast & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & Q_{\tilde{N}_1^+}^\ast \end{bmatrix} \begin{bmatrix} u_M \\ u_{M-1} \\ \vdots \\ u_1 \end{bmatrix} = \tilde{Q} \cdot u. \tag{5.212}
\]

**Real baseband-DMT case:** The additional subdivision of an FFT matrix is necessary as

\[
Q_{\tilde{N}_i} = \begin{bmatrix} Q_{\tilde{N}_i}^+ \\ Q_{\tilde{N}_i}^- \end{bmatrix}. \tag{5.213}
\]

In the real case, band 1’s matrix \( Q_{\tilde{N}_1}^+ \) is a special case where DC may or may not carry nonzero energy. With nonzero DC energy, no alteration is necessary (and \( N_1 \) is necessarily odd). When DC has zero energy (meaning \( N_1 \) is even), the corresponding non-singular time-domain input \( u_k \) adjusts through multiplication by \( e^{j \frac{2\pi}{N_1} k} \) so

\[
Q_{\tilde{N}_1} \rightarrow Q_{\tilde{N}_1} \cdot \text{diag} \left[ e^{j \frac{2\pi}{N_1} (N_1-1)} \ldots e^{j \frac{2\pi}{N_1} (1)} \right]. \tag{5.214}
\]

Similarly Band M’s matrix \( Q_{\tilde{N}_M}^- \) is a special case where Nyquest may or may not carry nonzero energy. When Nyquist carries zero energy, implying even \( N_M \), then no offset is necessary. With nonzero Nyquist energy (implying that \( N_M \) is odd), the alteration multiplies the corresponding nonsingular time-domain input \( u_k \) by \( e^{j \frac{2\pi}{N_M} k/2} \) (an offset of half a carrier) so

\[
Q_{\tilde{N}_M} \rightarrow Q_{\tilde{N}_M} \cdot \text{diag} \left[ e^{j \frac{2\pi}{N_M} (N_M-1)} \ldots e^{j \frac{2\pi}{N_M} (1)} \right]. \tag{5.215}
\]
With these possibly modified baseband transforms, then the (correspondingly modified) matrices corre-
sponding to (5.213) become components of the formation

$$\tilde{X} = \begin{bmatrix} X_1^- & X_2^- & \cdots & X_M^- \\ X_1^+ & X_2^+ & \cdots & X_M^+ \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & Q_{N_1} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & Q_{N_M} & \cdots & 0 \\ 0 & 0 & \cdots & Q_{N_1} \end{bmatrix} \begin{bmatrix} u_M \\ u_{M-1} \\ \vdots \\ u_1 \end{bmatrix} = \tilde{Q}u \cdot . \quad (5.216)$$

Also, then for both real and complex cases, understanding that both $J_g$ and $\tilde{Q}$ are different for these
cases,

$$X = J_g \cdot \tilde{X} = J_g \cdot \tilde{Q} \cdot u \cdot . \quad (5.217)$$

**Autocorrelation formation in complex case:** For the complex case, Equation (5.217) leads to a
computation $u$’s autocorrelation matrix as

$$R_{uu} = \tilde{Q}^* \cdot \tilde{X} \tilde{X} \cdot \tilde{Q} \cdot . \quad (5.218)$$

**Autocorrelation formation in the real baseband case:** There is some regrouping of positive and
negative frequencies together for each of the bands in forming $R_{uu}$ terms. In either the real or complex
cases, if $u$ is a stacked vector of $N_i$ baseband-equivalent and decimated time-domain contributions of
the different bands. Those contributions are $u_i$, $i = 1, \ldots, M$, and

$$R_{uu} = \begin{bmatrix} R_{uu}(M) & 0 & \cdots & 0 \\ 0 & R_{uu}(M-1) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & R_{uu}(1) \end{bmatrix}. \quad (5.219)$$

In the real case, it is easier from a notational standpoint to compute the $R_{uu}(i)$ for the $i^{th}$ band
according to

$$R_{uu}(i) = \tilde{Q}_{N_i}^* \cdot \begin{bmatrix} R_{\tilde{X}\tilde{X}}(i) & 0 \\ 0 & R_{\tilde{X}\tilde{X}}(i) \end{bmatrix} \cdot \tilde{Q}_{N_i} \cdot . \quad (5.220)$$

**Discrete Modulator:** The relationship between the modulated channel input $x$ and $u$ is

$$x = Q^* \cdot X = Q^* \cdot J_g \cdot \tilde{Q} \cdot u \cdot . \quad (5.221)$$

The $M$ corresponding summed components of $x$ could be called $x_i$, each corresponding to a $u_i$. Then

$$x = \sum_{i=1}^{M} x_i \cdot , \quad (5.222)$$

where $x_i$ forms from all other components of $u_{j \neq i} = 0$. Each of these bands’ baseband equivalent
could appear stationary if considered by themselves. The multiplication of $u$ by the matrix $Q^* \cdot J_g \cdot Q$
essentially performs carrier modulation and interpolation (including the construction of the baseband
equivalent). The minimal number of bands is $M$, but it is possible to subdivide a band into adjacent
subbands and execute the same procedure. In the limit, if each band is subdivided into its constituent
DMT frequencies used in water-filling, the entire modulation reduces to DMT.
The full $A$ Matrix: The non-white inputs $u_i$ relate to white inputs $v_i$ by the usual Cholesky factorization

$$R_{uu} = \Phi \cdot \Phi^* ,$$

or

$$R_{uu}(i) = \Phi(i) \cdot \Phi^*(i) ,$$

decomposing the Cholesky factorization into $M$ such factorizations. $\Phi$ is block diagonal with each block upper triangular. The blocks are of different sizes when the $N_i$ are different. The matrix $A$ for CDFE construction is then

$$A = \underbrace{Q^* \cdot J_f \cdot \tilde{Q}}_{f_c \text{-mod}} \cdot \Phi \begin{cases} \text{causal} \\ \text{bb-equiv} \end{cases} .$$

The $\Phi$ will converge to stationary filters in each block-Cholesky factors’ middle rows as the individual block lengths $N_i \to \infty$. These are the baseband-equivalent transmit filters of the corresponding $M$ MMSE-DFE’s, each effectively operating at a sampling rate equal to the optimized symbol rate for that MMSE-DFE band.

Any nonsingular input with transmit matrix $Q$ and $E_n \neq 0 \forall n = 1, ..., N$ can be used to construct an acceptable (PWC-satisfying) CDFE input. Such an input necessarily will have a circulant $R_{xx} = \ldots$
\( Q^* \cdot \text{diag}(E_n) \cdot Q \) (the IDFT in this form always produces a circulant matrix). Thus, the CDFE channel input \( R_{xx} \) is circulant, and the resultant \( x \) represents a sum of contributions from the different bands. As \( N \to \infty \), all the diagonal terms of \( E_n \to E \) will become constant, and the constant rows of \( \Phi \to G(D) \) become each bands’ DFE feedback section.

\[
\begin{align*}
\frac{1}{T} &= N + \nu \\
e^{-j2\pi f_c t} &\text{ (carrier phase)} \\
y(t) &\xrightarrow{\text{DAC}} \text{Cyclic strip} \xrightarrow{\text{FFT}} y_N \\
y_N^* &\xrightarrow{J_g} y_M \\
y_M &\xrightarrow{\text{IFFT}} \psi_M \\
\end{align*}
\]

Figure 5.17: Comparison of decimated CDFE receiver structure for multiple bands with infinite-length multi-MMSE DFE receiver. (IFFT’s for Bands 1 and \( M \) may contain rotation factor for zero DC or nonzero Nyquist in real baseband case.)

**Carrier Frequencies and Symbol Rates:** Then for complex systems, each band’s actual carrier frequencies of each band are (where \( \Delta_i \) is the first nonzero tone index of the \( i^{th} \) band)

\[
f_{c,i} = f_c + \frac{2\pi}{NT} \cdot \left( \frac{N_i}{2} + \Delta_i \right), \quad (5.226)
\]

The symbol rate for each band is

\[
\frac{1}{T_i} = \frac{N_i}{N \cdot T}. \quad (5.227)
\]

Figures 5.16 and 5.17 compare the CDFE transmitter and receiver respectively with the corresponding limiting structures of the multi-band MMSE-DFE. Note that the \( H \cdot R_{nn}^{-1} \) is cyclic for white noise, and approximately cyclic for non-white noise with large \( N \). Cyclic matrices commute, but the combination of \( J_g \) and the smaller DFT matrices only become effectively cyclic as \( N \to \infty \).
EXAMPLE 5.3.2 [Cellular’s Uplink] Cellular’s uplink reuses a CDFE structure. The CDFE transmitter structure is as in Figure 5.16, where the sizes of the sub-blocks are cellular’s 12-tone resource blocks. Each uplink transmit IFFT operates on an integer multiple of 12 input subsymbols (tones). The carrier spacing is $\Delta f = 15$ kHz, and the same number of bits is used on each of the 12 tones (so 2, 4, 6, 8, ... bits/tone).

Cellular’s different uplink users use these different 12-tone groups, but share the same symbol clocks and carrier frequencies (they are synchronized to common clocks at the base station). DSL’s zippering used in DSL is not needed in cellular systems that are either frequency-division or time-division multiplexed so that systems with different clocks are separated greatly in frequency or time for uplink and downlink.

Thus, the cellular uplink transmitter has both a $k \cdot 12$ FFT and a larger full IFFT following the interpolation matrix $J_g$ that inserts zeros. Many such synchronized uplink transmitters will have zeros in different places and may use an appropriate IFFT simplification if many of the inputs are zeroed to it (wherever they don’t energize a resource block).

The name single-carrier OFDM is somewhat inappropriate in that neither is quite right. The system is circulant and uses basically the CDFE (that was introduced in earlier versions of this text long before any cellular standards committee re-invented the technique and gave it a different name). The $12 \cdot k$ time-domain uses of $q$ QAM input for $v_m$.

The base station’s uplink receiver best uses Figure 5.17’s structure with feedback sections. Some implementations may attempt to circumvent the feedback section by using just a feedforward section that attempts elimination of all intersymbol interference. This is similar to C-OFDM’s FEQ system, but with an IFFT to return to time domain dimensions and then a linear filter as in Chapter 3’s MMSE-LE (or DFE with feedback section included). Clearly such an exclusively linear filter can perform no better than a CDFE system (and is worse if $G \neq I$).

EXAMPLE 5.3.3 [Real baseband optimized CDFE for 0dB Gap] The $1 + .9 D^{-1}$ example 4.6.1 is revisited here with a gap of 0 dB and the CDFE. The following matlab commands illustrate the sequence of computations

```matlab
>> [gn,en_bar,bn_bar,Nstar,b_bar]=DMTra([.9 1],.181,1,8,0)
```

```matlab
gn =  
19.9448  
17.0320  
10.0000  
2.9680  
0.9552  
2.9680  
10.0000  
17.0320
en_bar = 1.2415  
1.2329  
1.1916  
0.9547  
0.9547  
1.1916  
1.2329
bn_bar = 2.3436  
2.2297  
1.8456  
0.9693  
0.9693  
1.8456  
2.2297
Nstar = 7
b_bar = 1.3814
```

```matlab
>> 10*log10(2^(2*b_bar) -1) = 7.6247 dB
>> rXX=diag([en_bar(8:-1:1)]) =
```

```matlab
1.2329  
0  
0  
0  
0  
0  
0  
0
0  
1.1916  
0  
0  
0  
0  
0  
0
0  
0  
0.9547  
0  
0  
0  
0  
0
0  
0  
0  
0  
0.9547  
0  
0  
0
0  
0  
0  
0  
0  
1.1916  
0  
0
0  
0  
0  
0  
0  
0  
1.2329  
0
0  
0  
0  
0  
0  
0  
0.9547
```

```matlab
>> rXXbar=diag([en_bar(8:-1:6) en_bar(4:-1:1)]) =
```

```matlab
1.2329  
0  
0  
0  
0  
0  
0  
0
0  
1.1916  
0  
0  
0  
0  
0  
0
```

860
\begin{verbatim}
>> J=hankel([zeros(1,7),1]);
>> Q=(1/sqrt(8))*J*fft(J);
>> J7=hankel([zeros(1,6),1]);
>> Qtilde=(1/sqrt(7))*J7*fft(J7);
>> ruu=real(Qtilde'*rXXbar*Qtilde) =
1.1429  0.0755 -0.0377  0.0116  0.0116 -0.0377  0.0755
  0.0755  1.1429  0.0755 -0.0377  0.0116  0.0116 -0.0377
-0.0377  0.0755  1.1429  0.0755 -0.0377  0.0116  0.0116
  0.0116 -0.0377  0.0755  1.1429  0.0755 -0.0377  0.0116
  0.0116  0.0116 -0.0377  0.0755  1.1429  0.0755 -0.0377
-0.0377  0.0116  0.0116 -0.0377  0.0755  1.1429  0.0755
  0.0755 -0.0377  0.0116  0.0116 -0.0377  0.0755  1.1429
>> norm(imag(Qtilde'*rXXbar*Qtilde)) = 1.0645e-16
(proves taking real part only for appearance)
>> Phibar=lohc(ruu) =
1.0625  0.0763 -0.0357  0.0077  0.0159 -0.0400  0.0706
  0  1.0652  0.0738 -0.0352  0.0088  0.0132 -0.0353
  0  0  1.0658  0.0735 -0.0357  0.0101  0.0108
  0  0  0  1.0658  0.0736 -0.0361  0.0108
  0  0  0  0  1.0660  0.0731 -0.0353
  0  0  0  0  0  1.0667  0.0706
  0  0  0  0  0  0  1.0690
>> Phi=Phibar*inv(diag(diag(Phibar))) =
1.0000  0.0716 -0.0335  0.0072  0.0149 -0.0375  0.0660
  0  1.0000  0.0692 -0.0330  0.0082  0.0123 -0.0330
  0  0  1.0000  0.0690 -0.0335  0.0095  0.0101
  0  0  0  1.0000  0.0691 -0.0338  0.0101
  0  0  0  0  1.0000  0.0685 -0.0330
  0  0  0  0  0  1.0000  0.0660
  0  0  0  0  0  0  1.0000
>> Sx=(diag(diag(Gubar)))*diag(diag(Gubar))) =
1.1289  0  0  0  0  0  0
  0  1.1347  0  0  0  0  0
  0  0  1.1360  0  0  0  0
  0  0  0  1.1360  0  0  0
  0  0  0  0  1.1363  0  0
  0  0  0  0  0  1.1379  0
  0  0  0  0  0  0  1.1429
>> Jg = [
1  0  0  0  0  0  0
0  1  0  0  0  0  0
0  0  1  0  0  0  0
0  0  0  0  0  0  0
0  0  0  1  0  0  0
0  0  0  0  1  0  0
0  0  0  0  0  1  0
0  0  0  0  0  0  1];
\end{verbatim}
\[
A = \text{real}(Q'\text{jg}\tilde{Q}\text{Gubar}) = \\
\begin{bmatrix}
0.9690 & -0.0430 & 0.0265 & -0.0400 & 0.0643 & -0.1047 & 0.2044 \\
0.3040 & 0.9208 & -0.1373 & 0.0784 & -0.0748 & 0.0893 & -0.1427 \\
-0.1969 & 0.4609 & 0.8251 & -0.1903 & 0.1093 & -0.0932 & 0.1060 \\
0.1576 & -0.2170 & 0.6189 & 0.6929 & -0.2052 & 0.1182 & -0.0938 \\
-0.1314 & 0.1408 & -0.2126 & 0.7640 & 0.5365 & -0.1870 & 0.1060 \\
0.1064 & -0.0937 & 0.1084 & -0.1770 & 0.8833 & 0.3691 & -0.1427 \\
-0.0729 & 0.0515 & -0.0489 & 0.0606 & -0.1068 & 0.9658 & 0.2044 \\
-0.0000 & -0.0000 & -0.0000 & -0.0000 & -0.0000 & 1.0000 & 0.2044 \\
\end{bmatrix}
\]

\[
C = [0.9 \\
\text{zeros}(6,1)] \\
1];
\]

\[
R = [0.9 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0];
\]

\[
H = \text{toeplitz}(C, R) = \\
\begin{bmatrix}
0.9000 & 1.0000 \\
0 & 0.9000 & 1.0000 \\
0 & 0 & 0.9000 & 1.0000 \\
0 & 0 & 0 & 0.9000 & 1.0000 \\
0 & 0 & 0 & 0 & 0.9000 & 1.0000 \\
1.0000 & 0 & 0 & 0 & 0 & 0.9000 \\
\end{bmatrix}
\]

\[
H_t = (1/\sqrt{0.181})H^*;
\]

\[
[\text{snrGDFEu, GU, WU, S0, MSWMFU, b, bbar}] = \text{computeGDFE}(H_t, A, 2, 9)
\]

(note use of \(N_x = 9\) because \(n_u = 1\), but also \(A\) reduced to 7 dimensions)

\[
\text{snrGDFEu} = 7.6247 \text{ dB}
\]

\[
\text{GU} = \\
\begin{bmatrix}
1.0000 & 0.4654 & -0.0309 & -0.0024 & 0.0245 & -0.0574 & 0.4464 \\
0 & 1.0000 & 0.5340 & -0.0310 & -0.0052 & 0.0327 & -0.2178 \\
0 & 0 & 1.0000 & 0.5510 & -0.0307 & -0.0091 & 0.1120 \\
0 & 0 & 0 & 1.0000 & 0.5554 & -0.0289 & -0.0499 \\
0 & 0 & 0 & 0 & 1.0000 & 0.5549 & -0.0102 \\
0 & 0 & 0 & 0 & 0 & 1.0000 & 0.5555 \\
0 & 0 & 0 & 0 & 0 & 0 & 1.0000 \\
\end{bmatrix}
\]

\[
\text{WU} = \\
\begin{bmatrix}
0.0776 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.0387 & 0.0896 & 0 & 0 & 0 & 0 & 0 \\
0.0222 & -0.0453 & 0.0924 & 0 & 0 & 0 & 0 \\
-0.0122 & 0.0257 & -0.0470 & 0.0932 & 0 & 0 & 0 \\
0.0045 & -0.0139 & 0.0265 & -0.0474 & 0.0933 & 0 & 0 \\
0.0037 & 0.0046 & -0.0139 & 0.0265 & -0.0473 & 0.0932 & 0 \\
-0.0649 & 0.0279 & -0.0055 & -0.0122 & 0.0315 & -0.0598 & 0.1178 \\
\end{bmatrix}
\]

\[
\text{S0} = \\
\begin{bmatrix}
13.8924 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 12.1626 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 11.8204 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 11.7315 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 11.7157 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 11.7319 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 9.4913 \\
\end{bmatrix}
\]

\[
\text{MSWMFU} = \\
\begin{bmatrix}
0.2144 & 0.0140 & -0.0036 & 0.0019 & -0.0022 & 0.0042 & -0.0120 & 0.1767 \\
\end{bmatrix}
\]
The limiting transmit filter appears to be \( G_{\text{unb}} \rightarrow 1 + 0.55 \cdot D - 0.031 \cdot D^2 \), and \( W_{\text{unb}} \rightarrow 0.093 - 0.047D^{-1} + 0.0265D^{-2} - 0.0139D^{-3} \) on interior rows, but that \( A \) does not approach any (obvious) stationary form. As \( N \rightarrow \infty \), the energy of the components of \( \nu \) become constant.

The modulation and interpolation process is not triangular (causal), and will require \( N^2 \) operations in the transmitter. Similarly \( H^* \) in the receiver, because of its dependence on \( A \), also requires \( N^2 \) operations. The complexity of this interpolating system is \( N^2 \) no matter how \( G_u(D), G(D), \) and \( W(D) \) may converge. Example 5.3.4 will avoid the non-causality and \( A \) altogether through analog resampling.

The following example provides an illustration an implementation that a second version of the example resamples for the same \( 1 + .9D^{-1} \) channel to the nearly optimum non-singular input for the CDFE on the previously well-known \( 1 + .9D^{-1} \) channel.

**EXAMPLE 5.3.4** *(CDFE for \( 1 + .9D^{-1} \) channel)* Water-fill loading for DMT modulation on the \( 1 + .9D^{-1} \) channel appeared as an example in Section 4.6. There and here, \( N = 8 \) and \( \nu = 1 \) with energies:

\[
\begin{align*}
n & \quad 0 \quad 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \\
E_n & \quad 1.24 \quad 1.23 \quad 1.19 \quad .96 \quad .96 \quad 1.19 \quad 1.23 \quad 0
\end{align*}
\]

Or, equivalently from matlab

\[
\begin{align*}
\text{>> } b' & = \quad 1.8981 \quad 1.8022 \quad 1.7816 \quad 1.7762 \quad 1.7752 \quad 1.7762 \quad 1.6233 \\
\text{>> } \text{bbar} & = \quad 1.3814
\end{align*}
\]

Figure 5.18 helps illustrate the resampling at a lower rate 7/8 instead of the original rate 1.
The channel is sampled at rate $1/T^* = 7/8$ or equivalently $T^* = 8/7$ and all tones will now be used. The symbol period is now $64/7$, corresponding to 8 dimensions at spacing $T^* = 8/7$. The DFT size will be $N = 7$, in this case, allowing for one dimension of cyclic prefix. Chapter 4’s TEQ might be necessary to restore intersymbol interference to one dimension, but since ISI is less than the original invocation of this example over a wider band, we assume any TEQ loss to be negligible. To maintain the same power, $E_x = 64/7$, but because 1/8 of this energy is lost in the cyclic prefix, the energy available for DMT loading is $8 = \left(\frac{64}{7}\right) \cdot \frac{7}{8}$ units.

The following sequence of matlab commands shows the resampling interpolation:

```matlab
>> D=exp(j*[0:100]*(7/8)*.01*pi);
>> H7=sqrt(8/7)*(ones(1,101)+.9*D);
>> H7=[H7,conj(H7(101:-1:2))];
>> h7=real(ifft(H7));
>> h=[h7(200:201),h7(1:5)] =
    -0.1011  0.9393  1.1979  -0.0603  0.0394  -0.0292  0.0232
>> H=toeplitz([h(1),h(7:-1:2)],h) =
    -0.1011  0.9393  1.1979  -0.0603  0.0394  -0.0292  0.0232
    0.0232 -0.1011  0.9393  1.1979  -0.0603  0.0394  -0.0292
    -0.0292  0.0232 -0.1011  0.9393  1.1979  -0.0603  0.0394
    0.0394 -0.0292  0.0232 -0.1011  0.9393  1.1979  -0.0603
    -0.0603  0.0394 -0.0292  0.0232 -0.1011  0.9393  1.1979
```

$$(T^*)^* = \frac{8}{7}; N = 7; \nu = 1; T = (N + \nu) \cdot (T^*)^* = \frac{64}{7},$$

$\frac{P_x}{T} = \frac{E_x}{T} = 1$W  ;  $E_x = \frac{64}{7}$W/Hz

$E_x$ (DMT Loading) $= \frac{7}{8} \cdot E_x = 8$ W/Hz
This channel will now have 7 nonzero water-fill input energies allocated to it, as executed by the following matlab commands:

\[
\begin{align*}
H &= \sqrt{1/1.181} H; \\
J7 &= \text{hankel}([\text{zeros}(1,6),1]); \\
Q7 &= (1/\sqrt{7}) J7 \ast \text{fft}(J7); \\
rXX &= \text{diag}([1.23, 1.19, .96, .96, 1.19, 1.23, 1.24]); \\
rxx &= \text{real}(Q7' \ast rXX \ast Q7) = \\
\begin{pmatrix}
1.1429 & 0.0735 & -0.0364 & 0.0115 & 0.0115 & -0.0364 & 0.0735 \\
0.0735 & 1.1429 & 0.0735 & -0.0364 & 0.0115 & 0.0115 & -0.0364 \\
-0.0364 & 0.0735 & 1.1429 & 0.0735 & -0.0364 & 0.0115 & 0.0115 \\
0.0115 & -0.0364 & 0.0735 & 1.1429 & 0.0735 & -0.0364 & 0.0115 \\
0.0115 & 0.0115 & -0.0364 & 0.0735 & 1.1429 & 0.0735 & -0.0364 \\
-0.0364 & 0.0115 & 0.0115 & -0.0364 & 0.0735 & 1.1429 & 0.0735 \\
0.0735 & -0.0364 & 0.0115 & 0.0115 & -0.0364 & 0.0735 & 1.1429 \\
\end{pmatrix}
\end{align*}
\]

The designer then relates this input to a diagonal input over the new interpolated band easily because it is nonsingular, via Cholesky factorization.

\[
\begin{align*}
\Phi &= \text{lohc}(rxx); \\
\Phi &= \Phi bar \ast \text{inv}(\text{diag}(\text{diag}(\Phi bar))); \\
Sx &= \text{diag}(\text{diag}(\Phi bar))^2 = \\
\begin{pmatrix}
1.1297 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1.1352 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1.1363 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1.1364 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1.1366 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1.1381 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1.1429 \\
\end{pmatrix}
\end{align*}
\]

The proximity of the input with resampling to a white input in that \( \Phi \) is close to a diagonal matrix, converging to \( \Phi(D) = 1 + 0.07D \). The CDFE then follows in the usual manner:

\[
\begin{align*}
\text{computeGDFE}(H, A, 2, 8) = \\
\text{snrGDFEu} &= 9.1416 \text{ dB (higher, but at lower symbol rate)} \\
\text{GU} &= \\
1.0000 & 0.4783 & -0.0492 & -0.0074 & 0.0208 & -0.0760 & 0.4583 \\
0 & 1.0000 & 0.5663 & -0.0507 & -0.0100 & 0.0385 & -0.2577 \\
0 & 0 & 1.0000 & 0.5952 & -0.0517 & -0.0208 & 0.1470 \\
0 & 0 & 0 & 1.0000 & 0.6049 & -0.0463 & -0.0833 \\
0 & 0 & 0 & 0 & 1.0000 & 0.6042 & -0.0105 \\
0 & 0 & 0 & 0 & 0 & 1.0000 & 0.6074 \\
0 & 0 & 0 & 0 & 0 & 0 & 1.0000 \\
\text{WU} &= \\
0.0686 & 0 & 0 & 0 & 0 & 0 & 0 \\
-0.0360 & 0.0805 & 0 & 0 & 0 & 0 & 0 \\
0.0237 & -0.0443 & 0.0845 & 0 & 0 & 0 & 0 \\
-0.0144 & 0.0287 & -0.0471 & 0.0858 & 0 & 0 & 0 \\
\end{align*}
\]
Thus, the higher SNR=9.14 dB requires a larger number of bits per dimension to obtain the same data rate, or reducing the SNR by the factor of $10 \log_{10}(8/7) = 1.3$ dB produces 7.8 dB, so close to the other design value, but slightly better. In this case, the good DFE performance is obtained by analog resampling rather than by digital interpolation, so differences here are in approximations made. Analog resampling appears simpler here, but such a variable sampling device would have to cover the range of channel bandwidths in practice, often a much more difficult analog design than using the FFT-based interpolation in digital-signal processing.

The upper rows again converge to a nearly constant feedback filter. As $N \to \infty$, these upper rows would converge to the $G_{\text{unb}}(D)$ of the MMSE-DFE for this new sampling rate on this channel of $1/T = 7/8$, which looks close to $1 + 0.6D - 0.05D^2$, while $W_{\text{unb}}(D) \approx 0.86 - 0.04D - 0.07D^2 + 0.15D^3$. The combination of matched filter and feedforward matrix must be interpreted (with delay) as non-causal, so viewed about the diagonal, or basically $0.56D - 0.079D^3 + 0.152D^2 - 0.216D + 0.027 + 0.01D$. A larger $N$ is necessary to see the types of 10-tap feedforward filters for this example that were found to be necessary in Chapter 3.

**Complexity Observations:** As $\Gamma \to 0$, the CDFE and DMT perform the same when the CDFE correctly resamples/interpolates and thus both are nonsingular for the same $R_{xx}$. DMT would clearly be preferred, because it has no feedback section and can be implemented with $N \cdot \log_2(N)$ operations as compared to $N^2$ for the CDFE. Also, DMT has a higher SNR for any nonzero gap, and thus would be better in any practical system where gaps must be greater than 0 dB. The CDFE may see many internal values of the matrix $G$ and the feedforward filter becoming zero or negligible, thus allowing a highly channel-dependent computational reduction. In Example 5.3.4, with 6 feedforward taps and 3 feedback taps, the receiver complexity per 8-dimensional symbol is 48+24 or 72 operations/symbol. The DMT system of same block size has 8 log$_2(8) = 24$ operations/symbol, 3 times less complex. In Example 5.3.3, the CDFE receiver complexity is $1.5N^2 + N$ per symbol, or 104 operations per dimension (but does avoid the resampling, which was not included in the complexity calculation for Example 5.3.4). In general, any MMSE-DFE that requires more than $\log_2(N)$ taps in both feedback and feedforward filters together to approximate the same performance as a DMT system of symbol length $N$ would be more
complex to implement. For example, when $N = 16$ even at the higher sampling rate of $1/T = 1$, DMT achieves the maximum of $8.8 \text{ dB}$ on the $1 + 0.9 \cdot D^{-1}$ channel. However, a CDFE or MMSE-DFE clearly requires more than $\log_2(N = 16) = 4$ taps to achieve such a high SNR on this channel (the feedforward filter of the CDFE in the example already has more than 4 taps that cannot be ignored at $N = 7$).

In practice, no matter how severe the channel ISI, DMT requires $\log_2(N)$ operations per dimension, while the CDFE would need to have the capability to perform up to $N$ operations per dimension. The CDFE is nearly always more complex than DMT, no matter what the channel for any specific choice of $N$. If some of the coefficients of CDFE filters are “zero at the end,” then in those situations complexity and delay are less than $N$. In order for this complexity to reduce below the $\log_2(N)$, it is likely the channel has very mild intersymbol interference, for which analog resampling was used to reduce the ISI, but then DMT can use a smaller $N$ also, and still likely get the same performance. No known examples of lower CDFE complexity are known to the author. Thus, on severe ISI channels, the complexity advantage is likely to be large for DMT while as the ISI severity decreases, the complexity advantage diminishes to the case of the AWGN channel with no ISI, in which case both have the same complexity and performance ($\Gamma = 0 \text{ dB}, N = 1$, and $\nu = 0$). The sole possible reason for using a MMSE-DFE might be that with continuous transmission, the length of the equalizer filters might lead to a delay of less than $N$, as Example 5.3.4 shows, thus leading with very long block lengths to a shorter delay to get essentially the same SNR – equivalently, there is no “resetting” of the packet to length $N$ necessary in a very long $N$ CDFE as the transmitter and receiver simply ignore block boundaries and use MMSE-DFE continuously. Good codes’ $\Gamma = 0 \text{ dB}$ implies infinite coding and decoding delay although with best codes this delay is often on the order of a symbol period if the code is applied across the symbol’s dimensions. Basically there is little reason to use single-carrier on any filtered AWGN, which is why most modern transmission systems use DMT/C-OFDM.

Two-Band Example: A last example shows a 2-band optimization\(^\text{36}\). This example should help designers with any multi-band channel analysis they may encounter in the field as almost all the detailed matrix issues arise in this example.

**EXAMPLE 5.3.5 [CDFE for $1 + 0.5D + D^2 + 0.5D^3$ two-band real baseband channel]**

The sampling rate for Figure 5.19’s channel with $H(D) = 1 + 0.5D + D^2 + 0.5D^3$ is 1 MHz, and AWGN with $\frac{\sigma_0^2}{2} = 0.1 \text{ nW/Hz}$. The allowed transmission power is 0.5 mW. The first step of optimized design begins with determination of the water-filling bandwidth for this channel and noise, which we will initially do for a gap of 0dB. The energy per dimension is the power times the sampling period, so $.005 \times 10^{-6} = 0.5 \text{ nW/Hz}$. With this and $P(D)$ known, DMTra determines a good water-filling solution as follows:

\(^{36}\)Thanks to former teaching assistant Dr. Haleema Mehmood for providing/correcting the details of this example.
\[ b \] can be as high as 1.53 bits/dimension asymptotically, but \( N = 512 \) leads to less than 1\% loss at 1.52 Mbps. The next step finds indices for bands.

\[
\begin{align*}
\text{find}\_\text{bands} &= \text{find}(\text{en}(1:256) == 0); \\
\text{b1plus}\_\text{start} &= 1; \\
\text{b1plus}\_\text{end} &= \text{find}\_\text{bands}(1)-1 = 115 \\
\text{b1plus}\_\text{len} &= \text{b1plus}\_\text{end} - \text{b1plus}\_\text{start} + 1 = 115 \\
\text{b2plus}\_\text{start} &= \text{find}\_\text{bands}(\text{end})+1 = 145 \\
\text{b2plus}\_\text{end} &= 256; \\
\text{b2plus}\_\text{len} &= \text{b2plus}\_\text{end} - \text{b2plus}\_\text{start} + 1 = 112 \\
\text{find}\_\text{bands} &= \text{find}(\text{en}(257:512) == 0); \\
\text{b1neg}\_\text{start} &= 256+\text{find}\_\text{bands}(\text{end})+1 = 399 \\
\text{b1neg}\_\text{end} &= 512; \\
\text{b1neg}\_\text{len} &= \text{b1neg}\_\text{end} - \text{b1neg}\_\text{start} + 1 = 114 \\
\text{b2neg}\_\text{start} &= 257; \\
\text{b2neg}\_\text{end} &= 256+\text{find}\_\text{bands}(1)-1 = 369 \\
\text{b2neg}\_\text{len} &= \text{b2neg}\_\text{end} - \text{b2neg}\_\text{start} + 1 = 113
\end{align*}
\]
\[
\begin{bmatrix}
X_1 & X_2 & \cdots & X_M \\
X_1^* & X_2^* & \cdots & X_M^*
\end{bmatrix}
\]

**Figure 5.20**: Two-band optimized spectra.

```matlab
>> rXXtilde = diag([ en(512:-1:399) en(369:-1:257) en(256:-1:145) en(115:-1:1)]);
```

```matlab
>> num_zeros = 29; % (must sum to 256)
>> Jneg = [eye(114) zeros(114,227-114)
           zeros(113,227-113) eye(113)];
>> Jgpos = [eye(112) zeros(112,227-112)
           zeros(115,227-115) eye(115)];
>> Jg = [ Jneg zeros(256,227)
           zeros(256,227) Jgpos ];
```

```matlab
>> JN = hankel([zeros(1,N-1) 1]);
>> Q = 1/sqrt(N)*JN*fft(JN);
>> Jb1 = hankel([zeros(1,228) 1]);
>> Qtb1 = 1/sqrt(229)*Jb1*fft(Jb1);
>> Jb2 = hankel([zeros(1,224) 1]);
>> Qtb2 = 1/sqrt(225)*Jb2*fft(Jb2);
>> twid = exp(-sqrt(-1)*(2*pi/225)*(0.5*ones(1,225)).*(0:224)); % no DC offset
>> twid = diag(twid);
>> Qtb2 = Qtb2*twid;
```

```matlab
>> Qtb1neg = Qtb1(1:114,:);
Qtb1plus = Qtb1(115:229,:);
Qtb2neg = Qtb2(1:113,:);
Qtb2plus = Qtb2(114:225,:);
```

```matlab
Qtilde = [ zeros(114,225) Qtb1neg
           Qtb2neg zeros(113,229)
           Qtb2plus zeros(112,229)
           zeros(115,225) Qtb1plus ];
```
>> ruu = Qtilde'*rXXtilde*Qtilde;
>> norm(imag(ruu)) = 1.4629e-15
>> ruu=real(ruu);
>> size(ruu) = 454 454

>> Jnt = hankel([zeros(1,453),1]);
>> Phibar = lohc(ruu);
>> Phi=Phibar*inv(diag(diag(Phibar)));
>> Sx=diag(diag(Phibar))^-2;
>> size(Sx) =
        454 454
>> H=toeplitz([1 zeros(1,508) 0.5 1 0.5],[1 0.5 1 0.5 zeros(1,508)]);
>> A=real(Q'*Jg*Qtilde*Phibar);
>> Ht=(1/sqrt(.1))*H;

>> [snrGDFEu, GU, WU, S0, MSWMFU] = computeGDFE(Ht, A, 2, N+3);
>> snrGDFEu = Inf (numerical issues with large N, compute with logs and sums)
Figure 5.21: Two-band Example with 8.8 dB gap.

>> GU(256+114:256+122,256+114:256+122) =

1.0000  0.7224  -0.0151  0.0154  -0.0105  0.0074  -0.0056  0.0043  -0.0035
0 1.0000  0.7224  -0.0151  0.0154  -0.0105  0.0074  -0.0056  0.0043
0 0 1.0000  0.7224  -0.0151  0.0154  -0.0105  0.0074  -0.0056
0 0 0 1.0000  0.7224  -0.0151  0.0154  -0.0105  0.0074
0 0 0 0 1.0000  0.7223  -0.0151  0.0154  -0.0105
0 0 0 0 0 1.0000  0.7223  -0.0151  0.0154
0 0 0 0 0 0 1.0000  0.7223
0 0 0 0 0 0 0 1.0000

>> WU(256+114:256+122,256+114:256+122) =

0.0496  0  0  0  0  0  0  0  0
-0.0341  0.0496  0  0  0  0  0  0  0
The feedback and feedforward sections for bands 2 and 1 (includes DC) are respectively:

\[
G_{2.Unb}(D) = 1 + 0.722 \cdot D - 0.0151 \cdot D^2 + 0.0154 \cdot D^3 - 0.0105 \cdot D^4
\]
\[
W_{2.Unb}(D) = 0.0496 - 0.0341 \cdot D^{-1} - 0.0242 \cdot D^{-2} - 0.0179 \cdot D^{-3} + 0.0137D^{-4} + 0.086 \cdot D^{-5}
\]
\[
G_{1.Unb}(D) = 1 - 0.285 \cdot D - 0.208 \cdot D^2 - 0.06D^3 - 0.03 \cdot D^4 - 0.0189D^5 - 0.0130D^6
\]
\[
W_{1.Unb}(D) = 0.225 + 0.052 \cdot D^{-1} + 0.05D^{-2} + 0.0315 \cdot D^{-3} + 0.024 \cdot D^{-4} + 0.018 \cdot D^{-5} + 0.014 \cdot D^{-6}
\]

The complexity per dimension is roughly 25-30 multiplies, while for the DMT system it is 9, again about 3 times more complex.

Figure 5.21 shows the two bands with $\Gamma = 8.8$ dB, as in Figures 5.22 and 5.23 detail the transmitter and receiver structures.

Thus, transmission fundamentals do not support claims that the designer may encounter that a MMSE-DFE with QAM performs the same, but is less complex. By contrast, a very special effort might make the performance the same, but in which case the MMSE-DFE is always more complex, although with possibly lower delay.
5.3.5 Some Simplifying ZF/MMSE-Equivalence Relationships

This subsection parallels Chapter 3 and also Chapter 2’s Section 2.8 on the BC transmitter, finding an interesting relationship between Subsection 5.1.3’s ZF and MMSE GDFEs with water-filling and finding their equivalence under worst-case noise situations with any input. All relations here occur on a nonsingular square channel, which of course is the result of water-filling for the GDFE with all singularity removed. This nonsingular situation also corresponds to all-primary users, and consequently maximum sum-rate is possible with good $R_{xx}$ design for both Section 2.8’s BC and Section 2.7’s energy-sum MAC. However, in the single-user case here, all primary users just corresponds to energizing only the best dimensions.

**Nonsingular Square Channel:** This section’s results presume a square non-singular equivalent channel $H$ (so after any channel-singularity elimination and input-singularity elimination). A tacit assumption is that waterfill’s $N^* = N$ (a later comment relaxes this assumption). The white-noise equivalent channel is

$z = \tilde{H} \cdot u + n$  \hspace{1cm} (5.228)

where $R_{nn} = I$. SVD, as well as QR factorization, are unambiguous\(^{37}\) on this non-singular $H$:

$\tilde{H} = D \cdot G_{zf} \cdot Q^* = F \cdot \Lambda \cdot M^*$  \hspace{1cm} (5.229)

where $D$ is nonsingular positive-real diagonal matrix, $G_{zf}$ is monic lower triangular, and $Q$ is a unitary matrix.\(^{38}\) $F$, $\Lambda$, and $M$ are the nonsingular square SVD matrices. $K$ is the water-filling constant such that $\text{diag}(\mathcal{E}) = K \cdot I - \Lambda^{-2}$.

### 5.3.5.1 Water-fill Simplifications

Vector coding determines best water-filling input covariance as

$R_{uu} = M \cdot [K \cdot I - \Lambda^{-2}] \cdot M^*$  \hspace{1cm} (5.230)

\(^{37}\)The word “unambiguous” here implies unique to within a possible dimensional reordering that is irrelevant for single-user channels.

\(^{38}\)The Matlab rq.m at web site can be used.
Removing the channel’s $Q$ factor by absorbing $Q^*$ into input $v$ construction uses Cholesky Decomposition of $Q^* \cdot R_{uu} \cdot Q$ to produce

$$ Q^* \cdot R_{uu} \cdot Q = G_\phi \cdot G_\phi^* , \quad (5.231) $$

with $R_{vv} = I$ and $u = G_\phi \cdot v$. $G_\phi$ is not necessarily monic. This then relates the VC-GDFE form to the Cholesky triangular form by

$$ R_{uu} = M \cdot [K \cdot I - \Lambda^2] \cdot M^* = Q \cdot G_\phi \cdot G_\phi^* \cdot Q^* . \quad (5.232) $$

![Figure 5.24: Relationship diagram for the ZF-GDFE and GDFE with water-filling input choice.](image)

**Water-Fall Feedback Section Convergence:** Figure 5.24's upper portion illustrates the consequent transmitter and simplified ZF receiver when the input $Q^* u$ derives from $v$ that has $R_{vv} = I$. The overall channel simplifies to monic upper triangular $G_{tot}$ that satisfies

$$ G_{tot} \triangleq S_{tot} \cdot G_{zf} \cdot G_\phi , \quad (5.233) $$
or

$$ G_{tot} \cdot S_{\phi}^{-1/2} \cdot G_{\phi}^{-1} = S_{tot} \cdot G , \quad (5.234) $$

where diagonal $S_{tot} \succeq 0$ makes $G_{tot}$ monic. Figure 5.24’s upper ZF-GDFE then uses a feedback section of $G_{tot}$ and the corresponding SNRs are

$$ \text{SNR}_n = D_n^2 , \quad (5.235) $$

using Figure 5.24 and (5.229). Basic matrix multiplication produces

$$ \tilde{H}^* \cdot \tilde{H} = M \cdot \Lambda^2 \cdot M^* = Q \cdot G^* \cdot D^2 \cdot G_{zf} \cdot Q^* $$

with corresponding inverse

$$ \left[ \tilde{H}^* \cdot \tilde{H} \right]^{-1} = \tilde{H}^{-1} \cdot \tilde{H}^{-*} = M \cdot \Lambda^{-2} \cdot M^* = Q \cdot G^{-1} \cdot D^{-2} \cdot G_{zf}^{-*} \cdot Q^* . \quad (5.237) $$

Substitution of (5.237) into (5.232) yields

$$ K \cdot I = Q \cdot G_\phi \cdot S_\phi \cdot G_\phi^* \cdot Q^* + Q \cdot G_{zf}^{-1} \cdot D^{-2} \cdot G_{zf}^{-*} \cdot Q^* \quad (5.238) $$

$$ K \cdot G_{zf} \cdot G_{zf} = S_{tot}^{-1} \cdot G_{tot} \cdot G_{tot}^* \cdot S_{tot}^{-1} + D^{-2} \quad (5.239) $$

$$ K \cdot D \cdot G_{zf} \cdot G_{zf}^* = \frac{(D \cdot S_{tot}^{-1}) \cdot G_{tot} \cdot G_{tot}^* \cdot (S_{tot}^{-1} \cdot D) + I}{R_{zf}} \quad (5.240) $$
Equation (5.240) relates that the “zero-forcing” $G_{zf}$ originally found with the channel $R_n^{1/2} \cdot H$’s QR factorization also determines the shape $G$ of the MMSE-GDFE’s equivalent-channel-plus-noise-with-water-filling $R_{uu}$, as in the lower portion of Figure 5.24:

$$G = G_{zf} \text{ when input is nonsingular water-filling} \ .$$

That is, this water-filling MMSE-GDFE has the same $G$ as the original nonsingular $\tilde{H}$ channel. The diagonal scaling only sets the matrices as monic. Despite the common shape, the ZF-GDFE detector will perform worse than the MMSE-GDFE unless $G_\phi = s_\phi \cdot I$ (white input), when they are the same. This white input is best for a unitary (all-pass) channel.

**Computational Simplification:** QR factorization is much easier to calculate than is singular-value decomposition. So, for a nonsingular square $\tilde{H}$, Equation (5.240) directly computes the necessary quantities and water-fill input energies

$$K \cdot S_{tot} \cdot G \cdot G^* \cdot S_{tot} - (S_{tot} \cdot D^{-1})^2 = G_{tot}^* \cdot G_{tot} \quad (5.242)$$

The left side of (5.242) uses the quantities $G$ and $D$ of QR factorization, and $K$, which for water-filling easily determines (recalling $N^* = N$ in this special case) as

$$K = \bar{\xi} - \frac{1}{N} \sum_{n=1}^{N} \frac{1}{\lambda_n^2} = \bar{\xi} - \frac{1}{N} \text{trace} \left\{ \left[ \tilde{H}^* \tilde{H} \right]^{-1} \right\} \ .$$

The $G$ needed for the GDFE calculations thus follows without SVD if the input is water-filling and nonsingular already.

**Avoid SVD:** If $N^* < N$, then SVD would be needed to find $N^*$. This occurs if (5.244) is negative, which then suggests the original problem did not have all singularity removed. However, in field use to avoid the SVD, the design can instead increase $K > 0$ to an arbitrary value $(1 + \alpha) \cdot K$ with $\alpha > 0$, recompute (5.244), and then scale all energies by $1/(1 + \alpha)$. This is not optimum but if the $\alpha$ is chosen judiciously small, it typically is not far from optimum on channels with reasonably large dimensionality. Chapters 3 and 4 discuss energization of bands that water-fill would zero and that if they occupy a significant bandwidth/dimensionality, the loss can be severe. However, in some wireless designs where the number of should-be-zeroed bands is small, the ZF GDFE can be used with little loss, simplifying adaptive real-time design calculations significantly. This is kind of “chicken and egg” in that how is the input water-filling and nonsingular without knowing the channel, but many wireless designers simply assume that flat energy over a judicious fixed set of dimensions (or bandwidth) is close enough to water-fill, and then the QR factorization completes the design.

### 5.3.5.2 Worst-case noise simplifications:

Chapter 2, Section 8 for the BC, and also Chapter 3, Section 3.12.8 for continuous frequency/time, found an $R_{xx}$-dependent worst-case noise autocorrelation with fixed (known) diagonal elements of $R_{nn}$. The BC worst-case noise correlation results in a diagonal MMSE GDFE receiver, which now is recognized as that of this section. With singularity removed, the worst-case-noise MMSE GDFE then becomes equivalent to the BC case of all primary users. Worst-case noise applies for any input $R_{xx}$ (not just water-filling as in Subsection 5.3.5.1), and a corresponding discrete-modulator matrix $A$ can be found to synthesize $R_{xx}$ as well as diagonalize the GDFE feedforward processing. In the single user case, the worst-case noise situation allows almost all GDFE complexity to be in the transmitter modulation matrices (including the noiseless precoder). The following lemma also then applies:
Lemma 5.3.1 [Best performance with diagonalized processing] For a given $R_{xx}$ and corresponding $R_{wcn}$, the canonical (best) performing GDFE requires no feedforward signal processing (coordination among receiver dimensions), presuming $x$ and the noise are Gaussian.

Proof: If a better receiver for any $R_{xx}$ and corresponding $R_{wcn}$ existed with coordinated signal processing, it would be a possible setting for the GDFE considered in Section 2.8’s optimization over BC rate sum, and thus would exceed the canonical performance data rate, which would violate basic capacity limits. QED.

Equivalence of ZF and MMSE GDFE’s under worst-case noise While Section 2.8’s BC design process found best BC design may assume $R_{nn} = R_{wcn}$, the $I(x; y)$-minimizing “worst-case” noise for given per user/receiver-noise autocorrelation matrices $R_{nn}(u)$, this did not mean the noise was actually worst-case. If instead, $R_{nn} = R_{wcn}$ is actually true, then the ZF and MMSE designs also perform the same and have the same (unbiased case for MMSE) lossless precoder for any channel $H$ is (square).

This chapter’s MMSE-GDFE linear-component receiver design of

$$W_{unb} = (S_0 - I)^{-1} \cdot G^{-*} \cdot R_{nn}^{-1} \cdot H^*$$

(5.245)

is the same as Section 2.8’s (unbiased) BC receiver(s). $W_{unb}$ is diagonal when $R_{wcn}$ is the noise autocorrelation\(^{39}\). This diagonal receiver is the highest performance (reliably decodable data rate) at $I_{wcn}$, which was the BC’s rate sum and is the single-user data rate here.

The ZF-GDFE uses the worst-case noise and “QR” factors\(^{40}\) the inverse-noise-weighted nonsingular (block) square channel matrix to obtain

$$R_{xx} = R_{ZF}^* \cdot Q_1 \cdot \Phi^* ;$$

(5.247)

An invertible transmitter modulator matrix $A = Q_{ZF}$ so that of $x = Q_{ZF} \cdot v$ with a lossless precoder of $G = G_{ZF}$ does not change the mutual information, and individual (ML for applied zero-gap code) decoders on each output dimension canonically achieve this data rate. However, it is possible that

$$R_{xx} \neq Q_{ZF} \cdot R_{vv} \cdot Q_{ZF}^* \triangleq R_{xx}(ZF) .$$

(5.248)

The overall ZF triangular section cascades the upper triangular $\Phi$ and $R_1$ to form

$$D_A \cdot G_{ZF} = R_1 \cdot \Phi$$

(5.249)

where $D_A$ is the left-side extracted diagonal that makes $G_{ZF}$ monic. The discrete modulator matrix is then

$$A = Q_1 \cdot \Phi .$$

(5.250)

The MMSE-GDFE with this modulator matrix has diagonal receiver forward processing as in Section 2.8.3.

---

\(^{39}\)There is no concern here for sub-users in the single-user case; indeed the user block-diagonal issues also trivially become single dimensions also when $H$ is nonsingular.

\(^{40}\)This is not the usual square-root-noise-whitened $R_{wcn}^{-1/2} \cdot H$ channel, but actually an “overwhitened” $R_{wcn}^{-1} \cdot H$ causing a deviation in $H$ definition.
Some Examples of WCN and GDFE

EXAMPLE 5.3.6 [Vector-coded input for WCN] This first example uses the $3 \times 4$ channel matrix well known in this text as the $1 + 0.9 \cdot D^{-1}$ channel. An input with discrete modulator $C = I$ and equal energy on each of the 3 pass modes of this channel is constructed, consequent worst-case noise determined, a GDFE designed, and the overall feedforward-processing determined to be consequently diagonal.

$$H = \begin{bmatrix} 0.9000 & 1.0000 & 0 & 0 \\ 0 & 0.9000 & 1.0000 & 0 \\ 0 & 0 & 0.9000 & 1.0000 \end{bmatrix}$$

$$H = \left(1/\sqrt{0.181}\right)H;$$

$$H = \text{eye}(4);$$

$$[\text{Rwcn}, \text{rate}] = \text{wcnoise}(Rxx, H, 1)$$

\[
\begin{align*}
\text{Rwcn} &= \begin{bmatrix} 1.0000 & 0.0694 & -0.0282 \\ 0.0694 & 1.0000 & 0.0694 \\ -0.0282 & 0.0694 & 1.0000 \end{bmatrix} \\
\text{rate} &= 4.8024
\end{align*}
\]

$$H_{\text{tilde}} = \text{inv}(\text{Rwcn}) \cdot H = \begin{bmatrix} 2.1281 & 2.2118 & -0.0990 & 0.0785 \\ -0.1527 & 1.9670 & 2.2214 & -0.1697 \\ 0.0707 & -0.0742 & 1.9584 & 2.3645 \end{bmatrix}$$

$$[R, Q] = \text{rq}(H_{\text{tilde}});$$

Unlike the multiuser BC case, there is no concern for the rq.m program’s dimensional reordering in the single-user case.

$$R = \begin{bmatrix} 0 & -2.7323 & -1.4039 & 0.0071 \\ 0 & 0 & -2.7077 & -1.2346 \\ 0 & 0 & 0 & -3.0719 \end{bmatrix}$$

$$Q = Q(:, 2:4);$$

$$R_{\text{xxrot}} = Q \cdot Rxx \cdot Q = \begin{bmatrix} 1.0000 & -0.0000 & 0 \\ -0.0000 & 1.0000 & 0.0000 \\ 0 & 0 & 1.0000 \end{bmatrix}$$

$$\Phi_{\text{bar}} = \text{lohc}(R_{\text{xxrot}})$$

$$\Phi_{\text{bar}} = \begin{bmatrix} 1.0000 & -0.0000 & 0 \\ 0 & 1.0000 & 0.0000 \\ 0 & 0 & 1.0000 \end{bmatrix}$$

$$R_{\text{up}} = R(:, 2:4) = \begin{bmatrix} -2.7323 & -1.4039 & 0.0071 \\ 0 & -2.7077 & -1.2346 \\ 0 & 0 & -3.0719 \end{bmatrix}$$

$$D_{A} = \text{diag}(\text{diag}(R_{\text{up}} \cdot \Phi_{\text{bar}})) = \begin{bmatrix} -2.7323 & 0 & 0 \\ 0 & -2.7077 & 0 \\ 0 & 0 & -3.0719 \end{bmatrix}$$

$$G = \text{inv}(D_{A}) \cdot R_{\text{up}} \cdot \Phi_{\text{bar}}$$
\[
\begin{bmatrix}
1.0000 & 0.5138 & -0.0026 \\
0 & 1.0000 & 0.4559 \\
0 & 0 & 1.0000
\end{bmatrix}
\]

\[[\text{A} = Q_1 \Phi_b] =
\begin{bmatrix}
-0.8133 & 0.0669 & -0.0230 \\
-0.4305 & -0.7375 & 0.0241 \\
0.3067 & -0.5297 & -0.6375 \\
-0.2433 & 0.4136 & -0.7697
\end{bmatrix}
\]

\[[\text{sri} = \text{inv} (\sqrt{\text{m}(\text{R}_w)})) =
\begin{bmatrix}
1.0022 & -0.0357 & 0.0160 \\
-0.0357 & 1.0037 & -0.0357 \\
0.0160 & -0.0357 & 1.0022
\end{bmatrix}
\]

\[[\text{snrGDFEu, GU, WU, S0, MSWMFU, } b, bbar] = \text{computeGDFE} (\text{sri} \ast H,  A, 2, 4)
\]

\[\text{snrGDFEu} = 1.1340 \text{ dB}
\]

\[GU =
\begin{bmatrix}
1.0000 & 0.5826 & -0.0030 \\
0 & 1.0000 & 0.5160 \\
0 & 0 & 1.0000
\end{bmatrix}
\]

\[WU =
\begin{bmatrix}
0.1339 & 0 & 0 \\
-0.0676 & 0.1317 & 0 \\
0.0244 & -0.0470 & 0.1031
\end{bmatrix}
\]

\[S0 =
\begin{bmatrix}
8.4657 & 0 & 0 \\
0 & 8.5956 & 0 \\
0 & 0 & 10.7006
\end{bmatrix}
\]

\[\text{MSWMFU} =
\begin{bmatrix}
-0.3657 & -0.0128 & 0.0054 \\
-0.0125 & -0.3560 & -0.0125 \\
0.0047 & -0.0111 & -0.3164
\end{bmatrix}
\]

\[b' = 1.5408 \ 1.5518 \ 1.7098
\]

\[bbar = 1.2006
\]

\[\text{sum(b)} = 4.8024 \text{ (checks)}
\]

The MSWMFU is not diagonal, but recall the unusual $\bar{H}$ definition that effectively adds an additional square-root matrix that the above matrix still includes. To remove it, that factor extracts by multiplying MSWMFU by the square-root (conjugate transpose) inverse above., so

\[[\text{MSWMFU} \ast \text{sri}'] =
\begin{bmatrix}
-0.3660 & -0.0000 & 0.0000 \\
-0.0000 & -0.3565 & 0.0000 \\
0.0000 & -0.0000 & -0.3167
\end{bmatrix}
\]

is indeed diagonal and admits the receiver's independent dimensional processing in parallel. The input $R_{xx}$ is rank $\rho_x = 4$ but the chanel has rank $\rho_H = 3 < \rho_x$; but the wcn program finds the correct worst-case noise corresponding to the part of $R_{xx}$ in the pass space. The $\Phi$ matrix is trivially an identity because the input was initially white in this example - this does not always occur. Note that the inverse square-root of $R_{wcn}$ was matlab's symmetric square root; indeed any square root could be used, but the corresponding correction factor for the full MSWMF that includes the noise whitening would be the conjugate transpose.
This first discrete modulator above has a diagonal $R_{xx}$ as the input autocorrelation.

The next example illustrates the more general case where the input covariance is more arbitrarily chosen, thus the initial modulation matrix $C$ is not a diagonal nor based on $M$, but the channel is the same otherwise.

\[
\begin{bmatrix}
1 & 1 & 1 \\
-1 & 1 & 1 \\
1 & -1 & -1 \\
-1 & -1 & 1
\end{bmatrix};
\]
\[
R_{xx} = C^*C' = \\
\begin{bmatrix}
3 & 1 & -1 & -1 \\
1 & 3 & -3 & 1 \\
-1 & -3 & 3 & -1 \\
-1 & 1 & -1 & 3
\end{bmatrix};
\]
\[
R_{xx} = (1/3)*((3/4)*R_{xx};
\]
\[
[R_{wcn}, rate] = wcnoise(R_{xx}, H, 1)
\]

\[
\begin{bmatrix}
1.0000 & -0.0556 & -0.1112 \\
-0.0556 & 1.0000 & -0.0004 \\
-0.1112 & -0.0004 & 1.0000
\end{bmatrix};
\]
rate = 2.7280

\[
H_{tilde} = inv(R_{wcn})*H;
\]
\[
[R, Q] = rq(H_{tilde})
\]

\[
R = \\
\begin{bmatrix}
0 & -2.5801 & -1.9449 & -0.8193 \\
0 & 0 & -2.7372 & -1.7985 \\
0 & 0 & 0 & -3.2336
\end{bmatrix};
\]

\[
Q = \\
\begin{bmatrix}
0.5776 & -0.8130 & 0.0048 & -0.0739 \\
-0.5198 & -0.3660 & -0.7670 & -0.0865 \\
0.4678 & 0.3905 & -0.4282 & -0.6673 \\
-0.4211 & -0.2294 & 0.4778 & -0.7360
\end{bmatrix};
\]

\[
R_{up} = R(:, 2:4);
\]
\[
Q_1 = Q(:, 2:4); \\
R_{xxrot} = Q_1*R_{xx}*Q_1; \\
Phibar = lohc(R_{xxrot}); \\
DA = diag(diag(R_{up}*Phibar)); \\
G = inv(DA)*R_{up}*Phibar = \\
\begin{bmatrix}
1.0000 & -0.0161 & -0.5946 \\
0 & 1.0000 & 0.0095 \\
0 & 0 & 1.0000
\end{bmatrix};
\]

\[
A = Q_1*Phibar = \\
\begin{bmatrix}
-0.8226 & 0.0383 & 0.3490 \\
-0.3704 & -0.0139 & 0.4432 \\
0.3951 & -0.0357 & -0.4357 \\
-0.2321 & 0.0301 & -0.5525
\end{bmatrix};
\]
\[
sri = inv(sqrtm(R_{wcn})) = \\
\begin{bmatrix}
1.0059 & 0.0281 & 0.0561
\end{bmatrix};
\]

879
\[
0.0281 \quad 1.0012 \quad 0.0026 \\
0.0561 \quad 0.0026 \quad 1.0047
\]

\[
\text{snrGDFEu} = 1.9699 \text{ dB}
\]

\[
\text{GU} =
\begin{bmatrix}
1.0000 & -0.0185 & -0.6818 \\
0 & 1.0000 & 0.7633 \\
0 & 0 & 1.0000
\end{bmatrix}
\]

\[
\text{MSWMFU} =
\begin{bmatrix}
-0.3823 & 0.0107 & 0.0213 \\
0.2459 & -8.8169 & 0.0087 \\
0.0251 & 0.0004 & -0.4497
\end{bmatrix}
\]

\[
b' = 1.4832 \quad 0.0090 \quad 1.2358 \\
bbar = 0.6820
\]

\[
4 \times bbar = 2.7280 \text{ (checks)}
\]

\[
\text{MSWMFU} \times sri'
\begin{bmatrix}
-0.2051 & 0.0000 & 0.0000 \\
0.0000 & -4.2034 & 0.0000 \\
0.0000 & 0.0000 & -0.2235
\end{bmatrix}
\]

In this case, \( \Phi \) is triangular, but not diagonal. However, there still exists an input that diagonalizes the GDFE feedforward processing (including the first noise-whitening step).

The adventurous reader might wonder what is largest possible data rate for this channel with worst-case noise maximum input energy 12? This would correspond to the maximum of a BC.

\[
\text{Rxx} =
\begin{bmatrix}
0.6318 & 0.3441 & -0.2357 & 0.1800 \\
0.3441 & 0.7754 & 0.2588 & -0.1977 \\
-0.2357 & 0.2588 & 0.8127 & 0.2602 \\
0.1800 & -0.1977 & 0.2602 & 0.7800
\end{bmatrix}
\]

\[
\text{Rwcn} =
\begin{bmatrix}
1.0000 & 0.0811 & -0.0352 \\
0.0811 & 1.0000 & 0.0811 \\
-0.0352 & 0.0811 & 1.0000
\end{bmatrix}
\]

\[
bmax = 4.8105 \text{ bits per 4-dimensional input symbol}
\]

\[
bmax/4 = 1.2026
\]

\[
\text{check with vector coding:}
\]

\[
\text{[F,L,M]=svd(H)};
\]

\[
L =
\begin{bmatrix}
4.1270 & 0 & 0 & 0 \\
0 & 3.1623 & 0 & 0 \\
0 & 0 & 1.7228 & 0
\end{bmatrix}
\]

\[
\text{gn} = \text{diag}(L);
\]

\[
\text{gn} = gn^2 = 17.0320
\]
10.0000
2.9680
>> [bn, en, Nstar] = waterfill_gn(gn', 1, 0, 2)

bn = 2.1554 1.7713 0.8950
en = 1.1065 1.0652 0.8283
Nstar = 3

>> sum(bn) = 4.8217
>> sum(bn)/4 = 1.2054
(> bmax/4 = 1.2026 bits/symbol but pretty close meaning that the
water-fill against wcn is only slightly worse than full water fill).

The examples emphasize that only for the worst-case noise, no matter what is the input covariance,
there always exists a modulation of the input that diagonalizes the feedforward section of the GDFE.
This is independent of “water-filling” or “vector-coding” input selections. That diagonalization also
Corresponds to a ZF-GDFE solution. Further, the output dimensions not zeroed by such diagonalization
would correspond to primary users only if a BC with $L_{y,u} = 1$.

Following from the BC analogy, when both water-fill and worst-case noise simultaneously occur, the
data rate is maximum over all receivers that cannot coordinate between dimensions. The bumax.m
program From Section 2.8 and also Appendix G can be used to find this rate.

5.3.6 Asymptotic Stationary Convergence of certain GDFEs

Section 5.2 observes that the CDFE with nonsingular, Toeplitz, circulant $H$ and stationary (circulant
also) $R_{xx}$ is very special, and appeared in all examples to converge to Chapter 3’s MMSE-DFE as
$N \to \infty$. This section further investigates such convergence. The results here are often known in the
statistical signal-processing area as arising from “Toeplitz Distribution Theory.”

There are 3 GDFE-essential asymptotic results of interest:

1. Toeplitz and cyclic autocorrelation matrices - Subsection 5.3.6.1
2. linear prediction and canonical factorization - Subsection 5.3.6.2
3. Use of 1 and 2 in the GDFE - Subsection 5.3.6.3.

5.3.6.1 Stationary Channels

Strictly speaking, for any finite $N$, random vectors like the transmission channel’s $x$, $y$ and $n$ in $y = H \cdot x + n$ are never stationary at subsymbol-sample level\(^{41}\), even when $L_x = L_y = 1$ and there is then
no space-time MIMO. These vectors are vector-random processes, e.g. $x \to x_k$, when $N < \infty$. When
the same autocorrelation matrix for each such vector recurs at all discrete-time indices $k \in \mathbb{Z}$, the vector
process is block stationary, which corresponds to these random-process vectors’ time-indexed elements
being cyclo-stationary with period equal to $N_x = N + \nu$ dimensions (samples).

However, singular random processes with $\rho_x < N$ are not block nor cyclo-stationary unless the
 singularity has been removed asymptotically, as in Section 3.11. The GDFE attain a SNR that is
 canonical in all cases for the block-stationary case, and the data rate of $\mathcal{I}$ is only reliably achievable if
the transmission system $(R_{xx}, H, R_{nn})$ is block stationary. Equivalently, Chapter 3’s MMSE-DFE is
the nonsingular (PWC-satisfying) stationary limit as $N \to \infty$.

However, for the ISI-specific situation investigated throughout Chapters 3 and 4 and often in this
chapter, another subsymbol-level stationarity exists: The subsymbol random processes $x_k$, $y_k$, and $n_k$
are stationary\(^{42}\). In this case only, additional GDFE structure occurs that can lead (after all singularity’s

\(^{41}\)The term “stationary” in this chapter means “wide-sense stationarity,” and so only means, covariances need be invariant
to time.
\(^{42}\)Often these are scalars $x_k = x_k$, $y_k = y_k$, and $n_k = n_k$; however, there can be vectors of vector-elements to which
Section 3.10 applies.
removal) to (a set of) stationary MMSE-DFE(s). In this case, (possibly block in MIMO case) Toeplitz autocorrelation matrices for all processes can occur at each and every block length $N$ (or $N_x = N + \nu$), and convergence to fixed structures occurs as $N \to \infty$. This convergence requires care in structuring the successive processing of symbols (or blocks) even when the underlying channel and its input are stationary.

For instance, a scalar FIR channel $H(D)$ with stationary AWGN noise and a stationary input with thus an autocorrelation function $R_x(D)$ may not lead to Toeplitz matrices in partitioning. A easy example is the $1 + 0.9 \cdot D^{-1}$ channel with $\nu = 1$ that this text often investigates. For this channel $H$ and the usual guard period, which is a Toeplitz $N \times N_x$ convolution matrix, the channel-output autocorrelation matrix is not Toeplitz, i.e., $R_{yy} \not\to R_{yy}(D)$. For instance, the real baseband case with $N = 2$ and $\tilde{E}_x = 1$ yields

$$R_f = \begin{bmatrix} 0.81 & 0.9 & 0 \\ 0.9 & 1.81 & 0.9 \\ 0 & 0.9 & 1 \end{bmatrix} + \sigma^2 \cdot I,$$

which is not Toeplitz (the diagonal is not constant). As $N \to \infty$, this matrix will approach Toeplitz, but it is not Toeplitz for each finite $N$. The repeated guard-period insertion thus essentially destroys the stationarity that is otherwise naturally present.

However, with the use of the cyclic prefix, the corresponding $R_{yy}$ becomes

$$R_{yy} = \begin{bmatrix} 1.81 & 0.9 \\ 0.9 & 1.81 \end{bmatrix} + \sigma^2 I,$$

which is Toeplitz. Indeed, a cyclic prefix of length $\nu$ samples when the channel (or TEQ equalized channel) has length of $\nu$ or less will always produce a stationary $R_{yy} = H^* \cdot R_{xx} \cdot H + R_{nn}$ as long as $R_{nn}$ is Toeplitz (for instance white), and $R_{xx}$ is not only Toeplitz, but also cyclic. 43 Clearly, the cyclic prefix insures a Toeplitz $R_{yy}$ signal component.44 Thus, the cyclic prefix has an additional benefit in asymptotic analysis in that Toeplitz matrices appear throughout the GDFE design for all $N$, leading to various Cholesky factorizations appearing to converge for smaller $N$ than would otherwise be necessary. This was clearly evident Subsection 5.3.4’s earlier CDFE examples’ convergence as $N \to \infty$, while a more general GDFE appeared to have no easy convergence evident (at least the $N$ values used in previous examples).

With multiple transmission bands ($M > 1$), the matrix $H = H \cdot A$ of Section 5.3 leads to a $\tilde{H}^* \cdot \tilde{H}$ that is block diagonal. Each block along the diagonal is Toeplitz in this matrix. Thus, the remarks in this section apply individually to each of these blocks, and of course cannot apply to the entire multiband-channel matrix, which is not Toeplitz in the multiband case.

### 5.3.6.2 Canonical Factorization for finite- and infinite-length stationary sequences

Appendix D shows canonical factorization results and the Paley Weiner Criterion for infinite-length random processes and relates a stationary process’ asymptotic finite-length Cholesky factorization to linear prediction. The results presume stationarity of all (possibly vector) random processes and thus for instance $R_{xx}$ is (possibly block) Toeplitz.

In the finite-length case, the sequence $v_k$ is the MMSE sequence corresponding to estimating $x_k$ from its past dimensions $x_{k-1}, ..., x_0$, which follows directly from forming the $m^{th}$-order linear-prediction error

$$v_{N} = x_{N} - \phi_1^* \cdot x_{N-1} - ... - \phi_{m-1}^* \cdot x_0,$$  (5.253)

43The product of Hermitian cyclic Toeplitz matrices can be shown to be Hermitian cyclic Toeplitz - hint, think circular convolution.

44In fact, any noise whitening by cyclic $R_{nn}^{-1/2}$ that is done for equivalent channels needs then also for $R_{nn}$ and $R_{nn}^{-1/2}$ to be cyclic – cyclic $R_{nn}$ occurs naturally in the white-noise case, but rarely otherwise. However, nearly all noise is close to cyclic when $N \to \infty$, and because the noise is much smaller than signal in cases where equalization of any type is of interest, the approximation to a cyclic $R_{nn}$ is usually very good. In fact, in DMT systems, noise-whitening is ignored and the SNRs measured directly with whatever noise at each frequency, tacitly assuming the cyclic nature of $R_{nn}$.  

882
and from noting via the orthogonality criterion

\[ E \left[ v_N \cdot x_{N-1}^* \right] = 0 \quad \forall \ i = 1, \ldots, k - 1 \ , \quad (5.254) \]

when the \( \phi_i^* \)'s are the entries from the corresponding row (s) of (possibly block) Cholesky factorization with subscript indices that now make the appearance more like a stationary convolutional filter.\(^{45}\) Clearly then through the linear-prediction problem,

\[ \lim_{N \to \infty} \left[ I \ \phi_1^* \ldots \phi_{N-1}^* \right] \to \Phi^*(D) \ , \quad (5.255) \]

because the MMSE solution is unique when the inputs are stationary or equivalently all \( R_{xx}(N) \) as \( N \to \infty \) are nonsingular.

The diagonal matrix \( R_{vv} \) must also become constant as \( N \to \infty \) because the diagonal values represent the single constant input-energy value

\[ \lim_{N \to \infty} R_{vv}(i) = E [v_i \cdot v_i^*] = S_v = \mathcal{E}_x = \text{trace} \{ R_{xx} \} \quad \forall \ i \geq 0 \ . \quad (5.256) \]

For GDFE designs that use \( R_{vv} = I \), the constant value \( S_v \) absorbs into the channel for analysis in previous sections. This result does not hold for a singular process – that means input singularity must be eliminated, or equivalently the PWC must be satisfied. When a water-filling or any other circulant \( R_{xx} \) has sets of nonzero DFT values that contain zeroed values, the consequent singularity must be eliminated as earlier in this section.

Then a set of separate Cholesky filters, each independently acting on its independent inputs, corresponds to and ultimately converges to each band’s baseband-equivalent linear prediction filter. Canonical factorization across the whole band does not work for finite length nor for infinite-length when water-filling produces zero-energy bands. The reader may replace \( v_k \) by \( v_k^* \) and \( x_k \) by \( x_k^* \) for each subsymbol and the results apply to MIMO \( x(D) \) throughout this subsection, with Cholesky → Block Cholesky and a final vector coding on the \( L_x \times L_x \) asymptotic \( x(D) \), \( y(D) \), \( H(D) \) as \( N \to \infty \).

### 5.3.6.3 Convergence of the Canonical Channel Models and the CDFE

The CDFE must separate into the \( M \) distinct bands of the block triangular \( \Phi \), creating a set of \( R_f(m) \) for \( m = 1, \ldots, M \) when \( M > 1 \). The following results will not hold otherwise. Each band has a baseband equivalent. Each band’s matrix \( R_f \) is Toeplitz when \( R_{nn} \) is Toeplitz, and \( H \) and \( R_{xx} \) are circulant. If \( R_{nn} \) is circulant\(^{47}\), then so is \( R_f \); then \( R_b^{-1} = R_f + I \) is also a circulant matrix.

**CDFE Filter Convergence:** The Cholesky factorization of

\[ R_b^{-1} = G^* \cdot S_0 \cdot G \quad (5.257) \]

corresponds to a inverse-order prediction problem (sometimes called reverse or “backward” linear prediction), or equivalently

\[ R_b = G^{-1} \cdot S_0^{-1} \cdot G^* \quad (5.258) \]

corresponds to a normal “forward” linear-prediction problem with \( G^{-1} \) itself viewed as the upper-triangular causal matrix that relates computation of sequence values to their components on past innovations input values. In either case, the rows of \( G \) and therefore \( G^{-1} \) (or vice versa) converge to constant settings. This means the CDFE (or any GDFE that has asymptotically Toeplitz \( R_b \)) must then converge

---

\(^{45}\)The satisfaction of (5.254) follows directly from \( R_{vv} \) being diagonal and thus \( v_{1} \ldots v_{N} \) or \( v_{N-1} \) all uncorrelated with \( v_{k} \) and thus so must also be \( x_{1} \ldots x_{k} \) or \( x_{N} \) since \( x_{k-1} = \Phi(k-1)v_{k-1} \) from Cholesky factorization of one order lower. The non-singularity of \( R_{xx} \) ensures Cholesky factorization of all orders. The use of the conjugate transpose instead of transpose is simple semantics as the optimization is over the quantity and would simply conjugate the ultimate optimum variable’s value. In the scalar case they may be removed. For the scalar case, they are unnecessary.

\(^{46}\)The limit here is taken in the two-sided sense of time going forward and backward to infinity so that is then true for all \( i \).

\(^{47}\)e.g., \( R_{nn} = I \) or anything close to constant-diagonal matrix.
to a system with a constant feedback section (including DMT and VC, which have that constant equal to zero at all lengths). This filter is known to be the unique $G(D)$ of the corresponding band of the MMSE-DFE, so then

$$
\lim_{N \to \infty} \text{any row of } G = G(D) .
$$

(5.259)

**MMSE Convergence:** The MMSE factorization must also then converge to a constant

$$
\lim_{N \to \infty} S_0(i) = s_0 \ \forall \ i .
$$

(5.260)

$S_0^{-1}$'s diagonal entries also represent the GDFE’s dimensional MMSE values. Thus because $S_0$ tends to a constant-diagonal matrix, there is then a constant MMSE for the GDFE as $N \to \infty$, which is known from Chapter 3 as

$$
S_0^{-1} = \frac{X_0}{\gamma_0 \cdot \|h\|^2} .
$$

(5.261)

or

$$
S_0 = \frac{\gamma_0 \cdot \|h\|^2}{X_0} .
$$

(5.262)

**SNR Convergence:** The feedforward matrix $S_0^{-1} \cdot G^{-*}$ will be thus be anti-causal (lower triangular) as $N \to \infty$. The circulant matched filter $H^*$ can be mixed phase, and thus a fixed filter for all $N$. Both the matched filter and the feedforward matrix need realization in practice with delay (which is at most $N + \nu$ with GDFE’s).

The CDFE SNR then converges in each band to

$$
\lim_{N \to \infty} \text{SNR}_{\text{cdf}} = \lim_{N \to \infty} \left[ |R_{\text{v}} \cdot |S_0| \right]^{N+\nu}
$$

(5.263)

$$
= \frac{S_x}{s_0}
$$

(5.264)

$$
= \frac{\mathcal{E} \cdot X_0}{\gamma_0 \cdot \|h\|^2}
$$

(5.265)

$$
= \gamma_0 \cdot \text{SNR}_{\text{mfb}}
$$

(5.266)

$$
= 2^2 F(x(D); y(D))
$$

(5.267)

The unbiased SNR is as always $\text{SNR}_{\text{cdf,u}} \to \text{SNR}_{\text{MMSE-DFE,u}} = \text{SNR}_{\text{MMSE-DFE}} - 1$. An overall geometric SNR characterizes the DFE set, as in Section 3.12.

**Some Final Comments:**

1. All asymptotic results hold $\nu$ as a constant (correctly because it is a constant). The corresponding “wasted dimensions” and any energy “wasted” in the cyclic prefix clearly then asymptotically zeroes as block length $N$ increases.

2. Error propagation may occur within each symbol in all GDFE’s, including the CDFE. Shorter blocks, $N \leq \infty$, can thus limit error bursts, while the MMSE-DFE has a potential for infinite catastrophic bursts. With $\Gamma \to 0$ dB, independent codes on each dimension, and infinite decoder delay, error propagation $\to 0$.

3. A designer could pick $N$ too large, much larger than is necessary to approach closely infinite-length results, thus resulting in a huge unnecessary complexity for no performance gain. Complexity comparisons where DMT (VC) perform at lower complexity than CDFE (GDFE, respectively) are

---

48 When $L-x > 1$, this becomes an $L_x \times L_x \frac{G}{G(D)}$ on block rows.

49 $S_0$ becomes an $L_x \times L_x$ matric to which VC can be applied, or simply absorbed into the matrix factors of $G(D)$, leaving a diagonal $L_x \times L_x$ matrix $S_0$. 

884
made in this same good-design context. Obviously, the channel $H(D) = 1$ works equally well with all methods with $N = 1$ and $\nu = 0$. As ISI increases, there is a judicious choice of $N$ that allows infinite-length performance without infinite-complexity in either DMT or CDFE. At this choice, DMT complexity is $\log_2(N)$ per sample while CDFE (or MMSE-DFE) complexity is upper bounded by $N$ and typically between $\log_2(N)$ and $N$, typically about $3 \cdot \log_2(N)$ ignoring modulation or interpolation complexity unique to CDFE, and about $5 \cdot \log_2(N)$ if those are included.

4. Any gap greater than 0 dB leads to the CDFE having a lower SNR than DMT because the CDFE is designed and optimized for 0 dB gap, while the DMT (because the feedback section is 0) simultaneously minimizes MSE, as well as maximizes product SNR, at all gaps. Beware of statements to the opposite regarding an “SNR averaging” property of the DFE – correct use of codes and gaps deviates from this purported property (and it is misunderstanding in codes’ use that leads to many misconceptions in performance equivalence).

5. DMT is a maximum-likelihood detector and is thus both canonical and optimal. The CDFE is only canonical and does not have ML-detector performance, and as well may have error propagation.

6. With careful modulation and interpolation implementation (not shown in this text), some designs may reduce delay from input to output in a CDFE system with respect to DMT.

7. C-OFDM achieves the performance levels of DMT or CDFE only when the code has zero dB gap and the $R_{xx}$ is the same. This result has significance in time-varying channels (i.e., some wireless applications) where it is not possible to convey to the transmitter the best transmit spectrum to use either because the channel is broadcast only (unidirectional, so no return channel to advise transmitter design) or because the channel changes so fast that the input dynamic spectra could not follow the channel. Thus, a fixed $R_{vv} = I$ has a certain $I$ that corresponds a certain performance level (in the absence of channel singularity) of both OFDM and MMSE-DFE systems if gap=0 dB codes are used in both cases. The $I$ will need to be averaged over all channels and thus bound the data rate of the code. Thus, in time-varying channels, there can be equivalent performance of Coded-OFDM system and a QAM-MMSE-DFE system, other than possibly the receiver’s ability to track channel changes (which is likely going to favor the multicarrier system for which each band is separately equalized by a single coefficient, rather than a complicated matrix system with many more coefficients ($N^2$ versus $N$). For this reason, most wireless systems use C-OFDM instead of DMT.

8. DMT systems do not require an optimum transmit filter matrix, and use the IFFT – a known and fixed transmit partitioning system that can be designed once. CDFE systems alone use an optimum transmission filter (as well as $R_{xx}$ that is common to DMT and CDFE) that is a consequence of the channel and thus must be implemented adaptively. MIMO VC systems, however, will need to specify a transmit matrix, at least one for each $L_x$-dimensional tone.

9. High-performance CDFE systems often exhibit the same “Gaussian” like high peak-to-average ratio of DMT systems, this is particularly true as ISI becomes severe, or equivalently with large $N$. Simply put, this is a function of the “central limit theorem” and the transmit signal being simply a sum of many independent random variables’ contributions, and thus approaches Gaussian. PAR-reduction methods, such as those of Section 4.10 for DMT and OFDM, have not been studied for CDFE or MMSE-DFE (which shares the high PAR problem). Such studies should investigate at least 4x the sampling rate of the transmission system to be of practical significance (often academic researchers forget that it is the analog PAR that is important, not the sample-time-only PAR), see Section 4.10.

10. The MMSE-DFE MS-WMF notching ability that occurs when the noise-equivalent ISI channel has notches is absolutely NOT equivalent to the use of multiple bands. This statement appears throughout the literature on MMSE-DFEs and is simply incorrect every single time it has been said. A single MMSE-DFE will perform worse than multiple MMSE-DFEs on each side of all the notches - no exceptions.

---

50 This presumes the channel-gain distribution $g = |h|^2/\sigma^2$, $p_g$, is stationary.
Ideal MMSE-DFE is lower bound for ML Performance

The ideal MMSE-DFE (or ideal MMSE-GDFE) upper bounds the performance $P_e$ of a maximum likelihood detector in that the sequence-error probability is always lower for ML. The ideal MMSE-DFE (and GDFE) assume(s) that all previous decisions are correct. Such correct previous decision making may not be the case in practice, so at least there is a chance this ideal GDFE system’s analysis could outperform the optimum detector because of the potentially false presumption of no decision errors. If results include error propagation; this theoretical anomaly cannot occur.

In short, Claude Shannon had it right as with so many other of his basic simple and information-age-envisioning results – a multi-carrier system is optimum for handling ISI.
5.4 The Gaussian MAC and GDFE

GDFE’s directly apply to Chapter 2’s Gaussian MAC. The GDFE’s canonical performance, with most discrete signal processing at the receiver, well matches the MAC’s separate-user-location constraint of a block diagonal $R_{\mathbf{xx}}$. Section 2.7’s MMSE MAC now has an additional GDFE interpretation, in which the MAC’s transmitter input, $\nu$ with $R_{\nu\nu} = I$, excites block diagonal $R_{\mathbf{xx}}^{1/2}$, equivalently $\mathbf{x} = R_{\mathbf{xx}}^{1/2}(u) \cdot \nu$. Any such (block) diagonal channel input $R_{\mathbf{xx}}$ can, and usually does, have nonzero-energy components in the channel null space $N_{\tilde{H}}$. These lost components reduce $I(\mathbf{x};\mathbf{y})$’s maximum value relative to its largest single-user GDFE values for the same $\tilde{H}$ and a sum-energy (that is like a single-user) constraint. This chapter’s single-user GDFE zeroes the $N_{\tilde{H}}$ components through Section 5.1’s non-singular discrete-modulator input construction. This removal is usually not feasible with a MAC. The MAC’s discrete modulator $A$ thus has loss with respect to single-user canonical performance, which this section also characterizes. The corresponding reduced-GDFE $I(\mathbf{x};\mathbf{y})$ is the MAC’s rate sum. The MAC’s highest rate sum uses Subsection 2.7’s Simultaneous Water Filling (SWF), which has performance bounded by single-user water filling.

This section finds best energy allocations (and associated information distribution $\{I_u\}_{u=1,...,U}$) for all points in capacity region $C_{MAC}(b)$. Subsection 5.4.1 details specific differences and similarities of the MMSE MAC and the GDFE and characterizes the MAC loss with respect to single user for the same $(H, R_{nn})$ pair or channel. Subsection 5.4.2 details more completely Section 4.6’s Vector DMT channel decomposition into a tone-indexed matrix-AWGN set, ultimately leading to tonal GDFE’s - one for each “tone.” Subsection 5.4.3 progresses to an alternative weighted-energy-sum characterization of $C_{MAC}(b)$ and an algorithm (along with software) to find specific “best” energy vectors $E_x$ (or autocorrelation matrices $R_{\mathbf{xx}}$) that correspond to a desired multi-user rate vector $b$. There can be multiple “best” solutions for points in $S(b)$; while boundary points in $C(b)$ have more distinct specific optima. The associated GDFE structure then completes the optimal MAC design. This design process can also assist Section 5.5’s optimal BC design through the vector-channel expansion of Section 2.8’s duality.

5.4.1 The GDFE’s Relationship to the MMSE MAC

Chapter 2’s MMSE MAC receiver appears very similar to the GDFE. However, there are subtle notational differences: Both the MMSE MAC’s $\nu$ and the MAC-GDFE’s $v$ have diagonal $R_{\nu\nu} = R_{vv} = I$ with any per-user-energy/autocorrelation-matrix scalings absorbed into the channel matrix such that

$$R_{\mathbf{fMAC}} = A^* \cdot \tilde{H}_GDFE^* \cdot \tilde{H}_GDFE \cdot A.$$  \hfill (5.268)

The GDFE’s $R_{\mathbf{xx}}$ is block diagonal with corresponding block-diagonal square roots that shape a white input $R_{\nu\nu} = I$ to the channel input $R_{\mathbf{xx}}$. Also the MMSE MAC’s $\nu$ and the GDFE’s $v$ differ in that $v$ best contains no channel-null-space components, so the MAC almost always has lower sum rate and lower corresponding SNR, than the same channel $(H, R_{nn})$ as single user. Another difference is that the GDFE does not have per-dimensional energy constraints; leaving the energy-sum MAC closer conceptually to the GDFE.

**EXAMPLE 5.4.1** [Comparison of a simple degraded MAC and GDFE] A matrix AWGN channel has $\sigma^2 = .01$ and $H = \begin{bmatrix} 8 & 5 \end{bmatrix}$. The MMSE MAC receiver follows Section 2.7’s developments:

```matlab
>> H=[80 50];
>> Au=sqrt(1/2)*eye(2);
>> Usize=[1 1];
>> cb=2;
>> [Bu, GU, WU, S0, MSWMFU] = mu_mac(H, Au, Usize , cb)
Bu =
    5.8222    0.2378
```

51 Complex dimension (or real dimension for the one baseband DC dimension at $n = 0$).

52 There can also be square roots that are not block diagonal, but are not of interest for the MAC.
\[
GU = \begin{bmatrix}
1.0000 & 0.6250 \\
0 & 1.0000
\end{bmatrix}
\]
\[
WU = \begin{bmatrix}
0.0003 & 0 \\
-1.6000 & 2.5608
\end{bmatrix}
\]
\[
S0 = 1.0e+03 \ast \begin{bmatrix}
3.2010 & 0 \\
0 & 0.0014
\end{bmatrix}
\]
\[
MSWMFU = \begin{bmatrix}
0.0177 \\
0.0283
\end{bmatrix}
\]
\[b_{\text{sum}} = \text{sum}(Bu) = 6.0600\]

The MMSE MAC expressions essentially are those of the GDFE, but without the null-space removal. The MMSE MAC calculations consequently result in a lower mutual information because some energy inevitably must be lost in this MAC’s null space (if both users are nonzero). The MMSE MAC clearly has primary user 2, corresponding to the channel with transfer 80. If all energy (1 unit in this case) is on this primary user, the maximum rate sum is
\[0.5 \log_2 (1 + 6400) = 6.3220\]

The input autocorrelation that corresponds to the maximum rate sum is trivially SWF on the one used dimension
\[
R_{xx}^o = \begin{bmatrix}
1 & 0 \\
0 & 0
\end{bmatrix}
\]

The single-user GDFE instead first removes the energy in the null space, and has performance computed by VC or any other of the equivalent GDFE forms:
\[\text{>> } [F,L,M] = \text{svd}(H)\; ;
\]
\[
F = 1 \\
L = 94.3398 \\
M = \begin{bmatrix}
0.8480 & -0.5300 \\
0.5300 & 0.8480
\end{bmatrix}
\]
\[0.5 \log_2 (1 + 0.5 \cdot L(1,1)^2) = 6.0600\]

A single-user-best GDFE design places 1 energy unit on the nonsingular mode, so then
\[
u = \begin{bmatrix}
0.848 \\
0.530
\end{bmatrix} \cdot v
\]

and \(R_{uu} = 1\). However, (5.270) tacitly requires coordination of the two inputs because \(v\) must enter both channel inputs. This entry to both channel inputs is not possible with the MAC, but is with the single-user GDFE.

A single-user-best GDFE data rate is \(I(x; y)\) or
\[0.5 \log_2 (1 + L(1,1)^2) = 6.5599\]

which is higher than the energy-sum MAC’s best data rate. There is thus a MAC loss relative to the single-user situation for the same \(\tilde{H}\) and \(E_x = 1\). That overall single user to MAC reduction is
\[b_{\text{MAC}} = 6.32 < 6.56\]

888
The energy loss is

\[
\gamma_{MAC} = 10 \cdot \log_{10} \frac{2^2 \mathcal{I}(x; y)}{2^2 b_{MAC} - 1} - 1 = 1.45 \text{ dB}, \tag{5.272}
\]

so the energy-sum MAC’s best rate sum can never exceed the maximum single-user \(\mathcal{I}(x; y)\) possible on the same channel \(H\) with the same (energy-sum for MAC) input energy constraint.

Example 5.4.1 then suggests the following MAC-loss measure caused by the MAC’s need for no input coordination:

**Definition 5.4.1** [MAC loss relative to single user] The MAC loss for any given matrix AWGN channel \([H, R_{nn}]\) relative to the single-user for that same channel is

\[
\gamma_{MAC} = \frac{2^2 \mathcal{I} - 1}{2^2 \mathcal{I}_{sum-MAC} - 1} \tag{5.273}
\]

*The numerator uses the single-user water-filling apacity \(C\) for the same channel, \([H, R_{nn}]\).*

**Gaussian MACs with variable dimensionality inputs:** Example 5.4.1 uses Chapter 2’s mu_mac.m (see also Appendix E) program

```matlab
function [Bu, GU, WU, S0, MSWMFU] = mu_mac(H, AU, Usize, cb)

--------------------------------------------------------------------------
Inputs: Hu, A, Uind
Outputs: snrGDFEu, Gub, Wub, S0, MSWMFU

H: noise-whitened channel matrix \([H1, ... , H1]\)
AU: Block Diag square root discrete modulators, blkdiag([AU, ... , A1])
Usize: # of dimensions for each user U \(1\) in \(1 \times U\) row vector
cb: = 1 if complex baseband or 2 if real baseband channel

G: feedback matrix
GU: unbiased feedback matrix
W: feedfoward linear equalizer
WU: unbiased feedforward linear equalizer
S0: sub-channel channel gains
MSWMFU: unbiased mean-squared whitened matched filter
Bu - user u’s bits/symbol

the user should recompute SNR if there is a cyclic prefix

This mu_mac program allows variable input user dimensionality through the Usize input, as per this example:

**EXAMPLE 5.4.2** [Split-dimensionality MAC] A 2-user channel has user 2 with 1 dimension, but user 2 with 2 dimensions. The matrix channel (noise whitened) and input autocorrelation functions appear below, along with the multi-user mac design.

```
5.4.1 MACs with intersymbol interference

The MAC with ISI recursively embeds the GDFE structures into Chapter 2’s MAC with finite-length time-frequency symbols that each have intersymbol interference, as well as crosstalk. Figure 5.25 re-illustrates this nested-GDFE-of-MMSE-DFE’s concept. With $x_U(D)$ being the vector $D$-Transform (See Appendix D) of the $U$-multiple users’ inputs, with $U$ at the top, the chain rule expands to

\[
I(x(U(D); y(D)) = \sum_{u=1}^{U} I(x_u(D); y(D)/x_{u-1}(D)) .
\] (5.274)

Figure 5.25’s receiver decodes first user 1 with all the other users as noise. Then this receiver removes user 1’s effect from the received signal $y(D)$ and decodes user 2. The MMSE-DFE design for user 1 proceeds as in Chapter 3, but adds all other users’ received spectra to the noise spectrum. If the other users have $\Gamma = 0$ dB (i.e., close to Gaussian/AEP) codes, this approach reliably achieves a first-user data rate of $I(x_1(D); y(D))$. The receiver then filters the first MMSE-DFE’s consequent ($\Gamma = 0$ dB) error-free decisions $x_1(D)$ by a replica of the channel $h_1(D)$ and subtracts it from $y(D)$. This implies infinite delay, but captures concept. This new signal enters a second MMSE-DFE for user 2 that treats users $3, ..., U$ as noise. The design then recursively repeats to remove successively each user’s effect upon remaining users. If there were no memory in the channel (thus no ISI within nor crosstalk among any of the users), each MMSE-DFE degenerates into Chapter 2’s simple scalar $1 \times U$ MAC.
Figure 5.25’s MAC with $\mathcal{N} \to \infty$ has $SNR = 2^{2\mathcal{I}(x(D);y(D))} - 1$ and is canonical, and the $\mathcal{I}(x(D);y(D))$ can correspond to a chain-rule implementation for any order. Instead for $\mathcal{N} < \infty$, a GDFE can replace Figure 5.25’s MMSE-DFE, which the following example illustrates:

**EXAMPLE 5.4.3 [Two-user Nested GDFE’s for ISI]** Two synchronized users use a guard period of $\nu = 1$ sample with $N = 2$ output samples on the two independent paths of a real baseband MAC with $H_2(D) = 1 + .9D$ and $H_1(D) = 1 - D$. The white noise has variance $\sigma^2 = .181$ and added as two dimensional noise to the two channel outputs’ sum for each 2-sample packet at the common MAC receiver. These are no longer one-dimensional channels but are symbol-length approximations to the original MAC. There are 6 input dimensions, 3 for each user and 2 output dimensions. Nonetheless, the GDFE receiver applies for any set of input user-autocorrelation matrices. This example first examines the situation of 6 independent input dimensions, each of energy 1 per sample:

```matlab
>> H = [H2 H1]
    1.0000  0.9000  0  1.0000 -1.0000  0
    0  1.0000  0.9000  0  1.0000 -1.0000
>> bsum = (.5/log(2))*log((det(H*eye(6)*H’+.181*eye(2))/det(.181*eye(2))))
```

4.4622

The receiver GDFE decodes user 1 first on each of its 3 dimensions. The sum rate 4.4622 is not the highest possible, but corresponds to the choice of $R_{xx} = I$ for the two users. While the users’ inputs cannot coordinate, each user still unnecessarily wastes energy in the channel’s respective null spaces. GDFE design continues as:

```matlab
>> H = (1/sqrt(.181))*H;
Usize = [3 3];
cb = 2;
>> [Bu, GU, WU, S0, MSWMFU] = mu_mac(H, eye(6), Usize, cb)
Bu = 3.2945 1.1677
```
As expected, the mutual information is the sum of the data rates. User 1 (decoded first) is at a significant disadvantage in terms of data rate and SNR that is evident. The order of the two users can reverse to get a different decomposition of bit rates where user 2 is at a disadvantage, as per Chapter 2.

>> [Bu, GU, WU, S0, MSWMFU] = mu_mac([H(:,4:6) H(:,1:3)] , eye(6), Usize , cb)
Bu =  3.4207  1.0416
GU =
     1.0000  -1.0000  0  1.0000  0.9000  0
     0  1.0000  -0.8671 -0.1329  0.7475  0.7804
     0  0  1.0000  -0.4585 -1.4127 -0.9000
     0  0  0  1.0000  1.3585 0.4127
     0  0  0  0  1.0000  0.5443
     0  0  0  0  0  1.0000
WU =
     0.1810  0  0  0  0  0
     -0.1227  0.1610  0  0  0  0
     0.5023 -0.6591  0.9558  0  0  0
     -1.0000 -0.4480  0.4068  1.5842  0  0
     0.4263 -0.1901 -0.5164  0.4263  0.7586  0
     0.0251  1.0226  0.9000  0.0251  0.9749 2.9885
S0 =
     6.5249  0  0  0  0  0
     0  7.2107  0  0  0  0
     0  0  2.0463  0  0  0
     0  0  0  1.6312  0  0
     0  0  0  0  2.3182 0
     0  0  0  0  0  1.3346
MSWMFU =
     0.4254  0
     0.0522  0.3785
    -0.2137  0.4727
     0.4254 -0.1923
    -0.1814  0.2441
    -0.0107 -0.4254
>> bvec=diag((0.5/log(2))*log(S0))
    1.3530
    1.4251
     0.5165
     0.3530
     0.6065
     0.2082
>> sum(bvec) = 4.4622 (checks)
S0 =

\[
\begin{bmatrix}
6.5249 & 0 & 0 & 0 & 0 & 0 \\
0 & 7.3716 & 0 & 0 & 0 & 0 \\
0 & 0 & 2.3841 & 0 & 0 & 0 \\
0 & 0 & 0 & 1.5806 & 0 & 0 \\
0 & 0 & 0 & 0 & 2.1364 & 0 \\
0 & 0 & 0 & 0 & 0 & 1.2548 \\
\end{bmatrix}
\]

MSWMFU =

\[
\begin{bmatrix}
0.4254 & 0 \\
-0.0565 & 0.3689 \\
-0.1951 & -0.4254 \\
0.4254 & 0.1951 \\
0.1868 & 0.2573 \\
0.0118 & 0.4727 \\
\end{bmatrix}
\]

>> bvec=diag((0.5/log(2))*log(S0))

\[
bvec =
\begin{bmatrix}
1.3530 \\
1.4410 \\
0.6267 \\
0.3302 \\
0.5476 \\
0.1638 \\
\end{bmatrix}
\]

>> sum(bvec) = 4.4622

Perhaps of yet more interest would be the case where the design zeroes energy in the individual null spaces, \( N_1 \) and \( N_2 \). A better design instead reallocates the saved energy to the pass spaces \( P_1 \) and \( P_2 \) equally:

\[
\begin{bmatrix}
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
0 & 0 \\
\end{bmatrix}
\]

% This F1 is essentially the same as F2, a happy coincidence that might be exploited with simplifications that could reduce feedback with carefully designed input.

Bu = 3.8233 1.2019
The sum rate is higher because the $R_{xx}$ changed, and the revised inputs each avoid their individual channels’ null spaces.

### 5.4.2 Vector DMT for the MAC

Section 4.7 introduced Vector DMT for a linear time-invariant channel. Figure 5.26 illustrates DMT-symbol synchronization at a common receiver.
Each transmitter uses the same length \((\mathbf{N} + \nu)\) DMT symbol and offsets transmit symbol boundaries so that all users arrive at the common receiver symbol boundary. Such alignment presumes the system supplies a common symbol clock to all MAC transmitters (essentially telling these transmitters to advance or delay their transmit symbol boundary until all align at the common MAC receiver). Section 4.6’s digital duplexing or “zippering” allows such alignment.\(^{53}\) Transmitter \(u\) has \(L_{x,u}\) IFFT’s, one at each of the \(U\) users’ transmitters, and \(\mathbf{L}_x = \sum_{u=1}^{U} L_{x,u}\). This \(\mathbf{L}_x\) is common\(^{54}\) to all tones \(n\). There are \(L_y\) FFT’s implemented at the common receiver. Such synchronization occurs with cyclic extension length that satisfies \(\nu T’ \geq \max_{u,u’}\{\text{length}(h_{u,u’}(t)) + \Delta_u\}\), which leads to no intersymbol interference. This alignment also ensures that any tone’s crosstalk is a function ONLY of other users’ signals on that same tone \(n\).

**Tonal Channels:** Figure 5.27’s \(L_y\) receiver-FFT outputs jointly satisfy

\[
\mathbf{Y}_n = \mathbf{H}_n \cdot \mathbf{X}_n + \mathbf{N}_n, \quad n = 0, \ldots, \mathbf{N} - 1
\]  

(5.275)

where

\[
\mathbf{H}_n = [H_{U,n} \ldots H_{1,n}]
\]  

(5.276)

\[
\mathbf{X}_n = \begin{bmatrix}
X_{U,n} \\
\vdots \\
X_{1,n}
\end{bmatrix}
\]  

(5.277)

\(^{53}\)This digital duplexing also allows simultaneous synchronization in the opposite-direction BC, as in Section 5.5.

\(^{54}\)Tones with zeroed dimensions and correspondingly zeroed (dummy) inputs may simplify concept and mathematical description, but of course do not improve \(I\) nor SNRs.
Often $L_{x,u} = L_x$ in practice, although MAC “uplink” users need not have the same number of transmit antennas/dimensions. There is an individual GDFE for each tone in the Vector DMT/OFDM case.

**Energy and autocorrelation-matrix constraints:** The tonal autocorrelation matrix is

$$R_{XX}(u, n) = \mathbb{E}[X_{u,n} \cdot X_{u,n}^*] .$$

This is essentially a 4-dimensional tensor\textsuperscript{55} with square $L_{x,u} \times L_{x,u}$ matrix for each value of $u = 1, \ldots, U$ and $n = 0, \ldots, N - 1$. Similarly, the $(l_y, l_x)^{th}$ entry of $H_{u,n}$ is DFT-tone-$n$’s transfer gain/phase from line/antenna $l_x$ of user $u$ to line/antenna $l_y$ of the common output, where $l_x = 1, \ldots, L_{x,u}$ and $l_y = 1, \ldots, L_y$. The energy constraints become

$$\sum_n \text{trace} \{R_{XX}(u, n)\} \leq E_u \quad \forall u = 1, \ldots, U ,$$

which entries $E_u$ comprise the energy vector $\mathbf{E}$. For an energy-sum MAC, the individual user energies are variable and need only satisfy

$$\sum_{u=1}^U E_u \leq \mathbf{E}_x .$$

\textsuperscript{55}A matlab object array dimensioned as $(L_x,u \times L_x,u, U, n)$ where size can be read for each $u = 1, \ldots, U$ and $n = 0, \ldots, N-1.$
**Tonal GDFE:** Figure 5.28’s tone-indexed model, with an individual GDFE for each tone, simplifies greatly full successive decoding (or GDFE) structure across all frequency and space dimensions. Essentially that structure now applies individually to each tone. Effectively, $\bar{N}$ GDFEs of size $L_y \times L_x$ replace a giant GDFE of size $[L_y \cdot \bar{N}] \times [L_x \cdot \bar{N}]$. The complexity reduction is enormous in application with large $\bar{N}$.

**Encoders, Decoders, and Delay:** Codes for any user can apply dimensionality over tones and of course time, and C-OFDM must do this, to reduce the delay. Each user must however have its own code. The user bits/symbol has several useful forms:

\[
b_{u,\ell} = \sum_{n=0}^{N-1} b_{u,\ell,n} \quad (5.283)
\]

\[
b_{u,n} = \sum_{\ell=1}^{L_x,u} b_{u,\ell,n} \quad (5.284)
\]

\[
b_u = \sum_{\ell=1}^{L_x,u} \sum_{n=0}^{N-1} b_{u,\ell,n} \quad (5.285)
\]

\[
b = \sum_{u=1}^{U} \sum_{\ell=1}^{L_x,u} \sum_{n=0}^{N-1} b_{u,\ell,n} \quad . \quad (5.286)
\]
5.4.2.1 Tonal GDFE Specification

Figure 5.28 shows the GDFE for any input-autocorrelation-matrix set \( \{R_{XX}(u,n)\} \). To implement the tonal GDFE with known \( R_{XX}(n) = A_n \cdot A_n^* \) and identity input \( R_{VV}(n) = I \), the receiver forms

\[
Z_n = \left[ A_n^* \cdot H_n^* \cdot R_{NN}^{-1}(n) \right] \cdot Y_n
= A_n^* \cdot H_n^* \cdot R_{NN}^{-1}(n) \cdot H_n \cdot A_n \cdot V_n + N'_n
= R_{f,n} \cdot V_n + N'_n\]  

(5.287)

(5.288)

(5.289)

a forward canonical channel for tone \( n \) across all MAC users. The entity \( R_{noise}(u) \) does not appear explicitly in the GDFE design, which instead uses directly \( R_{f,n} \). The given GDFE order, which the ordering of inputs in \( V_n \) implies, tacitly assigns later users (higher-indexed dimensions) as uncanceled crosstalk noise. Different orders then simply correspond to re-indexing the GDFE dimensions. There are \( N \) such forward canonical channels that act independently\(^{56} \), one for each tone \( n = 0, \ldots, N - 1 \). The canonical backward channel has Cholesky factorization

\[
R_{b,n}^{-1} = R_{f,n} + I = G_n \cdot S_{0,n} \cdot G_n^*\]  

(5.290)

\(^{56}\)Apart from any outer code’s potential application across \( n \) to reduce implementation delay for a given \( \Gamma > 0 \) dB.
for each tone. The MAC order \( \pi \) is the same for all tones. The upper triangular Cholesky factor \( G_n \) determines that tone’s feedback section. The diagonal matrix \( \text{SNR} \) contains the biased SNR’s for the estimation of all users’ dimensions and is

\[
\text{SNR}_n = S_{0,n} .
\]

The unbiased feedforward section that processes the channel output vector \( Y_n \) is

\[
W_{n,\text{unb}} = [\text{SNR}_n] \cdot [\text{SNR}_n - I]^{-1} \cdot S_{0,n}^{-1} \cdot G_n^{-*} \cdot A_n^* \cdot H_n^* \cdot R_{NN}^{-1}(n) .
\]

The unbiased\(^{57} \) feedback section is

\[
G_{n,\text{unb}} = I + \text{SNR}_n \cdot [\text{SNR}_n + I]^{-1} \cdot (G_n - I) .
\]

When the channel rank is \( \mathcal{L}_x = U' \), then there is no “other user” noise\(^{58} \) and each (consequently primary) user occupies its own dimension(s). This corresponds to only primary-user’s dimensional use on tone \( n \). When the rank is less, other (necessarily secondary) users naturally become significant constituents of most dimensions’ error signals in Figure 5.28. Decisions occur for each element of the vector \( V_n \) in succession from bottom to top (user \( U \)) according to the users’ order. A GDFE structure exists for each and every order, which from a single-user GDFE perspective simply re-indexes dimensions and thus changes nothing fundamentally for multi-user, the components of \( b \) can vary with order, but \( 1^* \cdot b = b \) for all orders.

EXAMPLE 5.4.4 [Revisit ISI Channel Example] This revisits Example 5.4.3. This ISI channel does have a guard period so any SNR calculations should use an exponent of \( 1/N_x = 1/6 \). A Vector DMT system could be used with equal energy on each of \( N = 8 \) tones. Because such a system is not quite as good as vector-coded (or GDFE based on same input \( R_{xx} \) as vector-coded), the performance might be expect to approach Example 5.4.3 original design’s data rate as \( N \) increases.

\[
\begin{align*}
\text{h0} &= [1 \ 1]; \\
\text{h1} &= [.9 \ -1]; \\
N &= 8; \\
H &= (1/\sqrt{.181}) \ast \text{fft(h, N, 3)}
\end{align*}
\]

In this case, the channel is real. The energy normalization by \( N = 8 \) corresponds to ensuring the channel has no energy gain; however with constant sampling rate, the input energy of 1 unit per sample accumulates over \( N = 8 \) successive subsymbols/samples, thus \( R_{xx} = 1 \) for each symbol or \( A_n = 1 \). The cb of mu \( \text{mac} \) should be set to 2 for this real baseband example.

\[
\begin{align*}
\text{cb} &= 2; \\
\text{Usize} &= [1 \ 1]; \\
\text{nu} &= 1; \\
\text{A} &= \text{zeros}(2,2,N); \\
\text{for n=1:N} & \\
\text{A}(:,:,n) &= \sqrt{8/9} \ast \text{eye}(2); \\
\text{end} \\
\text{b} &= \text{zeros}(2,8); \\
\text{GU} &= \text{zeros}(2,2,8); \\
\text{WU} &= \text{zeros}(2,2,8); \\
\text{S0} &= \text{zeros}(2,2,8); \\
\text{MSWMFU} &= \text{zeros}(2,1,8); \\
\text{for n=1:N} & \\
\text{end}
\end{align*}
\]

\(^{57}\) A superscript of “unb” is used to denote “unbiased” to avoid confusion with the use of \( U \) as a user index.

\(^{58}\) In a MMSE sense.
The last calculation adjusts for the original vector coded system being 3 dimension, while this system is 8 dimensional, really 9 dimension with cyclic prefix penalty. The data rate is higher because more dimensions makes the 2/3 dimensional loss in the original system look large. The feedback and MSWMFU are

```
>> GU
GU(:,:,1) =
    1.0000 + 0.0000i   -0.3991 + 0.0000i
     0.0000 + 0.0000i    1.0000 + 0.0000i

GU(:,:,2) =
     1.0000 + 0.0000i   -0.2928 + 0.0737i
     0.0000 + 0.0000i    1.0000 + 0.0000i

GU(:,:,3) =
     1.0000 + 0.0000i    0.0931 + 0.1414i
     0.0000 + 0.0000i    1.0000 + 0.0000i

GU(:,:,4) =
     1.0000 + 0.0000i    0.9056 + 0.1552i
     0.0000 + 0.0000i    1.0000 + 0.0000i

GU(:,:,5) =
     1.0000 + 0.0000i    1.5833 + 0.0000i
     0.0000 + 0.0000i    1.0000 + 0.0000i

GU(:,:,6) =
     1.0000 + 0.0000i    0.9056 - 0.1552i
     0.0000 + 0.0000i    1.0000 + 0.0000i

GU(:,:,7) =
     1.0000 + 0.0000i    0.0931 - 0.1414i
     0.0000 + 0.0000i    1.0000 + 0.0000i

GU(:,:,8) =
     1.0000 + 0.0000i   -0.2928 - 0.0737i
     0.0000 + 0.0000i    1.0000 + 0.0000i
```

```
>> MSWMFU
MSWMFU(:,:,1) =
     0.2375 + 0.0000i
     0.0000 + 0.0000i

MSWMFU(:,:,2) =
     0.2395 + 0.0932i
     0.2256 - 0.5447i

MSWMFU(:,:,3) =
     0.2493 + 0.2244i
     0.2256 - 0.2256i

MSWMFU(:,:,4) =
     0.3054 + 0.5346i
     0.2256 - 0.0935i

MSWMFU(:,:,5) =
     4.5125 + 0.0000i
     0.2256 + 0.0000i
```
Example 5.4.4 has all user inputs as the same size (1 antenna each). Variable size inputs follow through the selection of the Usize parameter; the dimensionality is the same on all tones.

5.4.2.2 SWF for the Vectored DMT MAC

Simultaneous water-filling (SWF) for Vector DMT follows Section 2.7’s maximum rate sum. Any user u water-fills with all other users’ crosstalk and its noise in autocorrelation matrix

\[ R_{\text{noise}}(u, n) = \sum_{i \neq u} H_{i,n} \cdot R_{XX}(i, n) \cdot H_{i,n}^* + R_{NN} \cdot \] (5.294)

(5.294)’s equivalent-white-noise channels thus become

\[ \tilde{H}_{u,n} = R_{\text{noise}}^{-1/2}(u, n) \cdot H_{u,n} \text{ with user-specific SVD} \] (5.295)

\[ = F_{u,n} \cdot \Lambda_{u,n} \cdot M_{u,n}^* \] (5.296)

The SWF energy distribution has each user satisfy

\[ E_{u,n,\ell} + \frac{1}{g_{u,\ell,n}} = K_u \quad \forall \quad u, n, \ell \quad \text{with } g_{u,\ell,n} = \lambda_{u,\ell,n}^2 \] (5.297)

Each user has its own water level \( K_u \). An \( L_{x,u} \times L_{x,u} \) water-fill input autocorrelation matrix for user \( u \) follows from SVD’s \( M_{u,n} \) as

\[ \{ R_{XX}(u, n) \mid R_{XX}(u, n) = M_{u,n} \cdot \text{Diag} \cdot (E_{u,n}) \cdot M_{u,n}^* \forall \quad n = 0, ..., N - 1 \quad u = 1, ..., U \} \] (5.298)

Such a set \( \{ R_{XX}(u, n) \} \) can follow from Section 2.7’s iterative water-filling that determines successive user energies and channel gain using \( R_{\text{noise}}(u, n) \) in a convergent algorithm. An energy-sum MAC’s maximum sum rate’s achievement energizes only primary (sub)users; these primary (sub)users correspond to the best \( U^o \) dimensions available for the different users (each across its best frequencies), as in Section 2.8. Usually with reasonably large \( N \), there are sufficient nonsingular dimensions (sufficiently high channel rank \( \sum_n \tilde{R}_{n}^* \)) so that the number of primary users \( U^o = U \). Each user water-fills energy with respect to all others as noise over dimensional indices \( n \) and \( \ell \) to water level \( K_u \). Other non-water-fill energy loading\(^{59} \) follows similarly and treats other users as noise in determining energy per dimension following the loading strategy/criteria. \( E_u = \sum_n E_{u,n} \) defines the diagonal energy matrix entries \( \text{Diag}(E_u) \). When \( L_{x,u} > 1 \), then the block-diagonal matrix \( R_{xx}(u, n) \) generalizes to \( \text{trace}\{ \sum_n R_{xx}(u, n) \} \leq E_u \) for user \( u \).

Each user’s bits/symbol \( b_u \) can vary with order. But the overall rate sum remains

\[ b = \sum_{u=1}^{U} b_u = \sum_{u=1}^{U} \sum_{n=0}^{N-1} b_{u,n} \] (5.299)

with the same maximum for all \( U! \) orders if an energy-sum MAC.

\(^{59}\) such as Chapter 4’s Levin Campello

901
Lemma 5.4.1 [Existence of the orthogonal-division multiplexed (ODM) point]
When \( L_x = 1 \), there exists a rate-sum vector \( b_{\text{max}} \) for which all orders provide the same \( \{b_u\}_{u \in U} \)-set - an “FDM point,” as \( N \to \infty \). When \( L_x > 1 \), then such an ODM point also exists when \( U^o = U = \leq \bar{\rho}_H \) where \( \bar{\rho}_H = \sum_{n=0}^{N-1} \bar{\rho}_{H_n} \). When \( N < \infty \) and \( U^o = U \), there is an SWF orthogonal-dimension-multiplexing point where each primary user occupies solely its own spatial dimensions with no crosstalk from other users.

Proof: The SWF optimization decomposes into \( U^o \) separate problems each has lowest possible (zero crosstalk) noise when they occupy a single dimension and trivially is then also SWF, as long as \( U^o = U \leq \bar{\rho}_H \). This latter condition certainly holds if \( N \to \infty \) (at constant sampling rate). QED.

The Matlab SWF.m program  Chapter 2 first introduced SWF.m, which this section revisits now with \( N \geq 1 \).

function [Rxx, bsum , bsum_lin] = SWF(Eu, H, user_ind, Rnn, cb)

Simultaneous water-filling MAC max rate sum (linear and nonlinear GDFE)
The input is space-time domain h, and the user can specify a temporal block symbol size \( N \) (essentially an FFT size).

Inputs:
Eu U x 1 energy/SAMPLE vector. Single scalar equal energy all users
any \( (N/N+nu) \) scaling should occur BEFORE input to this program.
H The FREQUENCY-DOMAIN Ly x sum(Lx(u)) x N MIMO channel for all users.
N is determined from size(H) where N = # tones
(equally spaced over \((0,1/T)\) at \( N/T \).
if time-domain h, \( H = 1/sqrt(N)*fft(h, N, 3) \)
Lxu 1xU vector of each user’s number of antennas
Rnn The Ly x Ly x N noise-autocorrelation tensor (last index is per tone)
\( cb \) \( cb = 1 \) for complex, \( cb=2 \) for real baseband

Outputs:
Rxx A block-diagonal psd matrix with the input autocorrelation for each user on each tone. Rxx has size \( \sum(Lx(u)) \times \sum(Lx(u)) \times N \).
sum trace(Rxx) over tones and spatial dimensions equal the Eu bsum the maximum rate sum.
bsum bsum_lin - the maximum sum rate with a linear receiver
\( b \) is an internal convergence sum rate value, not output

This program is modified version of one originally supplied by student
Chris Baca

EXAMPLE 5.4.5 (SWF Program Use) >> H = [80 30; 40 -50; 30 -15; 20 25]’;
H(:,:,2) = [20 0; -20 10; 35, 45; 10, 0]’;
>> Rnn=zeros(2,2,2);
>> Rnn(:,:,1)=eye(2);
>> Rnn(:,:,2)=eye(2);
>> user_ind = [1,3];
>> Eu = [1, 2];
>> cb=2;
>> [Rxx, bsum, bsum_lin] = SWF(Eu, H, [2 2], Rnn, cb)

Rxx(:,:,1) =
1.4141 0.1921 0 0
\[
\begin{bmatrix}
0.1921 & 1.0847 & 0 & 0 \\
0 & 0 & 0.6966 & -0.9584 \\
0 & 0 & -0.9584 & 1.3187 \\
\end{bmatrix}
\]

\[Rxx(:,:,2) =
\begin{bmatrix}
0.5097 & -0.7109 & 0 & 0 \\
-0.7109 & 0.9915 & 0 & 0 \\
0 & 0 & 5.9681 & 0.3152 \\
0 & 0 & 0.3152 & 0.0166 \\
\end{bmatrix}\]

\[bsum = 25.1071\]
\[bsum\_lin = 19.9260\]

\[\text{>> sum(diag(Rxx(1:2,1:2,1))+diag(Rxx(1:2,1:2,2))) = 4 (checks)}\]
\[\text{>> sum(diag(Rxx(3:4,3:4,1))+diag(Rxx(3:4,3:4,2))) = 8 (checks)}\]

**Energy-Sum MAC maximum rate sum:** Simultaneous water-filling corresponds to the largest data rate for any given \(\{Rxx\}\); the energy-sum MAC's largest rate sum will correspond to a particular set (or possibly sets) of \(Rxx\). A search through all possible energy sums produces this energy-sum maximum. The energy-sum MAC however is of most interest in duality with a corresponding BC channel (which always has the energy-sum constraint); Section 5.5 introduces duality, and eventually introduces a bcmax.m matlab program that implements Chapter 2's iterative double loop of worst-case-noise and water-filling to compute the saddle-point maximum BC rate sum. That software effectively short cuts the dual MAC's energy-sum maximum rate computation to a single calculation.

Recalling also Chapter 2's macmax.m program with \(N \geq 1\):

```
function [Rxx, bsum, bsum_lin] = macmax(Eu, h, Lxu, N, cb)
```

Simultaneous water-filling Esum MAC max rate sum (linear & nonlinear GDFE)
The input is space-time domain \(h\), and the user can specify a temporal block symbol size \(N\) (essentially an FFT size).

The above example has 3 units of total energy, and the macmax.m program needs the time-domain
input, so:

```matlab
>> h=ifft(H,2,3)
```

```matlab
h(:,:,1) =
  50.0000  10.0000  32.5000  15.0000
  15.0000 -20.0000  15.0000  12.5000

h(:,:,2) =
  30.0000  30.0000 -2.5000  5.0000
  15.0000 -30.0000 -30.0000  12.5000
```

```matlab
>> [Rxx, bsum , bsum_lin] = macmax(3, h, Lxu, N , cb)
```

```matlab
Rxx(:,:,1) =
  1.5004  0.0000  0  0
  0.0000  1.5003  0  0
  0  0  0.0000 -0.0000
  0  0 -0.0000  0.0000

Rxx(:,:,2) =
  0.5090 -0.7099  0  0
 -0.7099  0.9901  0  0
  0  0  1.4959  0.0790
  0  0  0.0790  0.0042
```

```matlab
bsum = 24.1062
bsum_lin = 23.9967
```

these data rates are indeed higher than the energy-vector MAC as necessarily must be the case (or equal).

5.4.2.3 Column-Wise Diagonal Dominance

A GDFE special case occurs when $R_{nn}(n) = I$ and $H_n \cdot A_n$ is column dominant. Column dominance means that each column’s diagonal element is much greater than the other elements in that column, so if $\tilde{H}_n = H_n \cdot A_n$, then

$$|\tilde{H}_n(u,u)| >> |\tilde{H}_n(i \neq u, u)| \forall i \neq u \quad .$$

With diagonal dominance, $R_f$ and thus $R_f^{-1}$ approximate diagonal matrices, and then $G \to I$. There is no feedback in this diagonally dominant situation, and order is irrelevant in the GDFE – all orders produce the same result. Indeed each user best uses an individual water-filling energy distribution with respect to only its noise, $R_{noise}(u,n) = R_{nn}(n) \forall u$. This situation occurs in most DSL channels, and also wireless in Subsection 5.2.4’s Massive MIMO channels, when the non-user upstream spatial noise is “white.” The GDFE feedforward matrix is solely sufficient to eliminate crosstalk on every tone.

The tonal GDFE bits/tone $n$ for user $u$’s $l^{th}$ input line/antenna is

$$b_{u,l,n} = \log_2 \left( C_{u,l,n} \cdot S_{0,u,l,n} \right) \quad .$$

Spatial Correlation of Noise

Spatial correlation refers to the possibly non-diagonal structure of the matrix $R_{NN}(n)$ on any (or many/all) of the independent tone-indexed matrix channels of the vector-DMT system. Since such noise leads effectively to a crosstalking noise-whitened channel, the GDFE also canonically handles such noise in the MAC. The consequent noise-whitened channel is not necessarily diagonally dominant, even if the original channel is diagonally dominant.

Earliest use of GDFE:

**EXAMPLE 5.4.6 (Vectored VDSL)**

VDSL is a zippered-DMT system as described in Section 4.6. The upstream direction is a vector-MAC if the DSLAM uses receiver coordination. The tone spacing is 4.3125 kHz with a cyclic extension of 640 samples on a sampling
clock of $32 \times 2.208$ MHz. Up to 8192 tones can be used in either direction. Two frequency plans have been used for a frequency-division separation of upstream and downstream bands. The so-called 998 plan of North America allows up and down transmission below 138 kHz (tone 32), and also up-only transmission between 4 MHz and 5.2 MHz and between 8.5 MHz and 17.6 MHz. Two uncorrelated and uncoordinated types of crosstalk noise noise A - (less severe) and noise F (more severe) are used for testing. These channels are all diagonally dominant.

Two options are investigated in Figure 5.29. In the lower curves, the upstream data rate uses water-filling in each of the FDM bands with no MAC processing nor feedback section (because of diagonal dominance). The upper curve is the data rate achieved with the GDFE vectoring and the same noises. At short line lengths, the self-crosstalk from same-direction VDSL signals dominates (while at longer distances it does not). No spatial correlation of the Noise A or Noise F was used in Figure 5.29. Figure 5.2.4’s systems have $U = 25$ users and $\rho_H = 25$. However, the crosstalk is largest between immediately neighboring connections, perhaps obviously so conceptually. The wires are also twisted-pairs with twisting at different number of twists per unit length on adjacent lines. Thus the crosstalk model is a random-selection process (across users, not time). For the $H$ matrix on the tones, this means that the not only are the crosstalk contributions random, but that the diagonal spatial-matched-matrix terms are much larger in a $25 \times 25$ channel matrix than are adjacent terms. This causes diagonal dominance effect, and so this is physically why vectored VDSL uses no feedback section. This observation later also found use in Massive MIMO wireless.

Figure 5.30 illustrates the situation if the other crosstalkers are fewer in number than the number of vectored users in the upstream direction, and thus their contributions appear less random and thus spatially correlated. In this case the diagonal may still dominate because $U = 25$ is greater than the number of correlated crosstalkers significantly. However the noise correlation allows the forward-matrix process to reduce further the crosstalk effect, again

---

60 Systems with up to 384 users are in field use circa 2020.

61 Not all wires can be physically next to one another in a cable full of such wires - so the nearest neighbor lines create most of crosstalk.
here without the feedback section. Figure 5.30’s data rate averages the channel samples over users (all have same length). Again, yet an even higher data rate is achieved at short lengths. In this case the other crosstalkers overlap the upstream VDSL band. Figure 5.30 uses no frequency plan and the users simply adapt to best band use with 14.5 dBm of upstream transmit power. Figure 5.30 yet further shows that data-rate increase is large with exploitation of noise spatial correlation (the rho value in plots). Section 5.5 revisits this example for bi-directional downstream vector BC. Figure 5.30’s rate goals were conceived prior to the corresponding standards group considering vectoring.

![Diagram](image_url)

Figure 5.30: VDSL with vectoring - other noises’ spatial correlation have fewer sources than the number of users.

### 5.4.3 MAC Energy-Sum Minimization

This section introduces a design method that minimizes the weighted energy sum\(^{62}\) to achieve a specific rate vector \(b\). This method finds the associated information distribution for a tonal GDFE design. This design is particularly useful in Section 5.5’s duality. The weighted energy-sum minimization is also an intermediate step that helps determine \(C_{MAC}(b)\) in Subsection 5.4.4. Rate vectors \(b \in C_{MAC}(b)\) can have multiple viable designs, even for the same energy weights; however \(b\) on the capacity region border\(^{63}\) \(b \in \mathcal{C}(b)\) may have only one design and correspond to a specific energy-weight vector. Rate vectors \(b\) in the interior \(\mathcal{S}(b)\) have a \(U\)-dimensional gap vector \(\Gamma\) that corresponds to the closest radially aligned border point \(b^* \in \mathcal{C}(b)\) where \(b^* = (1 + \gamma) \cdot b\), as in Chapter 2. The scalar \(\gamma \geq 0\) essentially measures the distance along a radially line including \(0\) through \(b\) to its closest border neighbor \(b^{opt} \in \mathcal{C}(b)\). This ratio then also applies to the bit-vector radial lengths also \(\|b^{opt}\|/\|b\| = 1 + \gamma\), and \(\Delta b = b^{opt} - b = \gamma \cdot b\) and \(\|\Delta b\| = \gamma \cdot \|b\|\). Then, the SNR based multi-user margin gap is (with an overline indicating dividing by the number of elements in \(b\), or \(U\))

\[
\Gamma_{mu} \triangleq \frac{2\|b^{opt}\| - 1}{2\|b\| - 1} \approx 2^{2\gamma}, \quad (5.302)
\]

\(^{62}\)or traces of input autocorrelation matrices in general

\(^{63}\)Recalling from Chapter 2 that the capacity-region border excludes the capacity region’s largest open subregion \(\mathcal{S}(b) \subset \mathcal{C}(b)\), so \(\mathcal{C}(b) = \mathcal{C}(b) \setminus \mathcal{S}(b)\).
and Chapter 2’s multiuser margin gap in dB is

\[ \Gamma \approx (6 \cdot \gamma) \text{ dB} \]  

(5.303)

The scalar margin gap \( \Gamma_{mu, b} \) is a single approximate indication of how close the overall multiuser design is to its best possible performance when each users’ rate is scaled proportionally to its size in \( b \). This is sometimes called proportional fairness. Any \( \Gamma > 0 \) dB for user-code application adds to \( \Gamma_{mu} \) to measure the loss with respect to best performance.

5.4.3.1 Minimizing a Weighted Energy Sum

The minimum MAC energy sum\(^{64}\) corresponds to the set of \( U \) input energies that minimize a weighted sum for a canonical (GDFE) design to attain a fixed rate vector \( b \):

\[
\min \left\{ \sum_{u=1}^{U} w_u \cdot \underset{E_u}{\text{trace}} \left\{ R_{XX}(u) \right\} \right\} \\
ST : \quad b \succeq [b_{1, min} \ b_{2, min} \ldots b_{U, min}]^\ast = b_{\text{min}}^\ast \succeq 0 \\
E \succeq 0
\]  

(5.304)

The vector \( w = [11\ldots1]^\ast \) corresponds to the energy sum; more generally \( w \succeq 0 \). Theoretically, (5.304) always has a solution for a given \( b \). However, (5.304) may not produce a \( b \in \mathcal{C}(b) \) because the designer may not know a priori the capacity region in selecting \( b \). When this happens, (5.304)’s minimized energy vector solution \( \mathcal{E} \) may exceed the energy limit \( \mathcal{E} \not\succeq \mathcal{E}_b \). (5.304) describes the capacity region point with lowest \( w^\ast \mathcal{E} \). When \( w = [1 \ 1 \ldots 1]^\ast \), this is the minimum-sum-energy point on the boundary \( \mathcal{C}_b(E) \).

Subsection 5.4.4 addresses the extension to determine if \( b \) is admissible, that is \( b \in \mathcal{C}(b) \) for the given \( E \). Again, Individual users’ energies are such that \( \sum_n \underset{E_u}{\text{trace}} \left\{ R_{XX}(u,n) \right\} \leq E_u \forall u \in U \) when for at least one user \( L_x > 1 \). When \( L_{x,u} = 1 \forall u \in U \), these terms reduce to the individual-user energies \( E_u \) that arise from summing energy over all individual user’s tones. Equation (5.304) specializes with vector DMT to

\[
\min \left\{ \sum_{u=1}^{U} \sum_{n=0}^{N} w_u \cdot \underset{E_u}{\text{trace}} \left\{ R_{XX}(u,n) \right\} \right\} \\
ST : \quad b = \sum_{n=0}^{N} [b_{1,n} \ b_{2,n} \ldots b_{U,n}]^\ast \succeq b_{\text{min}} \succeq 0 \\
b_{u,n} \succeq 0 \forall u, n
\]  

(5.305)

Each tone can use an independent GDFE with the set of minimum-sum energies \( E_{\text{min}} \) for \( b \) over all tones. Equation (5.304) follows from Equation (5.305) by setting \( N = 1 \) mathematically, but that ignores the vector-DMT/tonal-GDFE’s large simplification with temporal ISI. The tonal GDFE decomposition instead can simplify solution. Both (5.304) and (5.305) are concave and thus always have a solution with sufficient available energy for any rate vector \( b \succeq 0 \) and any weight vector \( w \succeq 0 \).

Figure 5.31 illustrates that the minimized weighted energy sum in (5.304) defines an energy-vector hyperplane (line in two dimensions) that rises with the \( E_{\text{min}} \) level, which must intersect first at a vertex\(^{65}\). Thus, a single vertex, or equivalently a single same order, is sufficient on all tones \( n \) at the optimum. Time/dimension-sharing is thus not needed when the only criterion is minimum weighted energy sum for a given fixed \( b \) and \( w \).

---

\(^{64}\)Assumes that all users employ capacity-achieving codes individually.

\(^{65}\)Or if a face is parallel to the weighted minimum-energy-sum plane, then at all points in the face, which certainly includes the vertices in that face for some particular orthotope among all those embedded within \( \mathcal{C}_b(E) \).
Equation (5.305)’s Lagrangian is

\[
L(R_{XX}(n), b, w, \theta) = \sum_{u=1}^{U} \left( w_u \cdot \left[ \sum_{n=0}^{N} \text{trace} \{ R_{XX}(u, n) \} \right] - \theta_u \cdot \left\{ \sum_{n=0}^{N-1} b_{u,n} - b_u \right\} \right)
\]  

(5.306)

(presuming no single-side-band uses of tone \( n = N \) tone if baseband real). Each side-constraint term tacitly implies that a \( \{b_{u,n}\} \) set for tone \( n \) must be within tonal-MAC matrix-AWGN (\( \tilde{H}_n \)) achievable region:

\[
b_n \in A_n(R_{XX}(u,n), b_n), R_{XX}(u,n), u = 1, ..., U.
\]  

(5.307)

As in Section 2.7, \( 2^U - 1 \) possible \( \mathbf{1}^* \cdot b_n \) rate sums bound \( A_n(R_{XX}(u,n), b_n) \) for any specific user autocorrelation matrix set \( \{R_{XX}(u,n)\}_{u=1,...,U} \) at each \( n = 0, ..., N - 1 \). The non-negative Lagrange side-constraint multipliers \( \theta_u \) are rate-control constants that contribute nothing to the Lagrangian’s value when the rate constraints are met, but can cause a suboptimal Lagrangian value when attempting to meet any tone’s corresponding side constraint (5.307). Equation (5.307)’s minimization for maximal \( \theta \geq 0 \) elements solves (5.305). Any solution has \( b_n \in A_n(b_n) \), because energy can increase in a way that minimizes the unbounded weighted energy sum and thus be larger than the desired \( \mathcal{E} \). This achievable-region constraint, however, simplifies a complete solution’s algorithmic development.

However, except when the \( \theta_u \equiv \theta \forall u = 1, ..., U \) (indicating a simple rate sum), the weighted-rate-sum’s maximizing order may differ from the order that maximizes the GDFE rate sum \( \mathbf{1}^* \cdot b \). The solution to the dual weighted rate-sum problem

\[
\max_{\{R_{XX}(u,n)\}} \sum_{u=1}^{U} \theta_u \cdot \left\{ \sum_{n=0}^{N-1} b_{u,n} \right\}
\]

(5.308)

\[
ST: \ \mathcal{E} \geq \sum_n \text{trace} \{ R_{XX}(u,n) \} \geq 0
\]

has essentially the same Lagrangian (differs only by constants that disappear under differentiation). Thus the same algorithm can address:

**original min weighted energy sum:** Thus, the original minimum-weighted-energy sum with specified energy weights \( w \geq 0 \) and \( \theta \geq 0 \) as side-constraint multipliers to minimize (maximize with negative sign) rate-constraint impact, or

**dual max weighted rate sum** the dual maximum-weighted-rate sum with \( \theta \geq 0 \) specified and \( w \geq 0 \) as the side-constraint multipliers to minimize (maximize with negative sign) energy-constraint impact.
Furthermore, when the minimum energy sum has all equal weights, (so with $w = 1$), the weighted rate-sum-maximization almost never has all equal weights, and vice-versa.

\begin{lemma}[MAC energy-rate duals]
These two optimizations are equivalent:

- Given: $b, w$ Find: $\mathcal{E}, \theta$ \hspace{1cm} (5.309)
- Given: $\mathcal{E}, \theta$ Find: $b, w$ \hspace{1cm} (5.310)

\textbf{Proof:} They both optimize the same Lagrangian. QED.

### 5.4.3.2 Tonal Decomposition of the Lagrangian

Interchange of Equation (5.430)’s finite sums over indices $u$ and $n$ obtains\(^{66}\)

\begin{equation}
L(R_{\times \times}, b, w, \theta) = \left( \sum_{u=1}^{U} \theta_u \cdot b_u \right) + \sum_{n=0}^{N-1} \left( \sum_{u=1}^{U} [w_u \cdot \text{trace} \{ R_{\times \times} (u, n) \} - \theta_u \cdot b_{u,n}] \right) + L_n(R_{\times \times}(n), b_n, w, \theta), \tag{5.311}
\end{equation}

where $L_n(R_{\times \times}(n), b_n, w, \theta)$ is a \textbf{tonal Lagrangian} term. For a given $\theta$, (5.311)’s first right-side sum term is not a function of $R_{\times \times}(u, n)$, because $b_u$ is given even though its exact constituency $b_{u,n}$ is not $(N > 1)$. $L_n(R_{\times \times}(n), b_n, w, \theta)$ for any given $\theta$ and $w$ depends only on tone $n$ quantities. Thus, the overall “max-min” is

\begin{equation}
L^* = \max_{\theta} \left\{ \sum_{n=0}^{N-1} L_{\min}(\theta, n) \right\} + \sum_{u=1}^{U} \theta_u \cdot b_u \quad \text{independent of } R_{\times \times}(u,n,n) \tag{5.312}
\end{equation}

where for each $n$ and $\theta$, noting $R_{\times \times}(n) = \text{blkdiag} \{ R_{\times \times}(U, n), ..., R_{\times \times}(1, n) \}$,

\begin{equation}
L_{\min}(\theta, n) \triangleq \min_{\{ R_{\times \times}(u,n) \}, b_{u,n}} L_n(R_{\times \times}(n), b_n, w, \theta). \tag{5.313}
\end{equation}

The last term in (5.312) has the given fixed $b_u$ (from the given $b$) values in the sum. (5.313)’s minimization is thus $\tilde{N}$ independent problems for a given $\theta$. The outer $\theta$ maximization proceeds after these $\tilde{N}$ inner minimizations.

There is also an achievable-region constraint that relates $b_n$ and $R_{\times \times}(n)$, or in this text’s context \textbf{GDFE constraint}, according to the achievable region:

\begin{equation}
b_n \in \left\{ b_n \mid 0 \leq \sum_{u \in U} b_{u,n} \leq \log_2 \left( \sum_{u=1}^{U} \tilde{H}_{u,n} \cdot R_{\times \times} (u, n) \cdot (\tilde{H}_{u,n})^* + I \right) \right\} = A_n \left( \{ R_{\times \times}(n) \}, \tilde{H}_n \right), \tag{5.314}
\end{equation}

where again

\begin{equation}
\tilde{H}_{u,n} = R_{NN}^{1/2}(n) \cdot H_{u,n}. \tag{5.315}
\end{equation}

\(^{66}\)The frequency-space $R_{\times \times}$ notation will use non-subscription arguments to include user and tonal dimensions, e.g. $R_{\times \times}(u,n)$ is an $L_{x,u} \times L_{x,u}$ smallest matrix, while $R_{\times \times}(u)$ includes all tones as a 3-dimensional tensor and similarly $R_{\times \times}(u, n)$ includes all users as a 3-dimensional tensor, while $R_{\times \times}$ includes all tones and all users for a 4-dimensional tensor. $R_{\times \times}$ is often a 4-dimensional array/tensor in vector MAC software programs.
Summing the solutions of (5.313) over the tones provides

$$L_{\min}(\theta) = \sum_{n=0}^{N-1} L_{\min}(\theta, n) \quad (5.316)$$

This $L_{\min}(\theta)$ has maximum at $\theta^o$, as in (5.312), for

$$L(\theta^o) = \sum_{u=1}^{U} \theta^o_u \cdot b_u + L_{\min}(\theta^o) \quad \text{point in} \ A(R_{XX}, b) \quad (5.317)$$

**Theorem 5.4.1 Maximum Weighted Rate Vertex Optimum:**

*For any given specific $\{R_{XX}(u, n)\}$, the maximum weighted rate sum, or hyperplane,

$$b_{max} = \sum_{u=1}^{U} \theta_u b_u \quad (5.318)$$

must occur at an achievable-region MAC vertex for tone $n$. \textbf{Proof:} There are $U$ GDFE-point vertices for a given block-diagonal $R_{XX}$ that define the maximum rate-sum hyperplane of the orthotope $A(R_{XX}, b)$. Any hyperplane (this is easy to visualize in 2 or 3 dimensions, see for instance Figure 5.32), $\theta^* b = b_{\theta-\text{sum}}$ is constant with reducing $b_{\theta-\text{sum}}$, first touching at an orthotope $A(R_{XX}, b)$ vertex. If the max-rate-sum hyperplane is $\theta = 1 \cdot c$, where $c$ is some constant, then many points including all vertices on the face share the same maximum weighted rate sum. Thus, the $\theta$, or equivalently intersection-vertex’ GDFE order, is the best order. All tonal GDFE’s use that order $\pi$ for this $R_{XX}$ - including the overall $R_{XX}$ that is optimal because $\theta$ does not depend on $n$. That best order corresponds to

$$\theta_{\pi^{-1}(u)} \geq \theta_{\pi^{-1}(u-1)} \geq \ldots \geq \theta_{\pi^{-1}(1)} \quad , \quad (5.319)$$

which follows this inductive argument that Figure 5.32 illustrates: If

$$\theta_2 \cdot b_2 + \theta_1 \cdot b_1 \geq \theta_2 \cdot b_1 + \theta_1 \cdot b_2 \quad (5.320)$$

(which follows from the vertex’ maximum), then the difference between the two sums is

$$b_2 \cdot (\theta_2 - \theta_1) - b_1 \cdot (\theta_2 - \theta_1) \geq 0 \quad (5.321)$$

which is only greater than zero if $b_2 > b_1$. By letting 1 be the next lower user in the order, and 2 represent the previous sum with a theta representing the average theta, an induction argument holds for the next smallest $\theta$ (which is necessarily less than the average of higher indices) remaining. \textbf{QED.}

This theorem and proof are analogous to Figure 5.31’s earlier analysis.

Figure 5.32: Illustration of plane (line) touching first at vertex. (Any line lowering must first touch at vertex.)
Computation of tonal mutual information: The relationship between the bits per user/tone $b_{u,n}$ and tone autocorrelation $R_{XX}(u, n)$ follows from the known best order (that is at given $\theta$), and the tonal-GDFE/achievable-region relation, which is the difference between two chain-rule mutual information terms, becomes (with zero gap)

$$b_{u,n} = \log_2 \left| \sum_{i=1}^{U} \bar{H}_{\pi^{-1}(i),n} \cdot R_{XX}(\pi^{-1}(i), n) \cdot \bar{H}_{\pi^{-1}(i),n}^* + I \right| - \log_2 \left| \sum_{i=1}^{U} \bar{H}_{\pi^{-1}(i),n} \cdot R_{XX}(\pi^{-1}(i), n) \cdot H_{\pi^{-1}(i),n}^* + I \right|. \quad (5.322)$$

When computing the sum $\sum_{u=1}^{U} \theta_u \cdot b_{u,n}$, each "log" term in (5.440) appears twice with different value of $\theta_i$, so with simplification (and $\theta_{\pi^{-1}(0)} = 0$) the sum simplifies to

$$\sum_{u=1}^{U} \theta_u \cdot b_{u,n} = \sum_{u=1}^{U} \left[ \theta_{\pi^{-1}(u)} - \theta_{\pi^{-1}(u-1)} \right] \cdot \log_2 \left| \sum_{i=1}^{U} \bar{H}_{\pi^{-1}(i),n} \cdot R_{XX}(\pi^{-1}(i), n) \cdot \bar{H}_{\pi^{-1}(i),n}^* + I \right|. \quad (5.323)$$

Equation (5.323) relates the weighted rate sum to the tonal autocorrelation matrices the order; $\pi^{-1}(u) \geq \pi^{-1}(u-1)$ ensures that (5.323) remains concave in $R_{XX}(u, n)$. Degraded MACs have linearly dependent $H_u$, or equivalently two users combine effectively into a macro-user. Such degraded channels have two successive equal $\theta$ values in (5.323). This situation corresponds to a need to time-share if the given desired $b$ is not coincidentally a vertex already; however if only the maximum weighted rate sum is of interest, any point on the face produces that maximum. This concave function can then be maximized to find the best set of autocorrelation matrices for this tone and the given value of $\theta$, as in the next subsection.

5.4.3.3 Finding $R_{XX}$ and the minPMAC program

Worst-case-noise master, Dr. Mehdi Mohseni, kindly also provides an algorithm and software to compute the solution to the overall sum-energy minimization problem. The algorithm and software were later improved and expanded by Dr. Yun Liao. Mohseni’s Algorithm iterates between a best tonal-autocorrelation matrix $R_{XX}(u, n)$ to step to minimize (5.313) that also sums terms for a fixed $\theta$ value and a subsequent best $\theta$ step for a fixed $\{R_{XX}(u, n)\}$. With appropriate initialization, this iterative alternation between $R_{XX}(u, n)$ and $\theta$ optimization rapidly converges to the overall optimum solution through the assistance of appropriate convex optimization procedures within each step. More accurately, after initialization of $\theta = 1$, there are two iterated steps:

[RXX step] For given $\theta$ and each tone $n = 1, ..., N$, this $R_{XX}$ step computes $L_{\min}(\theta, n)$ in (5.313) for each tone via a convex minimization procedure, most likely a steepest descent procedure as in the upcoming software and description. This step further sums the $L_{\min}(\theta, n)$ over $n$ and adds $\theta^T b_u$ as in (5.311) and (5.312) to form $L^*$ for the current optimized set of all users’ tonal-autocorrelation matrices.

[Order Step ($\theta$)] This step determines for the $R_{XX}$-step’s now-fixed output that maximizing value of $\theta$ for $L(R_{XX}, b, w, \theta)$ and the given $[b, w]$. The algorithm then iterates through this new $\theta$ value by return to the $R_{XX}$ step. This order step typically uses an elliptical or “sub-gradient” optimization algorithm as in the upcoming software and description.

The iteration converges if each step uses a convergent algorithm for the intermediate convex optimization. The algorithm terminates when successively computed $L(R_{XX}, b, w, \theta)$ on successive order-step completions are sufficiently close to each other, or when $b$ attains the specified value.

$R_{XX}$-step - gradient-descent algorithm: Appendix C covers matrix-calculus basics that are helpful for those desiring more complete derivational details. The $R_{XX}$ step can use a gradient-descent method.
independently for each tone. Basically, Mohseni’s Algorithm minimizes the Lagrangian \( L_{\min}(\theta, n) \) incrementally in the negative-gradient direction according to

\[
R_{XX}(\ell, n, k + 1) = R_{XX}(\ell, n, k) - \mu \cdot (\nabla^2 L_k)_{n,k}^{-1} \cdot \nabla L_{n,k} \quad (5.324)
\]

where the gradient matrix simplifies\(^{67}\) to a block-cell vector:

\[
\nabla L_{n,k} \triangleq \begin{bmatrix}
\nabla_{R_{XX}:U}(n, k) \\
\vdots \\
\nabla_{R_{XX}:1}(n, k)
\end{bmatrix} \quad (5.325)
\]

This simplification exploits the MAC’s block-diagonal \( R_{XX}(n) \) requirement to compress the number of quantities into a cell-array of \( L_{x,u} \times L_{x,u} \) components, reducing to the energy vector \( \mathcal{E} \) when \( L_{x,u} = 1 \), and a block-column vector with constant \( L_{x,u} = L_x > 1 \). The corresponding Hessian has cell-matrix components (necessarily square when \( \ell = j \)) that expand each cell of \( \nabla_{R_{XX}(u,n)} \) into a larger row set with \( \ell = 1, ..., U \) and \( j = 1, ..., U \) such that

\[
\nabla^2_{R_{XX}} L(\ell, j, n, k) = \begin{bmatrix}
\nabla_{R_{XX}:U,U}(\ell, j, n, k) & \cdots & \nabla_{R_{XX}:U,1}(\ell, j, n, k) \\
\vdots & \ddots & \vdots \\
\nabla_{R_{XX}:1,U}(\ell, j, n, k) & \cdots & \nabla_{R_{XX}:1,1}(\ell, j, n, k)
\end{bmatrix}, \quad (5.326)
\]

and which can be organized into full Hessian as a matrix of component matrices. The product \( (\nabla^2 L_k)_{\ell,j}^{-1} \cdot \nabla L_{n,k} \)

\( \nabla L_k \) is of the same size as \( R_{XX} \) and updates it in a steepest-descent direction a column-array that stacks vertically the \( U \) matrices \( R_{XX}(u, n) \) that are the nonzero components along the block-diagonal matrix \( R_{XX}(n) \).

Using (5.440), the gradient has \( U \) components (with \( R_{YY}^{-1}(0) = I \) and letting \( R_{YY}^{-1}(j) \) be the matrix inside the determinants), and presuming the order notation simplifies\(^{68}\) to just \( j \) instead of the \( \pi^{-1} \) notation, while using the chain rule of derivatives (index \( n \) suppressed on \( R_{YY}(u) \) \( \triangleq \sum_{j=u}^U \tilde{H}_u \cdot R_{XX}(u, n) \cdot \tilde{H}_u^* + I \)).

\[
\nabla L_{R_{XX}}(u) = \frac{1}{\ln(2)} \cdot \sum_{j=u}^U \theta_j \cdot |\tilde{H}_u \cdot R_{YY}^{-1}(j) \cdot \tilde{H}_u| - w_u \cdot I + \sum_{i=1}^{L_{x,u}} \lambda_{R_{XX}}^{-1}(i) \text{ where for } u = 1...U \quad (5.327)
\]

\[
R_{YY}^{-1}(u) = R_{YY}^{-1}(u-1) - R_{YY}^{-1}(u-1) \cdot \tilde{H}_u \cdot \left[ R_{XX}^{-1}(u) + \tilde{H}_u^* \cdot R_{YY}^{-1}(u-1) \cdot \tilde{H}_u \right]^{-1} \cdot \tilde{H}_u^* \cdot R_{YY}^{-1}(u-1). \]

The \( \sum_{i=1}^{L_{x,u}} \lambda_{R_{XX}}^{-1}(i) \) term corresponds to a simplified non-negative user-energy side constraint in the form of individual \( \sum_{\ell=1}^{L_{x,u}} \ln\left(\lambda_{R_{XX}}^{-1}(\ell)\right) \) singular values’ logarithms penalizing infinitely any user’s negative energy. The Hessian has \( U^2 \) components (\( \ell = 1, ..., U, j = 1, ..., U \))

\[
\nabla^2 L_{\ell,j} = \frac{1}{\ln(2)} \cdot \sum_{i=\max(\ell,u)}^U \theta_i \cdot \text{trace}\left\{ \tilde{H}_u \cdot \tilde{H}_u^* \cdot R_{YY}^{-1}(i) \cdot \tilde{H}_u \cdot \tilde{H}_u^* \right\} - \left\{ \lambda_{R_{XX}}^2(u) \right\} \quad (5.328)
\]

This Hessian’s inverse will exist when the channel is non-degraded; otherwise a pseudoinverse is necessary and there is ambiguity that does not appear in the weighted energy sum, but would occur with an energy-vector constraint.

(5.324)’s positive step size \( \mu \) must be less than half the inverse of the Hessian’s trace to converge. The Hessian calculation is complex, but contained within Subsection 5.4.3.3’s software. The proper Hessian calculation also includes constraints that ensure the autocorrelation matrices are positive semi-definite. The ratio \( (\nabla^2 L_k)^{-1} \cdot \nabla L_k \) also must be real because \( L_k \) is real\(^{69}\). Complex gradient and Hessian matrices need to follow Appendix C’s more involved relationships.

\(^{67}\)Cells allow elements of what nominally appears as a vector or matrix with constant element sizes to generalize to variable sizes (like \( L_{x,n} \)).

\(^{68}\)This reorders users indices by the current \( \theta \), which can occur after each order step. The relationship to the original under indices must be stored for eventual final \( \pi(u) \).

\(^{69}\)Thus, good finite-precision software implementation should ensure this.
Order step and initialization: The “order step” effectively determines the best order and converges rapidly through use of an elliptical algorithm:

The $R_{XX}$ step produces $R_{XX}(u, n, k)$ on iteration $k$ and corresponding related set of $b_{u,n}(k)$. These input as constants to the Lagrangian expression to create a function of $\theta$, $L^o(\theta)$, that is

$$L^o(\theta) = \sum_{n=1}^{\mathbb{N}} \sum_{u=1}^{U} L_n(R_{XX}(u, n, k), b_{u,n}(k), w, \theta) ,$$

(5.329)

The order step evaluates (5.329) at an offset from the previous fixed $\theta$ (used in the $R_{XX}$ step), which is $\theta \rightarrow \theta + \Delta \theta$. That offset has the function $L^o(\theta + \Delta \theta)$ is then

$$L_{\text{min}}(\theta + \Delta \theta) \leq L^o(\theta + \Delta \theta) = L_{\text{min}}(\theta) + \sum_{u=1}^{U} \Delta \theta_u \cdot \left[ b_u - \sum_{n=1}^{\mathbb{N}} b_{u,n} \right] = L_{\text{min}}(\theta) + \Delta \theta \cdot \Delta b ,$$

(5.330)

where $L_{\text{min}}(\theta)$ is the value from (5.313). If $\Delta b = 0$, then the best $\theta$ is already known and equal to the value used\(^\text{70}\) in the immediately preceeding $R_{XX}$ step, and the algorithm completes. The order step determines a $\Delta \theta$ that maximizes $L^o(\theta + \Delta \theta)$, and then the algorithm returns with that new $\theta$ to the $R_{XX}$ step. Essentially, the components of $\Delta b$, sign and magnitude, indicate desirable $\theta$ values that a subsequent $R_{XX}$ step can use to adjust $\Delta b$, with any consequent change in order, to zero. $\Delta b$ is known as a “sub-gradient” in optimization theory for the function $L_{\text{min}}(\bullet)$.

Ellipsoid Algorithm: The order step’s ellipsoid method iteratively uses a descent procedure to converge to the optimum $\theta^o$ value. The ellipsoid method must initialize with an ellipsoid $O(\theta)$ that contains $\theta^o$. The algorithm iteration index is $k$ with estimate of $\theta^o$, $\theta_k$ is the center of an ellipsoid, $O_k(\theta)$. The algorithm recursively finds a smaller-volume ellipsoid, $O_{k+1}(\theta)$, at iteration $k + 1$ as the smallest containing the intersection of $O_k$ and the half-space $(\theta - \theta_k)^* \cdot \Delta b \geq 0$, because $\theta - \theta_k$’s projection on $\Delta b > 0$ means the surviving planar half space contains $\theta$.

Each ellipsoid has mathematical description:

$$O_k(\theta) \overset{\Delta}{=} \{ \theta \mid (\theta - \theta_k)^* \cdot A_k^{-1} \cdot (\theta - \theta_k) \leq 1 \} ,$$

(5.331)

where positive-definite $A_k$ (this $A$ has no direct relation to Chapter 2’s special square root discrete modulator $A$ and is known to be the Hessian matrix for the ellipsoid) specifying the ellipsoid axes. $A$ has eigenvectors $v_{A,u}$. The matrix $A$’s eigenvalues $\lambda_{A,u}$ specify the semi-axis lengths as $\sqrt{\lambda_{A,u}}$. Defining $\Delta \theta_k \overset{\Delta}{=} (\theta - \theta_k)$, the new ellipsoid is then

$$O_{k+1}(\theta) = O_{\text{min}} \left\{ O_k(\theta) \cap \Delta \theta_k^* \cdot \Delta b \geq 0 \right\} .$$

(5.332)

When $U' > U$, equivalently at least one $L_{u,u} > 1$, the matrix $\tilde{H}_u = F_u \cdot \Lambda_u M_u^*$ redefines as $\tilde{H}_u \rightarrow \tilde{H}_u \cdot M_u$ where user $u$ has autocorrelation matrix $M_u \cdot \text{Diag}(E_u) \cdot M_u^*$. While this is an additional MAC-input restriction, it does not reduce the MAC capacity region, nor preclude any $b \in C(b)$ from realization with such restriction.

The ellipsoid-normalized rate-difference sub-gradient vector is

$$\Delta \tilde{b}_k \overset{\Delta}{=} \frac{\Delta b_k}{\sqrt{\Delta b_k^* \cdot A_k^{-1} \cdot \Delta b_k}} ,$$

(5.333)

which simplifies (5.332) and (5.331) to produce a new ellipsoid that covers the intersection of the half plane and the old ellipsoid:

$$\theta_{k+1} = \theta_k - \frac{1}{U + 1} \cdot A_k \cdot \Delta \tilde{b}_k \text{ new center}$$

(5.334)

$$A_{k+1} = \frac{U^2}{U^2 - 1} \left[ A_k - \frac{2}{U + 1} \cdot A_k \cdot \Delta \tilde{b}_k^* \cdot A_k \right] \text{ new axes} .$$

(5.335)

\(^{70}\) If $\Delta \theta^o \cdot \Delta b < 0$, then $L_{\text{min}}(\theta + \Delta \theta) \leq L_{\text{min}}(\theta)$ and $\theta + \Delta \theta$ cannot be the optimal solution. Recall the idea is to find a $\Delta \theta$ that does minimize $L_{\text{min}}$.
These steps execute once during each order step after the $R_{XX}$ step updates the current autocorrelation matrix. The first step in (5.334) is a steepest descent (with $A$ as the inverse of the Hessian matrix $A^{-1}$) weighting the direction by the ellipsoid axes' lengths together with step-size $1/(U + 1)$ that moves so that $\Delta \theta_{k+1} \cdot \Delta b_k \geq 0$. The second update in (5.335) follows through insertion of (5.334) into (5.331) and solving for $A_{k+1}$ in terms of $A_k$, $U$, and $\Delta b_k$. The new ellipsoid’s volume decreases as
\[
\text{vol}(O_{k+1}) \leq e^{-\frac{1}{2U}} \cdot \text{vol}(O_k),
\]
implying exponential convergence in terms of the number of iterations through the two successive steps of Mohseni’s algorithm.

**Initialize Ellipsoid:** To determine the initial ellipsoid that contains $\theta^o$ on any order step, this step’s initialization selects a random order and a user-bit/symbol distribution $b^o$ such that
\[
\begin{align*}
b_i \neq u &= b^o_i \\
b_u &= b^o_u + 1.
\end{align*}
\]
Initialization makes a single-pass through Chapter 4’s fixed-margin iterative water-filling, for the selected order and bit distribution in (5.337) and (5.338), generates the bit distribution $b^o$, thus generating a set of $\{R_{XX}(u,n)\}$. This set of autocorrelation functions substitutes into the Lagrangian equation, which must always be non-negative for $\theta^o \succeq 0$, so
\[
0 \leq L_{\min}(\theta^o) \leq \left[ \sum_{u=1}^U \sum_n w_u \cdot \text{trace}(R_{XX}(u,n)) \right] + \left[ \sum_{u=1}^U \left( b_u - \sum_n b_{u,n} \right) \right] \cdot \theta^o_u
\]
must hold. Rearranging this equation using (5.338) leads to
\[
0 \leq \theta^o_u \leq \sum_{u=1}^U \sum_n w_u \cdot \text{trace}(R_{XX}(u,n)) = \theta_{u,max}. \tag{5.340}
\]
The step in (5.338) repeats for each user (that is incrementing each user successively by one bit as in (5.337) and (5.338)) to generate a $\theta_{u,max}$ for each, as in (5.340). Thus by executing $U$ single-pass FM IW’s for each of the $U$ bit distributions (each incrementing one user by one bit while others are held constant), a box containing $\theta^o$ is obtained. The eigenvector-axised ellipsoid that covers this box has diagonal Hessian with squared lengths at least $\theta^2_{u,max}$ on each of these axes.
\[
A_0^{-1} = \begin{bmatrix}
\left( \frac{1}{\theta_{1,max}} \right)^2 & 0 & \ldots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & \ldots & 0 & \left( \frac{1}{\theta_{U,max}} \right)^2
\end{bmatrix}.
\]
This step’s algorithm runs until $\theta_{k+1} - \theta_k < \epsilon$.

**Matlab Implementation of Mohseni’s Algorithm:** The “minPMAC” program is for $L_x = 1$ (so all subusers). However, multiple antennas per user transmitter readily can be “spoofed” by use of SVD on spatial-multi-dimensions/per user and then adding them as additional “tones”.

The program call is:
\[
(E, \text{theta}, b) = \text{minPMAC}(G, \text{bu}, w)
\]
MinPMAC’S inputs are

---

71 The water-filling need be executed only once for each user because a single pass using $R_{noise}(u)$ for some order will produce a solution, which is all that is necessary for initialization.

72 Because an SVD certainly could be done for each user $\tilde{H}_u$, effectively making it look like more tones - although this trick works only with energy-sums and not fixed spatial-energy vectors.
• w is the $U \times 1$ energy-sum weight vector (typically $w=\text{ones}(U,1)$).
• $bu$ is the $U \times 1$ non-negative user bits-per-symbol vector.
• $H$ is an $L_y \times U \times N$ tensor of $L_y \times U$ matrices $[\tilde{H}_1 \ \tilde{H}_2 \ \ldots \ \tilde{H}_N]$ with $H_n = R_{NN}^{-1/2}(n) \cdot H_n$ from a vector DMT system. (This is the same as $\tilde{H}$, specified as an array/tensor for $n$.)

The outputs from the function are

• $E$ is a $U \times N$ matrix of energies $E_{u,n}$.
• $\theta$ is the $U \times 1$ rate-weight vector $\theta$ that determines the best order.
• $b$ is a $U \times N$ matrix of $b_{u,n}$.

These programs are available to all at the Cioffi-course web site(s). The main program is again minPMAC and appears here for the curious reader. The functions listed comments describe all inputs and outputs. The program minPmac basically runs ellipsoid algorithm for the order step and computes Hessian and the weighted gradient for the $R_{xx}$ step.

```matlab
>> help minPMAC
function [E, theta, b] = minPMAC(H, bu, w)
This is the main function, and includes the ellipsoid-descent step

Inputs:
1) $H$, an $L_y \times U \times N$ channel matrix. $L_y$ is the number of receiver antennas,
   $U$ is the total number of users and $N$ is the total number of tones.
   $H(:,:,n)$ is the channel matrix for all users on tone $n$
   This code assumes each user has only a single transmit antenna, thus $H(:,u,n)$ is a column vector.
   If $H$ is complex, then outputs are for two-dimensional tones
   If $H$ is real-baseband, then the upper half tones can be ignored or all tones interpreted as real one-dimensional outputs
2) $w$, a $U \times 1$ vector containing the weights for each user’s energy.
3) $bu$, a $U \times 1$ vector containing the target rates for all the users.

Outputs:
1) $E$, a $U \times N$ matrix containing the tonal energies for all users
   that minimizes the weighted-sum power. $E(u,:)$ is the power allocation for user $u$ over all tones.
2) $\theta$, the optimal $U \times 1$ dual variable vector containing optimal weights
   of rates. $\theta$ determines the decoding order.
3) $b$, a $U \times N$ matrix containing the rates of all users on all tones

Special subroutines called
StartEllipse, Lag_dual_f

The subroutine minPtone is the core of the code (that is, the Interior Point method). It optimizes the weighted rate sum minus the weighted energy over the capacity region. The minPmac program solves for one tone and the results are combined in Lag_dual_f to compute the dual function over all tones. The subroutine minPtone is the $R_{xx}$ step. It optimizes the weighted rate sum minus the weighted energy over the capacity region for any tone. The minPMAC program accumulates the results over tones through other subroutines Lag_dual_f and eval_f that also follow:

```matlab
>> help minPtone
function [f, b, e] = minPtone(H, theta, w, N)
```
\text{minPtone} maximizes \( f = \sum_{u=1}^{U} \lambda_u \cdot b_u - \sum_{u=1}^{U} w_u \cdot e_u \)
subject to \( b \in C_g(G,e) \); presumably for a single tone.

Inputs:
1) \( H \), an \( Ly \) by \( U \) channel matrix. \( Ly \) is the number of receiver antennas, \( U \) is the total number of users.
   \( H(:,;u) \) is the channel for user \( u \). \text{minPtone} assumes each user has only a single transmit antenna, thus \( H(:,;u) \) is a column vector.
2) \( \theta \), a \( U \) by 1 vector containing the weights for the rates.
3) \( w \), a \( U \) by 1 vector containing the weights for each user’s power.
4) \( N \) is needed to pass to subroutines so that PSD is increased at fixed sampling rate relative to noise-whitening.

Outputs:
1) \( f \), the minimum value (or maximum value of the \(-1 \cdot \) function).
2) \( b \), a \( U \) by 1 vector containing the rates for all users that optimizes the given function.
3) \( e \), a \( U \) by 1 vector containing the PSD for all users that optimizes the given function.

Subroutines called are \text{eval}_f\ , \text{Hessian}

\text{Lag}\_\text{dual}_f\sums\ the\ individual\ tones\ to\ compute\ the\ overall\ Lagrangian:

\text{eval}_f\ calls\ subroutine\ \text{Hessian} to compute the value of the cost function, using the gradient and the Hessian thereof. I have added the expressions for the gradient and the Hessian inside the Hessian file.
vector for user u. Again each user has just one transmit antenna.
3) e, a U by 1 vector containing each user’s power.
4) w, a U by 1 vector containing weights for each user’s power

the output is f the function value given above.
no subroutines are called and this function does not need to know N 4) w, a U by 1 vector containing

>> help Hessian
function [g, H] = Hessian(theta, G, e, w)

This function calculates the gradient (g) and the Hessian (H) of the function f(e) = (theta_1 - theta_2)/2 * log det(I + G_1 * G_1^H * e_1) + (theta_2 - theta_3)/2 * log det(I + G_1 * G_1^H * e_1 + G_2 * G_2^H * e_2) + ... (theta_{U-1} - theta_U)/2 * log det(I + G_1 * G_1^H * e_1 + ... + G_{U-1} * G_{U-1}^H * e_{U-1}) + theta_U/2 * log det(I + G_1 * G_1^H * e_1 + ... + G_U * G_U^H * e_U) - w^T * e + sum_{u=1}^U log(e_u)
theta should be in decreasing order, theta_1 >= theta_2 >= ... >=theta_U.

the inputs are:
1) theta, a U by 1 vector of weights for the rates.
2) G, an Ly by U channel matrix. G(:,u) is the channel vector for user u. Again each user has just one transmit antenna.
3) e, a U by 1 vector containing each user’s power.
4) w, a U by 1 vector containing weights for each user’s power

the outputs are:
1) g, U by 1 gradiant vector.
2) H, U by U Hessian matrix.

The program minPMAC also calls Ellipsoid initialization subroutine StartEllipse:

>> help startEllipse
function [A, g] = startEllipse(H, bu, w)
This function initializes the ellipsoid method for power minimization.

function [A, g] = startEllipse(H, bu)
G containes the channel matrices as is defined in minPMAC.m and bu is the target U by 1 rate vector, w is the U by 1 power weights.
For all fixed margin WF steps, decoding order is 1,2,...U.

A is the matrix describing the staring ellipsoid and g is its center subroutine called fmwaterfill_gn

Finally, startEllipse uses an FM waterfill subroutine:

>> help fmwaterfill_gn
function [bn, en] = fmwaterfill_gn(gn, b_bar, gap)

INPUT
gn is the channel gain (a row vector).
b_bar is the target bit rate (b_total / Ntot)
gap is the gap in dB

OUTPUT
en is the energy in the nth subchannel
bn is the bit in the nth subchannel
dB into normal scale

There is a second program that is more general and allows variable numbers of antennas, but uses CVX software.

function [Rxxs, E, theta, bun] = minPMAC_cvx(H, Lxu, bu_min, w)
minPMAC_new solves minimize sum_u(w_u*tr(Rxx(u))), subject to bu>=bu_min
The inputs are:
1) H, an Ly by Lx by N channel matrix. Ly is the number of receiver antennas, Lx is the total number of transmit antennas and N is the total number of tones.
   H(:,:,n) is the channel matrix for all users on tone n.
   If H is complex, then outputs are for two-dimensional tones
   If H is real-baseband, then the upper half tones can be ignored
   or all tones interpreted as real one-dimensional outputs
2) Lxu, a scalar or a length-U vector containing the number of transmit antennas of each user. If Lxu is a scalar, each user has Lxu antennas.
3) bu_min, a U by 1 vector containing the target rates for each user.
4) w, a U by 1 vector containing the weights for each user’s power.

The outputs are:
1) Rxxs, cell array containing the Rxx’s for each user on each tone.
   Rxx{n}{u} is the Rxx for user u on tone n.
2) E, a U by N matrix containing the transmit power of each user on each tone.
3) theta, the optimal U by 1 dual variable vector containing optimal weights of rates. theta also determines the decoding order.
4) bun, a U by N matrix containing the rates of each user on each tone

Example uses of minPMAC software:

**EXAMPLE 5.4.7 [simple scalar MAC]** This first example returns to Example 2.7.1’s simple scalar MAC with gains $h_2 = 0.8$ and $h_1 = 0.6$. For this channel the number of tones can be set as $N = 1$, while $U = 2$ and $L_y = 1$. The input energies were $E_1 = E_2 = 1$, but not of use in the minPMAC software. This example attempts $b_1 = b_2 = 3$ bits/dimension for each user. The noise variance is 0.0001, so the effective noise-whitened channel gains are 80 and 60 respectively. The matlab steps follow:

```matlab
>> H=zeros(1,2,1) % dimensioning a tensor
H =
   0   0
>> H(1,1,1)=80;
>> H(1,2,1)=60
H= 80 60
>> b = [3 3];
>> w = [ 1 1];
>> [E, theta, B]=minPMAC(H, b, w)
count = 96
E =
 0.6300
 0.0175
theta =
```
\[ B = 3.0000 \]

\[ \log_2(1+(6400*E(1))/(1+3600*E(2))) = 2.9858 \]

\[ \log_2(1+(3600*E(2))) = 3.0043 \]

This use of minPMAC achieves desired data rates, 3 bits/subsymbol for each of the users, with energies that fortunately satisfy constraints. Recall the maximum rate sum was \( 0.5 \log_2 |H \cdot 1 \cdot H^*| = 6.64 \) bits/subsymbol, so \( b = b_2 + b_1 = 6 < 6.64 \).

This same example with \( N = 4 \) obtains (note that the rate vector is multiplied by 4 since there are now 12 bits over a subsymbol that is 4 times longer for each user. Further, the discrete Fourier Transform multiplies by \( 1/\sqrt{N} = 1/2 \) here:

\[
\begin{align*}
\text{(Example 5.4.7 continued)} & \quad H = \text{zeros}(1,2,4); \\
& \quad H(1,1,:) = 40; \\
& \quad H(1,2,:) = 30; \\
& \quad [E, theta, B] = \text{minPMAC}(H, 4*b, w)
\end{align*}
\]

\[ E =
\begin{array}{cccc}
0.6300 & 0.6300 & 0.6300 & 0.6300 \\
0.0175 & 0.0175 & 0.0175 & 0.0175 \\
\end{array}
\]

\[ \theta =
\begin{array}{c}
1.2809 \\
1.2972 \\
\end{array}
\]

\[ B =
\begin{array}{cccc}
3.0000 & 3.0000 & 3.0000 & 3.0000 \\
3.0000 & 3.0000 & 3.0000 & 3.0000 \\
\end{array}
\]

Increasing \( b \) at the same bandwith eventually causes the energy limit to be exceeded because minPMAC only addresses minimum energy sum for the given data rate.

\[
\begin{align*}
\text{count} = 98 \\
\quad [E, \theta, B] = \text{minPMAC}(H, 5*b, w)
\end{align*}
\]

\[ E =
\begin{array}{cccc}
5.0917 & 5.0917 & 5.0917 & 5.0917 \\
0.0500 & 0.0500 & 0.0500 & 0.0500 \\
\end{array}
\]

\[ \theta =
\begin{array}{c}
10.2400 \\
10.2840 \\
\end{array}
\]

\[ B =
\begin{array}{cccc}
3.7500 & 3.7500 & 3.7500 & 3.7500 \\
3.7500 & 3.7500 & 3.7500 & 3.7500 \\
\end{array}
\]

In this case, the attempted data rate is too high, so the minPMAC output energies violate the energy constraint of 1 unit/per user. This second example illustrates that minPMAC does not accept energy constraints and only minimizes a weighted energy sum. Thus, it will always produce an “answer,” but thus the design may not be implementable. The next Subsection’s admMAC program addresses solution admissibility when there is an energy constraint.

As a final example, the channel changes to have different gains on all tones
\[
H(:,:,1) = \\
\begin{bmatrix}
80 & 60 \\
40 & 30 \\
50 & 50 \\
30 & 40 \\
\end{bmatrix}
\]

\[
H(:,:,2) = \\
\begin{bmatrix}
9 & 7 \\
2 & 1 \\
3 & 5 \\
4 & 6 \\
\end{bmatrix}
\]

\[
H(:,:,3) = \\
\begin{bmatrix}
1 & 2 \\
3 & 4 \\
5 & 6 \\
7 & 8 \\
\end{bmatrix}
\]

\[
H(:,:,4) = \\
\begin{bmatrix}
1 & 0 \\
2 & 3 \\
4 & 5 \\
6 & 7 \\
\end{bmatrix}
\]

\[
>> [E, theta, B] = \text{minPMAC}(H, 4*b, w)
\]

\[
E = \\
\begin{bmatrix}
1.6187 & 1.6184 & 0.0008 & 0.0000 \\
0.0002 & 0.0000 & 1.6180 & 1.6186 \\
\end{bmatrix}
\]

\[
theta = \\
\begin{bmatrix}
3.2380 \\
3.2384 \\
\end{bmatrix}
\]

\[
B = \\
\begin{bmatrix}
6.3303 & 5.6692 & 0.0004 & 0.0000 \\
0.3391 & 0.0002 & 5.9911 & 5.6696 \\
\end{bmatrix}
\]

With different gains, the energy/bit distributions correspondingly change, and appears to
trend toward an FDM solution for minimum energy sum.

For a more sophisticated example, \( U = 3 \) and all 3 users have crosstalk and temporal intersymbol
interference.

**EXAMPLE 5.4.8 [3-user Vector Gaussian MAC with 2-dimensional output]** A
(complex baseband) vector ISI MAC has 3 user inputs and a two-dimensional output, essentially guaranteeing secondary-user existence for non-trivial ISI. The noise is white with variance \( N_0 = .01 \). The channel is

\[
H(D) = \begin{bmatrix}
1 + .9D \\
.5D - AD^2 \\
\end{bmatrix}
\]

The \( D \)-transform roughly allows the designer to see user 3 as roughly lowpass, user 2 as
somewhat bandpass, and user 1 as highpass. This example first computes data rates for
equal energies, starting with the matlab commands

\[
h=\text{cat}(3,[1 0 .8 ; 0 1 1],[.9 -.3 0 ; .5 -1 -1],[0 .2 0 ; .4 -.63 0],[0 0 0 ; 0 .648 0])*10;
\]

\[
h(:,:,1) = \\
\begin{bmatrix}
10 & 0 & 8 \\
0 & 10 & 10 \\
\end{bmatrix}
\]

\[
h(:,:,2) = \\
\begin{bmatrix}
9 & -3 & 0 \\
5 & -10 & -10 \\
\end{bmatrix}
\]

\[
h(:,:,3) = \\
\begin{bmatrix}
0 & 2.0000 & 0 \\
4.0000 & -6.3000 & 0 \\
\end{bmatrix}
\]

\[
h(:,:,4) = \\
\begin{bmatrix}
0 & 0 & 0 \\
0 & 6.4800 & 0 \\
\end{bmatrix}
\]

before continuing to minPMAC use. The DFT for vector DMT with \( N = 8 \) is then

\[
>> N=8;
\]

\[
H = \text{fft}(h, N, 3)/\text{sqrt}(N)
\]

\[
>> H = \text{fft}(h, N, 3)
\]

\[
H(:,:,1) = \\
\begin{bmatrix}
19.0000 + 0.0000i & -1.0000 + 0.0000i & 8.0000 + 0.0000i \\
\end{bmatrix}
\]
9.0000 + 0.0000i 0.1800 + 0.0000i 0.0000 + 0.0000i
H(:,:,2) =
16.3640 - 6.3640i -2.1213 + 0.1213i 8.0000 + 0.0000i
3.5355 - 7.5355i -1.6531 + 8.7890i 2.9289 + 7.0711i
H(:,:,3) =
10.0000 - 9.0000i -2.0000 + 3.0000i 8.0000 + 0.0000i
-4.0000 - 5.0000i 16.3000 +16.4800i 10.0000 +10.0000i
H(:,:,4) =
3.6360 - 6.3640i 2.1213 + 4.1213i 8.0000 + 0.0000i
-3.5355 + 0.4645i 21.6531 - 3.8110i 17.0711 + 7.0711i
H(:,:,5) =
1.0000 + 0.0000i 5.0000 + 0.0000i 8.0000 + 0.0000i
-1.0000 + 0.0000i 7.2200 + 0.0000i 20.0000 + 0.0000i
H(:,:,6) =
3.6360 + 6.3640i 2.1213 - 4.1213i 8.0000 + 0.0000i
-3.5355 - 0.4645i 21.6531 + 3.8110i 17.0711 - 7.0711i
H(:,:,7) =
10.0000 + 9.0000i -2.0000 - 3.0000i 8.0000 + 0.0000i
-4.0000 + 5.0000i 16.3000 -16.4800i 10.0000 -10.0000i
H(:,:,8) =
16.3640 + 6.3640i -2.1213 - 0.1213i 8.0000 + 0.0000i
3.5355 + 7.5355i -1.6531 - 8.7890i 2.9289 - 7.0711i

Use of equal energy on all tones and users, assuming 1 unit of energy when \( N = 1 \) provides Figure 5.33 when \( N \) is increased. The noise-whitened normalization of \( \tilde{H} \) causes an equal scaling by \( \sqrt{N} \) in both channel transfer/tone from an orthogonal DFT (with thus \( 1/\sqrt{\tilde{N}} \) scale factor) \( \tilde{H} \cdot \tilde{H}^* \to \tilde{N} \cdot \tilde{H} \cdot \tilde{H}^* \), but this happens also for the noise now over \( \tilde{N} \) samples in a packet, so the \( \tilde{N} \)'s cancel in the frequency-domain inputs of the \texttt{mumac.m} program. The input energy to the \texttt{mu_mac} program also takes energy per tone for each user, which for \( \nu = 3 \) offsets to \( \tilde{N}/(\tilde{N} + \nu) \):

\[
\begin{align*}
N_{max} &= 32; \\
U &= 3; \\
Ly &= 2; \\
cb &= 1; \\
Usize &= [1 1 1]; \\
bsum &= \text{zeros}(1,N_{max});
\end{align*}
\]

for index=1:Nmax
  i=2*index; \( \% \) (don't need to plot a point for every number of tones)
  H = \text{fft}(h, i, 3);
  GU=zeros(U,U,i);
  WU=zeros(U,U,i);
  S0=zeros(U,U,i);
  Bu=zeros(U,i);
  MSWMFU=zeros(U,Ly,i);
  AU=zeros(3,3,i);
  for n=1:i
    AU(:,:,n)=sqrt(i)/sqrt(i+3)*eye(3);
  end
  for n=1:i
    [Bu(:,n), GU(:,n) , WU(:,n), S0(:,n) , MSWMFU(:,n)] = ... \\
    \text{mu_mac}(H(:,n), AU(:,n), Usize, cb);
  end
end
Bvec=sum(Bu');
bsum(index)=sum(bvec);
end
plot(2*[1:Nmax], bsum)
bvec = 445.1264  412.8794  132.7477
sum(bvec) = 990.7535
N=64;
for n=1:N
Rnn(:,:,n) = eye(2);
end
[[Rxx, bsum, bsumlin] = SWF(N/(N+3)*[1 1 1], H, [1 1 1], Rnn, 1);
>> bsum = 1011.1 > 990.8
>> bsumlin = 578.8502- linear loss almost 50 \%

The sum data rate for equal users’ energy per temporal dimension approaches about 15.2 bits/tone within each symbol (longer symbols of course have higher total number of bits/symbol). There are also two spatial dimensions, so this is about 7 bits/spatial dimension for each dimension within a symbol. For $N \geq 64$, there is very little additional rate-sum gain. Since maximum rate sums often take a smaller $N^*$ than full flat-bandwidth input spectra, this example will proceed with $N = 64$. Some illustrative quantities are feedback sections and overall feedforward processing for the receiver, so by selecting some tones:

$$
\text{GU(:,:,23)} =
\begin{bmatrix}
1.0000 + 0.0000i & -1.3365 + 0.3447i & -0.3093 + 0.3130i \\
0.0000 + 0.0000i & 1.0000 + 0.0000i & 0.8108 + 0.4218i \\
0.0000 + 0.0000i & 0.0000 + 0.0000i & 1.0000 + 0.0000i \\
\end{bmatrix}
$$

$$
\text{WU(:,:,37)} =
\begin{bmatrix}
0.0558 - 0.0000i & 0.0000 - 0.0000i & 0.0000 + 0.0000i \\
0.0102 - 0.0066i & 0.0067 - 0.0000i & 0.0000 + 0.0000i \\
-0.2761 - 0.0233i & -0.1830 - 0.1201i & 0.1328 + 0.0000i \\
\end{bmatrix}
$$

$$
\text{MSWMFU(:,:,15)} =
\begin{bmatrix}
0.0454 + 0.0341i & -0.0105 + 0.0249i \\
0.0183 + 0.0179i & 0.0275 - 0.0468i \\
0.0589 + 0.0199i & 0.0179 - 0.0416i \\
\end{bmatrix}
$$

$$
\text{SO(:,:,29)} =
\begin{bmatrix}
18.9071 & 0 & 0 \\
0 & 149.3696 & 0 \\
0 & 0 & 8.5326 \\
\end{bmatrix}
$$

$$
10\times\text{log10(diag(S0(:,:,29)))} =
\begin{bmatrix}
12.7663 \text{ dB} \\
21.7426 \text{ dB} \\
9.3108 \text{ dB} \\
\end{bmatrix}
$$

The nonzero feedback sections illustrate best-performance’s need for the nonlinear GDFE/MAC operations to obtain highest sum data rate, albeit these feedback sections were for the design with equal energy (not the SWF Rxx distribution). Inspection of a few randomly chosen tones’ $R_{xx}(n)$ values also shows that all 3 users are energized because this is for vector MAC.

$$
\text{Rxx(:,:,23)} =
\begin{bmatrix}
0.7615 & 0 & 0 \\
0 & 1.1885 & 0 \\
0 & 0 & 0.9480 \\
\end{bmatrix}
$$
(Example 5.4.8 continued) A single feedback section is shown to be non-trivial on this channel (tone 22 corresponds to n=23 in matlab). The bit distributions appear in Figure 5.34. These are for a complex channel, so remembering that minPMAC works in bits/real-dimension, a design for the bit distribution \(b = \frac{1}{2} \cdot [520 \ 480 \ 130]\) is clearly admissible, but a design that minimizes the sum of energies for the 3 users would follow through use of minPmac. The \(b_{min}\) input to minPMAC is in bits/user viewed as real dimensions (so the \(1/2\log_2\) factor is included in minPMAC in interpreting the sum of each users bits/dimension). The output bit distribution is similarly in bits/real-dimension. So the input bit-rate target \(b\) is scaled by \(1/2\) if the channel is baseband complex and the output bits/dimension can be doubled for complex baseband. The energy outputs are also in bits/real-dimension, so for instance if a complex baseband input channel has \(N\) tones, then to get the average energy/dimension, a users' total energy for a user can be added over the tones (the minPmac output \(E\) distribution) and then divided by \(N = 2N\); however the division would be by \(N = N\) in the case of a real baseband channel. This example is complex, and so the following steps illustrate the correct use of minPMAC when \(N = 64\):

\[
\text{bu}_{\text{min}} = 0.5 \cdot [440 \ 410 \ 130]' \; \% \text{slightly < flat-energy earlier, times } 1/2
\]

\[
w = [1 \; 1 \; 1];
\]

\[
\text{Lxu} = 1;
\]

\[
>> [E, \theta, b] = \text{minPMAC}(H, \text{bu}_{\text{min}}, w);
\]

\[
>> \text{sum}(b') = 219.9070 \ 204.9593 \ 64.9617
\]

\[
>> \text{sum}(E') = 1.4910 \ 1.2218 \ 0.4903
\]

\[
>> \theta' = 0.0812 \ 0.0801 \ 0.0548
\]

\[
>> \text{sum(sum}(E')) = 3.2030 \ (> 3)
\]
which does have energies that are above those for the equal-energy case of 1 unit on each user. The data rates are very close to the targets.

The margin would be determined by the ratio.

\[
\Gamma = 6 \times \left(2^{\left(\frac{\text{sum}(\text{sum}(b))}{67}\right)} - 1\right) / \left(2^{\left(2 \times \text{sum}(\text{sum}(B_u))/67\right)} - 1\right) = -6.0000 \text{ dB}
\]

or effectively about 1 of a bit/real-dimension below fundamental limit on a radial line through the point to \( \mathcal{C}(b) \). The corresponding MAC receiver design for this alternative data rate close to maximum rate sum is then given by:

\[
\begin{align*}
\text{cb} &= 1; \\
\text{Usize} &= [1 \ 1 \ 1]; \\
\text{Bu} &= \text{zeros}(64,3); \\
\text{Aopt} &= \text{zeros}(3,3,64); \\
\text{for } n = 1:N \\
\quad \text{Aopt}(:,:,n) &= N \times \text{diag}(E(:,n)); \\
\text{end} \\
\text{for } n = 1:N \\
\quad [\text{Bu}(n,:), \text{GU}(:,:,n), \text{WU}(:,:,n), \text{S0}(:,:,n), \text{MSWMFU}(:,:,n)] &= \ldots \\
\quad \text{mu_mac}(:, :, n), \text{Aopt}(:, :, n), \text{Usize} , \text{cb}; \\
\text{end} \\
\text{sum}(\text{sum}(\text{Bu}')) &= 1044.9
\end{align*}
\]

\[
\begin{align*}
\text{GU}(:, :, 23) &= \\
&= \begin{bmatrix}
1.0000 + 0.0000i & -2.1570 + 0.5562i & -0.0704 + 0.0712i \\
0.0000 + 0.0000i & 1.0000 + 0.0000i & 0.1145 + 0.0597i \\
0.0000 + 0.0000i & 0.0000 + 0.0000i & 1.0000 + 0.0000i
\end{bmatrix}
\end{align*}
\]

The feedback section for the same tone changes slightly because the input energies changed slightly. Also, since minPMAC found an energy set that violates at least one user energy level, the sum rate of 1045 (or 15.6 bits/tone) exceeds the previous data rates because those were under strict adherence to energy constraints. The maximum sum rate for the energy-sum MAC (so 3 \((64/67)\) total energy for BC follows from the macmax program.

\[
\begin{align*}
\text{[Rxx, bsum, bsum lin]} &= \text{macmax}(3 \times (64/67), h, \text{Usize, N, 1}); \\
\text{bsum} &= 1011.3 \\
\text{bsum lin} &= 571.7289
\end{align*}
\]

This data rate is also higher and is a maximum when the total symbol energy for all 3 MAC users is limited to \( 3 \times 64 \cdot (64/67) \). Also, the linear-only receiver loses roughly half the data rate. Thus, for this highpass-bandpass-lowpass user-channel combination, the linear system only is very limited.

The dual channel (see Section 5.5 to understand the remainder of this example) maximum data rate should be the same and follows from

\[
\begin{align*}
\text{>> J2} &= \text{hankel([0 1]);} \\
\text{J3} &= \text{hankel([0 0 1]);} \\
\text{rxxb} &= \text{zeros}(2,2,64); \\
\text{Hbc} &= \text{zeros}(3,2,64); \\
\text{bbc} &= \text{zeros}(3,64); \\
\text{for } n = 1:64 \\
\quad \text{Hbc}(:,:,n) &= \text{J3} \times \text{H}(:,:,n) \times \text{J2}; \\
\quad \text{rxxb}(:,:,n) &= (1.5/67) \times \text{eye}(2); \\
\text{end}
\end{align*}
\]
[oRxx, Rwcn, bmax] = bcmax(rxxb, Hbc, 1);
bmax = 505.6484
2*bmax=1101.3
2*bmax/67 = 15.0940

so as expected the the max rate sum of the Esum MAC matches its dual maximum rate sum.

5.4.4 Tracing the Rate Region and Admissible MAC Designs

The capacity region $C_{MAC}(b)$ recognizes the $U$ individual energy constraints, one for each user, in an energy vector $E_X$. Energy-sum minimization can have meaning in the context of all MAC transmitters' radiated energy (into other systems). It is also crucial to Section 5.5’s duality. However, the Section 5.4.3.3’s minPmac can find (with minor modification) a solution that also satisfies the energy constraints. Subsection 5.4.4.1 describes a modified procedure that can trace $C(b)$.

5.4.4.1 The MAC with energy-vector constraint

The energy-vector constraint objective modifies to:

$$\min \{R_{XX}(u,n)\}$$

$$ST: 0 \leq b_{min} \leq \sum_n [b_{1,n}, b_{2,n}, ..., b_{U,n}]$$

$$0 \leq \sum_n \text{trace} \{R_{XX}(u,n)\} \leq \text{[E}_1 E_2 ... E_U]^* = E$$

$$b_n \in A_n(R_n, \{R_{XX}(u,n)\})_{u=1,...,U}.$$ 

The problem in (5.343) is the same as the minimum-energy-sum-constrained problem, except that (5.343) adds the new $E$ constraint. The new Lagrangian becomes

$$L(R_{XX}, b, w, \theta) = \sum_{u=1}^{U} w_u \left[ \sum_{n=1}^{N} \text{trace} \{R_{XX}(u,n)\} - E_u \right] - \theta_u \left[ \sum_{n=1}^{N} b_{u,n} - b_u \right],$$

(5.344)
with both \( w \geq 0 \) and \( \theta \geq 0 \) now as constraint variables to be determined along with the best set of positive-semi-definite autocorrelation matrices. Previously \( x \) was given; now the optimization also determines \( w \) for a given energy vector. Thus, after the Lagrangian’s minimization over the set of autocorrelation matrices for any given \( \theta \) and \( w \), then the order step becomes a function of both \( \theta \) and \( w \), so

\[
L_{\min}(w, \theta) = \min_{\{R_{XX}(u,n)\}} \{L(R_{XX}, b, w, \theta) \leq 0 \}.
\]

(5.345)

The minimized Lagrangian is only positive (by inspection) if (not all) the constraints cannot be satisfied for the given \( b \). If such a positive result occurs for any \( \theta \geq 0 \) and/or \( w \geq 0 \), then no solution exists. Then either \( b \) must reduce for at least one user and/or \( E \) must increase for at least one user.

**Enlarged-parameter MAC Optimization:** Mohseni’s algorithm still applies, but the “order” step now enlarges ellipsoid dimensionality to include both \( \theta \) and \( w \), which combine into a larger vector \( \tilde{\theta} = [w, \theta] \). If after any \( R_{XX} \) step in Mohseni’s algorithm, a positive \( L_{\min}(w, \theta) \) occurs, then the algorithm terminates and declares that the energy constraints \( E \) (or equivalently \( b \)) cannot be met. With

\[
\Delta E \equiv \begin{bmatrix}
\sum_n \text{trace} \{R_{XX}(1,n)\} - E_1 \\
:\:\:
\sum_n \text{trace} \{R_{XX}(U,n)\} - E_U
\end{bmatrix},
\]

(5.346)

the sub-gradient vector now becomes

\[
g_k = \begin{bmatrix} \Delta E \\ \Delta b \end{bmatrix}.
\]

(5.347)

and the recursions now use normalized sub-gradient

\[
\hat{g} \equiv \frac{g_k}{\sqrt{g_k \cdot A_k^{-1} \cdot g_k}},
\]

(5.348)

which simplifies to:

\[
\tilde{\theta}_{k+1} = \tilde{\theta}_k - \frac{1}{U + 1} \cdot A_k \cdot \hat{g}_k,
\]

(5.349)

\[
A_{k+1} = \frac{4 \cdot U^2}{4 \cdot U^2 - 1} \cdot \left[A_k - \frac{2}{2 \cdot U + 1} A_k \cdot \hat{g}_k \cdot \hat{g}_k^* \cdot A_k\right],
\]

(5.350)

The volume of the ellipsoid decreases as

\[
\text{vol} (O_{k+1}) \leq e^{-\frac{1}{4U}} \cdot \text{vol} (O_k).
\]

(5.351)

Ellipsoid initialization is easier in this case. Because the Lagrange multipliers \( w \) and \( \theta \) could be arbitrarily scaled, essentially any ellipse that satisfies the positivity constraints would contain a solution if such a solution exists. Thus an acceptable initialization covers the unit cube and would then be:

\[
A_0^{-1} = \begin{bmatrix}
1 & 0 & \ldots & 0 \\
0 & \ddots & \ddots & \ddots \\
\vdots & \ddots & \ddots & \ddots \\
0 & \ldots & 0 & 1
\end{bmatrix}.
\]

(5.352)

The \( R_{XX} \) step remains the same for each tone with the \( w \) and \( \theta \) from previous iterations (the first iteration can set both equal to all ones). The algorithm computes the tonal Lagrangian in exactly the same manner, but the overall sum should conform to the Lagrangian in (5.344).
The capacity region is then found by gradually increasing the elements of \( b \) in a nested \( U \)-loop that calls Mohseni’s algorithm and checks to see if \( b \) has a solution. If the algorithm does not abort, then the \( b \) point will be in the capacity region. If not, the point is not in the capacity region. This is sometimes called the admission problem to see if \( b \) is an “admissible” data rate vector.

There are two programs that follow. The first admMAC (original version written by Mohseni), current version slightly modified determines admissibility of a rate vector \( b \) with an energy-vector constraint \( \mathcal{E} \). admMAC also provides an output bit vector \( b_{u,n} \) and energy distribution \( \mathcal{E}_{u,n} \); however, admMAC’s bit and energy distributions can fail when time-sharing is necessary of different rate-region vertices. admMAC must also have \( L_{x,u} = 1 \) for all users. A second enhanced program that uses CVX optimization (but takes longer to run) called admMAC_mimo will accept any \( L_{x,u} \) and provides energy and bit distributions and the fractions of time/dimensions they must be shared if the only way to achieve the given input \( b \) is with time-sharing.

The admMAC program: Mohseni also provides the admMAC program for the admissibility. This program also makes use of the minPtone, Hessian, and eval_f subroutines from Section 5.4.3’s minPMAC. The program admMAC also uses a subroutine called dual_adm also that appears here.

The inputs to the admMAC function are

- \( H \) (or \( \bar{H} \)) is the same \( L_y \times U \times N \) channel-tensor of minPmac \((L_x = 1)\).
- \( b_u \) is a \( U \times 1 \) non-negative user bits-per-symbol vector \( b \).
- \( E_u \) is the \( U \times 1 \) vector \( E_{vec} \) of non-negative energies per symbol for each user.

The outputs from the function are

- \( E \), a \( U \times N \) matrix containing the energies \( \mathcal{E}_{u,n} \). An all zero \( E \) indicates that the given rates in \( b \) are not achievable with given powers in \( \mathcal{E}_x \).
- \( b \) is a \( U \times N \) matrix of \( b_{u,n} \). Again, all zero rates indicates that given \( b \) is not achievable with the given \( \mathcal{E} \).
- \( \theta \), the \( U \times 1 \) user-rate weight vector.
- \( w \), the \( U \times 1 \) user-energy weight vector.
- \( f \), the minimized Lagrangian value (negative of \( L \)).

The admMAC program:

The inputs to the admMAC function are

- \( H \) (or \( \bar{H} \)) is the same \( L_y \times U \times N \) channel-tensor of minPmac \((L_x = 1)\).
- \( b_u \) is a \( U \times 1 \) non-negative user bits-per-symbol vector \( b \).
- \( E_u \) is the \( U \times 1 \) vector \( E_{vec} \) of non-negative energies per symbol for each user.

The outputs from the function are

- \( E \), a \( U \times N \) matrix containing the energies \( \mathcal{E}_{u,n} \). An all zero \( E \) indicates that the given rates in \( b \) are not achievable with given powers in \( \mathcal{E}_x \).
- \( b \) is a \( U \times N \) matrix of \( b_{u,n} \). Again, all zero rates indicates that given \( b \) is not achievable with the given \( \mathcal{E} \).
- \( \theta \), the \( U \times 1 \) user-rate weight vector.
- \( w \), the \( U \times 1 \) user-energy weight vector.
- \( f \), the minimized Lagrangian value (negative of \( L \)).

The admMAC program:

Mohseni also provides the admMAC program for the admissibility. This program also makes use of the minPtone, Hessian, and eval_f subroutines from Section 5.4.3’s minPMAC. The program admMAC also uses a subroutine called dual_adm also that appears here.

The inputs to the admMAC function are

- \( H \) (or \( \bar{H} \)) is the same \( L_y \times U \times N \) channel-tensor of minPmac \((L_x = 1)\).
- \( b_u \) is a \( U \times 1 \) non-negative user bits-per-symbol vector \( b \).
- \( E_u \) is the \( U \times 1 \) vector \( E_{vec} \) of non-negative energies per symbol for each user.

The outputs from the function are

- \( E \), a \( U \times N \) matrix containing the energies \( \mathcal{E}_{u,n} \). An all zero \( E \) indicates that the given rates in \( b \) are not achievable with given powers in \( \mathcal{E}_x \).
- \( b \) is a \( U \times N \) matrix of \( b_{u,n} \). Again, all zero rates indicates that given \( b \) is not achievable with the given \( \mathcal{E} \).
- \( \theta \), the \( U \times 1 \) user-rate weight vector.
- \( w \), the \( U \times 1 \) user-energy weight vector.
- \( f \), the minimized Lagrangian value (negative of \( L \)).
in bu are not achievable with given powers in Eu.

2) b, a U by N matrix containing the rates of all users on all tones after convergence. Again, all zero rates indicates that given bu is not achievable with Eu.

3) theta, the optimal U by 1 dual variable vector containing optimal weights of rates. theta determines the decoding order.

4) w, the optimal U by 1 dual variable vector containing optimal weights of powers.

5) f, the dual optimal value.

AdmMAC calls function dual_adm, which computes the Lagrangian dual function as the sum of the results from each tone.

>> help Dual_adm

function [f, b, E] = Dual_adm(H, theta, w, bu, Eu);
this function computes the Lagrange dual function by solving the optimization problem (calling the function minPtone) on each tone.
the inputs are:
1) H, an Ly by U by N channel matrix. Ly is the number of receiver antennas, U is the total number of users and N is the total number of tones. 
   H(:,;:;n) is the channel matrix for all users on tone n and H(:,;:;u,n) is the channel for user u on tone n. In this code we assume each user only has single transmit antenna, thus H(:,;:;u,n) is a column vector.
2) theta, a U by 1 vector containing the weights for the rates.
3) w, a U by 1 vector containing the weights for each user’s power.
4) bu, a U by 1 vector containing the target rates for all the users.
5) Eu, a U by 1 vector containing the power constraints for all the users.

the outputs are:
1) f, the Lagrange dual function value.
2) b, a U by N vector containing the rates for all users and over all tones that optimizes the Lagrangian. b(u,:) is the rate allocation for user u over all tones.
3) E, a U by N vector containing the powers for all users and over all tones that optimizes the Lagrangian. E(u,:) is the power allocation for user u over all tones.

A more sophisticated version of admMAC credits Dr. Y. Liao and allows for variable antenna size in input Lxu and as well finds and reports a time-sharing solution if that is necessary to achieve the desired data rate. This is admMAC_rate_region.

>> help admMAC_rate_region

function [FEAS_FLAG, bu_a, info] = admMAC_rate_region(H, Lxu, bu, Eu)

admMAC_rate_region determines whether the target rate vector bu is feasible for (noise-whitened) channel H and energy/symbol Eu via rate region

Input arguments:
- Lxu: number of transmit antennas of each user. It can be either a scalar or a length-U vector. If it is a scalar, every user has Lxu transmit antennas; otherwise user u has Lxu(u) transmit antennas.
- **bu**: target rate of each user, length-U vector.
- **Eu**: Energy/symbol on each user, length-U vector.

**Outputs:**
- **FEAS FLAG**: indicator of achievability. FEAS FLAG=0 if the target is not achievable; FEAS FLAG=1 if the target is achievable by a single ordering; FEAS FLAG=2 if the target is achievable by time-sharing.
- **bu_a**: U-by-1 vector showing achieved sum rate of each user.
- **info**: various length output depending on FEAS FLAG
  - if FEAS FLAG=0: empty
  - if FEAS FLAG=1: 1-by-5 cell array containing {Rxxs, Eun, bun, theta, w} corresponds to the single vertex
  - if FEAS FLAG=2: v-by-6 cell array, with each row representing a time-shared vertex {fraction, Rxxs, Eun, bun, theta, w}

- **Rxxs**: U-by-N cell array containing Rxx(u,n)’s if Lxu is a length-U vector; or Lxu-by-Lxu-by-U-by-N tensor if Lxu is a scalar. If the rate target is infeasible, output 0.
- **Eun**: U-by-N matrix showing users’ transmit power on each tone. If infeasible, output 0.
- **bun**: U-by-N matrix showing users’ rate on each tone. If infeasible, output 0.
- **theta**: U-by-1 Lagrangian multiplier w.r.t. target rates
- **w**: U-by-1 Lagrangian multiplier w.r.t. power constraints

**EXAMPLE 5.4.9 (Time-Share Example)** This example revisits the well-known 80 60 channel to achieve a point that requires time sharing from previous invocations:

\[
H = \begin{bmatrix} 80 & 60 \end{bmatrix};
\]

\[
\text{>> } bu_{\text{min}} = \begin{bmatrix} 1.72; 4.92 \end{bmatrix};
\]

\[
\text{>> } Eu = \begin{bmatrix} 1; 1 \end{bmatrix};
\]

\[
\text{>> } [\text{flag}, \text{bu}_{\text{achieved}}, \text{info}] = \text{admMAC\_rate\_region}(H, 1, bu_{\text{min}}, Eu)
\]

**flag** = 2 (indicates time share)

**bu\_achieved** =

\[
\begin{align*}
1.7200 \\
4.9200
\end{align*}
\]

**info** =

\[
\text{table}
\]

<table>
<thead>
<tr>
<th>bu_v</th>
<th>Rxxs</th>
<th>Eun</th>
<th>bun</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6.322</td>
<td>0.32189</td>
<td>{2x1 double}</td>
<td>{2x1 double}</td>
</tr>
<tr>
<td>0.73684</td>
<td>5.9071</td>
<td>{2x1 double}</td>
<td>{2x double}</td>
</tr>
</tbody>
</table>

**theta**

**w**

**frac**

\[
\begin{align*}
\text{\{2x1 double\}} & \quad \text{\{2x1 double\}} & 0.17611 \\
\text{\{2x1 double\}} & \quad \text{\{2x1 double\}} & 0.8233
\end{align*}
\]

So the vertex \(b^*_{\text{A}} = [6.322 \ 3.22]\) occurs roughly 17/6% of the uses and the other vertex
$b^*_B = [0.737 \ 5.907]$ occurs the remaining 82.3\% of the uses.

### 5.4.4.2 Examples of the use of admMAC

This section returns to the Example 5.4.3.3 to find the admissibility and then also the proper weight vector for use in minPMAC.

**EXAMPLE 5.4.10 (scalar MAC revisited)** Example 5.4.7’s scalar MAC channel has a data rate of 3 bits per symbol on each of the two users.

```matlab
>> H
H(:,:,1) = 80 60
H(:,:,2) = 80 60
H(:,:,3) = 80 60
H(:,:,4) = 80 60
>> b =
  3
  3
>> energy = [1
  1];
>> [E, B, theta, w, L] = admMAC(H, b\*4, 4\*energy)
count = 352
E =
1.0690 1.0690 1.0690 1.0690
0.7625 0.7625 0.7625 0.7625
B =
6.3702 6.3702 6.3702 6.3702
0.2433 0.2433 0.2433 0.2433
theta = 1.0e-07 *
  0.4042
  0.3013
w = 1.0e-06 *
  0.1084
  0.1368
L = 2.4746e-07
>> [Eopt, thetaopt, Bopt] = minPMAC(H, 4\*b, w)
count = 158
Eopt =
0.9227 0.9227 0.9227 0.9227
0.7312 0.7312 0.7312 0.7312
thetaopt = 1.0e-09 *
  0.1150
  0.1493
Bopt =
0.8485 0.8485 0.8485 0.8485
5.6814 5.6814 5.6814 5.6814
>> [Eopt, thetaopt, Bopt] = minPMAC(H, 4\*b, 1e6\*w)
count = 200
Eopt =
0.6300 0.6300 0.6300 0.6300
0.0175 0.0175 0.0175 0.0175
thetaopt =
  0.1388
  0.1415
```

930
Bopt =
3.0000 3.0000 3.0000 3.0000
3.0000 3.0000 3.0000 3.0000

The first instance of the use of minPMAC has numerical problems, but notice the uniform scaling of the w weighting of the users' energy removes the issue on the second instance. Continuing, the rate pair (2,4) produces:

```matlab
>> b=[2
4];
>> energy=[1
1];
>> [E, B, theta, w, L]=admMAC(H, 4*b, 4*energy)
count = 343
E =
0.9450 0.9450 0.9450 0.9450
1.1697 1.1697 1.1697 1.1697
B =
0.6422 0.6422 0.6422 0.6422
6.0201 6.0201 6.0201 6.0201
theta = 1.0e-07 *
0.1978
0.2054
w = 1.0e-06 *
0.1120
0.0893
L = 4.6516e-09
```

```matlab
>> Htemp(1,:,:)=80 60
(REPEATED 4 TIMES)
>> [Eopt, thetaopt, Bopt]=minPMAC(H, 4*b, 1e6*w)
count = 199
Eopt =
0.6000 0.6000 0.6000 0.6000
0.0708 0.0708 0.0708 0.0708
thetaopt =
0.1433
0.1471
Bopt =
2.0001 2.0001 2.0001 2.0001
3.9999 3.9999 3.9999 3.9999
```

However, increasing the rates to just outside the region produces:

```matlab
>> [E, B, theta, w, L]=admMAC(H, 5*b, 4*energy)
E = 0 0 0 0
0 0 0 0
B = 0 0 0 0
0 0 0 0
theta =
1.1029
1.2991
w =
The negative Lagrangian value means the point is outside the capacity region and thus cannot be realized within the energy constraints.

For the inter-symbol interference example of Section 2.7 a similar investigation produces

**EXAMPLE 5.4.11**

```matlab
>> H=zeros(1,2,8);
>> H(1,1,:)=fft([1 .9],8);
>> H(1,2,:)=fft([1 -1],8);
>> H=(1/sqrt(8*.181))*H;
>> b=[1
1];
>> energy=[8
8];
```

This higher energy essentially corresponds to the 8 dimensions each with one energy unit. Additionally, these 8’s are scaled by the 8/9 cycle prefix factor loss to get 64/9.

```matlab
>> [E, B, theta, w, L]=admMAC(H, 9*b, (64/9)*energy)
count = 381
E =
  5.6099  5.9185  6.9030  7.8390  8.1115  7.8390  6.9030  5.9185
B =
  2.3424  2.2089  1.7668  0.9405  0.0294  0.9405  1.7668  2.2089
  0  0.0912  0.4967  1.4076  2.3744  1.4076  0.4967  0.0912
theta = 1.0e-06 *
  0.1376
  0.0926
w = 1.0e-07 *
  0.2016
  0.1783
L = 2.0255e-07
```

```matlab
>> [Eopt, thetaopt, Bopt]=minPMAC(H, 9*b, 1e7*w)
count = 189
Eopt =
  8.4687  8.4117  2.2727  0.0000  0.0000  0.0000  2.2727  8.4117
  0.0000  0.0000  6.6326  8.9386  8.9902  8.9386  6.6326  0.0000
thetaopt =
  3.5490
  3.3124
```

So this is a valid solution because the total energy is less than 64/9 for each user and the bit rates are those desired.

Lastly, the 64-tone 3-user channel from earlier is revisited here:

**EXAMPLE 5.4.12** 3-user 64-tone Example revisited with admMAC  An alternative approach to the 3-user design of Example 5.4.2 might use admMAC
\[ [E, b, \theta, w, f] = \text{admMAC}(H, \text{bu}\_\text{min}, 64/67*\text{ones}(3,1)); \]
\[ \text{sum}(b') = 220.0081 \quad 206.0735 \quad 68.2555 \]
\[ \text{sum}(E') = 59.9944 \quad 59.1474 \quad 58.0499 \]
\[ w' = 0.0708 \quad 0.0122 \quad 0.0019 \]
\[ \theta' = 1.4047 \quad 0.1777 \quad 0.0019 \]
\[ f = 327.3138 \]
\[ \text{for } n=1:N \]
\[ \text{Aopt}(:,:,n) = \text{diag}(E(:,n)); \]
\[ \text{end} \]
\[ \text{for } n=1:N \]
\[ \text{[Bu(n,:), GU(:,:,n), WU(:,:,n), S0(:,:,n), MSWMFU(:,:,n)] = ...} \]
\[ \mu\_\text{mac}(H(:,:,n), \text{Aopt}(:,:,n), \text{Usize}, \text{cb}); \]
\[ \text{end} \]
\[ \text{GU}(:,:,23) = \]
\begin{align*}
1.0000 + 0.0000i & \quad -1.3365 + 0.3447i & \quad -0.3093 + 0.3130i \\
0.0000 + 0.0000i & \quad 1.0000 + 0.0000i & \quad 0.7327 + 0.3367i \\
0.0000 + 0.0000i & \quad 0.0000 + 0.0000i & \quad 1.0000 + 0.0000i
\end{align*}

where the user energies no longer have equal weight in the minimized energy sum, so a different solution emerges that actually has slightly better data rates with more energy on user 1 but still meets (roughly) the energy-vector constraint.

Another example:

\[ h = \text{cat}(3,[1 \ 0 \ .8 \ ; \ 0 \ 1 \ 1], [.9 -.3 \ 0 \ ; \ .5 -1 -1], [0 \ .2 \ 0 \ ; \ .4 -.63 \ 0], [0 \ 0 \ 0 \ ; \ 0 \ .648 \ 0])*1000; \]
\[ \text{N}=512; \]
\[ H = \text{fft}(h, \text{N}, 3)/\text{sqrt}(\text{N}); \]
\[ \text{bu}\_\text{min} = [1040 \ 1360 \ 680 ]'/2; \]
\[ \text{Eu} = 512/515*[1 \ 1 \ 1]'; \]
\[ \text{[flag, bu}\_\text{achieved}, \text{info}] = \text{admMAC}\_\text{rate}\_\text{region}(H, 1, \text{bu}\_\text{min}, \text{Eu}) \]
\[ \text{[flag, bu}\_\text{achieved}, \text{info}] = \text{admMAC}\_\text{rate}\_\text{region}(H, 1, \text{bu}\_\text{min}, \text{Eu}) \]

\[ \text{flag} = 1 \]
\[ \text{bu}\_\text{achieved} = \]
\begin{align*}
720.1222 \\
712.8740 \\
403.2180
\end{align*}

\[ \text{info} = \]
\begin{table}[h]
\begin{tabular}{cccc}
\hline
bu\_v & Rxxs & Eun & bun \\
\hline
720.12 & 712.87 & 403.22 & {3x512 double} \\
\hline
\end{tabular}
\end{table}

\[ \text{theta} \quad \text{w} \]
\[ \begin{align*}
\text{theta} & \quad \text{w} \\
\{3x1 \text{ double}\} & \quad \{3x1 \text{ double}\}
\end{align*} \]
**Comment on ODM:** Cellular and Wi-Fi wireless systems often aggregate many tones into resource blocks (cellular) or “channels” (Wi-Fi) for which the energy/tone is constant for all tones within the tone set. Such systems can treat each set as a single tone in the use of minPMAC and admMAC processes. A restriction to OMA (orthogonal multiple access) means no user sharing, or no crosstalk allowed. Clearly such systems cannot improve performance over the “NOMA” optimums, and indeed have more complex searches necessary to calculate best ODM dimensional use. It may nonetheless with a sufficient number of tone sets (which may occur through MIMO’s ability to allocate different energy to different spatial dimensions) to see an ODM like nature to the minPMAC and admMAC solutions, which provides a simpler estimate. Indeed, such ODM restriction, in effect, creates an IC and Chapter 2’s optimum spectrum balancing and approximations thereto become a means to find best energy allocation, albeit at performance loss relative to this section’s more simply attained actual optimum if the situation is indeed a MAC or a BC.

### 5.4.4.3 Distributed Implementations

Figure 5.35 shows a possible MAC implementation where a controller executes the resource (bits and energy) allocation for all $U$ users and passes the resulting information/user and energy/user distributions to each user. A noise change (or an energy/rate change of any of the users), or a channel $H_u$ change, could cause need for the re-execution of Mohseni’s Algorithm. The area of efficient incremental changes remains is a good one for research. Alternatively, Section 4.4’s Separation Theorem and Nested Loading may find use with appropriate gaps or tolerances.

![Central Control for swapping](image)

It is also possible to continue distributed energy/code allocation (and possibly some minor gain swapping within a user’s tones) to accommodate a channel change. Such an implementation might ensure some gap $\Gamma_{mu}$ for the users. Then an initial pass of Mohseni’s algorithm runs with the following constraints:

1. reduce the energy constraints by $\Gamma_{mu}$ for each user if an admission problem
2. increase crosstalk by $\Gamma_{mu}$ for any crosstalk component generated by user $u$, whether admission or min-E-sum problem

3. upon (any) completion increase the individual user energies by $\Gamma_{mu}$ (recalling that the actual crosstalk in the transmission system and not in the algorithm execution is smaller by $\Gamma_{mu}$ for any crosstalk generated by user $u$).

The results of the above single Mohseni Algorithm execution the are initialized to the transceivers (or updated at sufficiently frequent intervals to allow the controller to be able to solve the basic problem for all users). The separation theorem and erasures of lost tonal symbols allows for recovery of minor time-variation. More significant change requires solution of the minimum-energy-sum problem. An incremental solution of that problem for changes remains a research challenge (of course it can be run fully upon changes).
5.5 The Gaussian BC via Duality and GDFE

The vector BC GDFE channel has diagonal receiver processing with the worst-case noise for any specified $R_{xx}$, as in Chapter 2. However for any given admissible rate vector $b$, vector duality instead uses Section 5.4's minPMAC with a GDFE-design series that computes a vector-Gaussian-BC transmit precoder, the linear transmit matrices, and the corresponding independent linear receiver matrices (which precede any receiver modulo operations). This process avoids worst-case noise by exploiting the BC's dual MAC with reversed order, but the MAC's $\mathcal{E}$ is more restrictive than a worst-case noise design that observes only an energy-sum constraint. Thus minPMAC can run for any given $b_0$ and $w = 1$ to check if the resultant $w^* \cdot \mathcal{E}_x \leq \mathcal{E}_x$; if so, $b_0 \in \mathcal{C}(b)$ is admissible, and if not, $b_0$ is an inadmissible rate vector for the energy constant $\mathcal{E}_x$, $b_0 \notin \mathcal{C}(b)$. Any boundary point $b_0 \in \mathcal{S}_{e-sum,mac}(b)$ corresponds to a best BC design and produces $\mathcal{E}_x \equiv \mathcal{E}_x \cdot 1 = \mathcal{E}_x$. The BC-design thus uses minPMAC to design $R_{xx}$ and/or $b$ from a dual vector MAC and applies it to a vector BC for such admissible data rate on the dual energy-sum MAC. The admMAC addresses the MAC's energy-vector constraint, which vector is not part of BC design. Thus, the minPMAC's utility relative to the more direct admMAC becomes evident with the BC.

Subsection 5.5.1 develops single-user vector-channel duality. Subsection 5.5.2 then focuses duality to the case of the MAC and corresponding dual BC. Subsection 5.5.3 then uses Chapter 2's MMSE design series (now of GDFEs) that comprise the BC designs precoder (without need of worst-case noise knowledge). Subsection 5.5.4 revisits Vector DMT for the BC. While Subsection 5.5.5 revisits generation of $\mathcal{C}_{BC}(b)$.

5.5.1 Single-User Dual Channels

A dual channel transposes the channel and reverses dimensional order. Dual channels have the same sum rate and energy sum. Subsection 5.5.2 specializes dual channels that match a multi-user MAC to a multi-user BC. The corresponding GDFEs for the dual channels will have the same bit distributions.

Vector duality generalizes Section 2.8's scalar concept. Subsection 5.5.1.1 introduces duality concepts for any matrix channel. Subsection 5.5.2 then refines duality to determine a special set of autocorrelation matrices that apply to the dual vector BC and MAC. The more general GDFE approach taken here easily shows that the mutual information of dual channels must be the same (so the rate sums are the same) and that the energy sum is also the same.

5.5.1.1 Single-User Dual-Matrix Channels - Any $H$, $R_{nn}$, and $R_{xx}$

Duality begins with a initial matrix AWGN

$$y = H \cdot x + n$$

where $H$ is an $L_y \times L_x$ matrix and so is $\tilde{H} \overset{\Delta}{=} R_{nn}^{1/2} \cdot H$ for the equivalent white-noise channel with $\tilde{H}$.

The input has autocorrelation matrix $R_{xx}$.

**Definition 5.5.1** [The Dual Channel] For (equivalent white-noise) channel matrices $\tilde{H}$, the dual channel is

$$H_{dual} \overset{\Delta}{=} \mathcal{J}_x \cdot H^* \cdot \mathcal{J}_y$$

where $\mathcal{J}_x$ is a permutation matrix that corresponds to reversing the input-dimension order $\pi$, and $\mathcal{J}_y$ does the same for the output dimensions. Usually, these $\mathcal{J}$ matrices have 1’s on the antidiagonal. However, these 1’s may yield to identity matrices of certain block size corresponding to input (or output) dimensional groups (for instance MAC and BC; multiuser dimensional groups corresponds to $L_{x,u} > 1$ and/or $L_{y,u} > 1$), respectively. This text’s $\tilde{H}$ corresponds to a MAC and $\tilde{H}_{dual}$ to a BC; however duality mapping can also be BC to MAC.
The order reversal may appear superfluous above, but it will be notationally convenient for MAC to/from BC duality. The channel has “economy mode” SVD
\[ \tilde{H} = F \cdot \Lambda \cdot M^* \] (5.355)
and so the dual channel has corresponding SVD
\[ \tilde{H}_{\text{dual}} = \mathcal{J}_x \cdot \tilde{H}^* \cdot \mathcal{J}_y = \mathcal{J}_x \cdot M \cdot \Lambda \cdot F^* \cdot \mathcal{J}_y . \] (5.356)
The matrices \( \mathcal{J}_x \cdot M \) and \( \mathcal{J}_y \cdot F \) retain all orthonormal columns, so (5.356) is a valid “economy-mode” SVD for the dual channel. The dual channel input is \( \tilde{x} \), as in Figure 5.36. When \( \tilde{H} \) is not square, the input \( x \) or \( \tilde{x} \) with the smaller dimensionality may decompose into a sum of components equal in number up to the larger dimensionality. The dual-channel input is
\[ \tilde{x} \triangleq J_y \cdot F \cdot M^* \cdot x \] (5.357)
with autocorrelation matrices related as
\[ R_{\tilde{x}\tilde{x}} = J_y \cdot F \cdot M^* \cdot J_x \cdot R_{xx} \cdot J_x \cdot M \cdot F^* \cdot J_y \] (5.358)
If \( x \) is \( U \times 1 \) with \( U > 1 \), then \( \tilde{x} \) consists of \( U \) components \( \tilde{x} = \sum_{u=1}^{U} \tilde{x}_u \). The dual of the dual is the original channel.

**Theorem 5.5.1 [Dual Single-User Channels]** The dual-squared channel \( \tilde{H} \) based on an initial non-square channel \( \tilde{H} \) in Definition 5.5.1’s (5.354) with inputs related by (5.357) have the same:

- **mutual information**
  \[ I(x; y) = I(\tilde{x}; \tilde{y}) \] (5.359)

- **input energy**, and
  \[ \text{trace} \{ R_{xx} \} = \text{trace} \{ R_{\tilde{x}\tilde{x}} \} . \] (5.360)

- **equal SNR determinants** With applicable GDFE
  \[ |S_{0,2}| = |S_{0,1}| . \] (5.361)

**Proof:** The energy preservation and mutual-information invariance follows because the input transformation in (5.357) is both unitary and invertible 1-to-1 mapping, respectively. The mutual information for the original channel with order reversal depends on the determinant
\[ |\tilde{H} \cdot J_x \cdot R_{xx} \cdot J_x \cdot \tilde{H}^* + I| = |\Lambda \cdot M^* \cdot J_x \cdot R_{xx} \cdot J_x \cdot M \cdot \Lambda + I| \] (5.362)
since \( |F| = 1 \). Similarly the dual channel’s mutual information depends on the determinant
\[ |\tilde{H}_{\text{dual}} \cdot R_{\tilde{x}\tilde{x}} \cdot \tilde{H}^*_{\text{dual}} + I| = |\Lambda \cdot F^* \cdot J_y \cdot R_{\tilde{x}\tilde{x}} \cdot J_y \cdot F \cdot \Lambda + I| . \] (5.363)
These two determinants are equal because when (5.356), or equivalently (5.358) are true. Thus the two GDFE’s have the same SNR’s, equivalently \( S_{0,1} \) and \( S_{0,2} \) have the same determinants and the channel and its dual under the input mapping have the same mutual information. QED.

---

\(^{74}\)Economy-mode SVD retains column vectors of \( F \) and \( M \) that correspond only to non-zero singular values, so \( H = F \cdot \Lambda \cdot M^* \) and \( \Lambda \) is a diagonal matrix of rank \( \varrho_H \). Matlab computes this with command svd(H,’econ’) for matrix H.
Study a dual may sometimes simplify design.

\[
\begin{align*}
\mathbf{v} \leftarrow & \mathbf{R}_{xx}^{1/2} \mathbf{x} \quad \mathbf{n} \quad R_{nn} = I \\
\mathbf{n} \leftarrow & \mathbf{R}_{xx} \\
\mathbf{y} = & \mathbf{H} = \mathbf{F} \cdot \mathbf{A} \cdot \mathbf{M}^* \\
\mathbf{w}_{\text{comb}} = & (\mathbf{S}_0 - I)^{-1} \cdot \mathbf{G}^* \cdot \mathbf{H}^* \\
\mathbf{dec} = & I - \mathbf{G}_1
\end{align*}
\]

**Dual channels**

\[
\begin{align*}
\bar{\mathbf{v}} \leftarrow & \mathbf{R}_{\bar{xx}}^{1/2} \mathbf{\bar{x}} \quad \mathbf{n} \quad R_{\bar{nn}} = I \\
\mathbf{n} \leftarrow & \mathbf{R}_{\bar{xx}} \\
\bar{\mathbf{y}} = & \mathbf{\bar{H}} = \mathbf{J}_y \cdot \mathbf{M}^* \cdot \mathbf{J}_x \cdot \mathbf{R}_{xx} \\
\mathbf{w}_{\text{comb}} = & (\mathbf{S}_0 - I)^{-1} \cdot \mathbf{G}^* \cdot \mathbf{J}_y \cdot \mathbf{\bar{H}} \\
\mathbf{dec} = & I - \mathbf{G}_2
\end{align*}
\]

When \( \text{rank}(\mathbf{R}_{xx}) = \rho_x > \rho_{\bar{H}} \), the input \( \mathbf{x} \) has autocorrelation matrix \( \mathbf{R}_{xx} \) that necessarily contains nonzero energy in the channel null space \( \mathcal{N}_{\bar{H}} \). There will be thus rate loss; since both duals have the same \( \mathcal{I} \), then both exhibit this same loss. Thus \( \mathbf{R}_{\bar{xx}} \) will also have nonzero energy in the dual-channel’s null space \( \mathcal{N}_{\bar{H}} \). The input energies of both the dual channel and the original channel satisfy

\[
\mathcal{E}_x = \mathcal{E}_{\bar{x}} = \mathcal{E}_x(\mathcal{P}_{\bar{H}}) + \mathcal{E}_x(\mathcal{N}_{\bar{H}}) = \mathcal{E}_x(\mathcal{P}_{\bar{H}_{\text{dual}}}) + \mathcal{E}_x(\mathcal{N}_{\bar{H}_{\text{dual}}}) .
\]

(5.364)

If \( \rho_x \leq \rho_{\bar{H}} \), the channel has sufficient dimensionality to avoid energy loss to the null space, although this requires proper input construction (which may not be possible, for instance with a MAC) so it is still possible for energy to enter the channel null space.

**Figure 5.37:** GDFE for subset of dimensions indexed by \( u \).

Figure 5.37 illustrates a GDFE for user \( u \)’s subset of the dual channel’s row dimensions, \( \tilde{\mathbf{H}}_u \). Figure 5.37’s receiver linear processing is variables separable when estimating \( \mathbf{v} \). Thus, it is the MMSE solution.
for those channel outputs that it processes\textsuperscript{75}, denoted $y_u$. The input $x$ depends on more dimensions and not on user index $u$. Figure 5.37’s receiver estimates all input dimensions. As before, a GDFE with lossless precoder essentially moves the feedback-section to the transmitter and retains the MMSE performance.

\subsection*{5.5.2 Specializing to MAC-BC Duality}

A vector MAC $\bar{H}$ with rank $\varrho_{\bar{H}}$ is

$$\bar{H}_{\text{MAC}} = \begin{bmatrix} \bar{H}_U & \ldots & \bar{H}_1 \end{bmatrix}$$

(5.365)

that is $\mathcal{L}_y \times \mathcal{L}_x$, and has a dual $\mathcal{L}_x \times \mathcal{L}_y$ matrix $\bar{H}_{\text{BC}} \overset{\Delta}{=} \bar{H}_{\text{dual}} = c J_x \cdot \bar{H}^* \cdot J_y$ that corresponds to dual BC vector

$$\bar{H}_{\text{BC}} = J_x \cdot \bar{H}^* \cdot J_y = J_x \cdot \begin{bmatrix} \bar{H}_U^* \\ \vdots \\ \bar{H}_1^* \end{bmatrix} \cdot J_y .$$

(5.366)

$\bar{H}_{\text{BC}}$’s input sets may need view as a sum of $U$ components.

In addition to the overall duality (viewing $H$ and $\bar{H}_{\text{dual}}$ as single-user channels), each vector BC user has dual $H_u^* \cdot J_y$ that corresponds to same dual-MAC user $\bar{H}_u$ with the same mutual information as in Figure 5.37’s specific receiver processing of only that user’s BC output dimensions. The BC’s order-reversal convention causes the best BC user position to correspond to the MAC user that is in the worst position; this is desired. This can reduce the energy-sum MAC’s rate sum for degraded MAC’s with $U^* > 0$. Indeed the dual channel’s reversed order can reduce the corresponding dual energy-sum-MAC/GDFE sum-rate because the BC’s energization of its primary users does NOT correspond to the energy-sum dual vector MAC’s energization of its primary users. Indeed the worst $U^*$ MAC users may correspond to the best $U^*$ BC-dual (primary) users (when $U^* > 0$). Any MAC order has a dual, and may result in $\{\bar{H}_{\text{dual},u}\}$ that has lower sum rate for the given $R_{\text{x},\text{x}}$ than the best when there is only an energy-sum constraint and a suboptimal order. Correspondingly, a BC maximum rate sum can never exceed it’s dual energy-sum MAC’s maximum rate sum, and is less (along with the dual of the energy-sum MAC) if $H$ was degraded and there are any energized BC secondary users. The BC rate sum will be the dual energy-sum (degraded) MACs rate sum. This max rate-sum point can be calculated with Matlab program bcmax.m because both maximum rate sums are the same and from (??):

$$\sum_{u=1}^{U} E_u^B = \sum_{u=1}^{U} E_u^M .$$

(5.367)

and the user bits/subsymbol also match in the duals, but a final order-reversal step is necessary to complete the mapping between a MAC and a BC.

While duality can create channels with the same sum-user data rates, a MAC requires independent inputs and a diagonal $R_{\text{x},\text{x}}$. Not every $R_{\text{x},\text{x}}^{1/2}$ choice leads to such a diagonal dual input. There is however at least one square root that does render both $R_{\text{x},\text{x}}$ diagonal for any input BC $R_{\text{x},\text{x}}$. A first step in generating that special square root is input deflection:

**Input Deflection:** Input deflection applies to Figure 5.38’s individual-user channel $\bar{H}_u$ in a MAC/BC dual set. While the channel noise $n$ is white ($R_{nn} = I$), the individual-user noises include crosstalk from other users, namely as

$$R_{nn}(u) = I + \sum_{i=u+1}^{U} \bar{H}_i \cdot R_{\text{x},\text{x}}(i) \cdot \bar{H}_i^*$$

(5.368)

$$R_{\bar{n}\bar{n}}(u) = I + \bar{H}_u^* \cdot \left[ \sum_{i=1}^{u-1} R_{\text{x},\text{x}}(i) \right] \cdot \bar{H}_u .$$

(5.369)

\textsuperscript{75}A better solution occurs when the receiver processes all dimensions; but only $y_u$ is available here. Figure 5.37 is a MMSE estimator for $y_u$ from $v$. 

939
Thus, the individual-users’ mutual information expressions for any given $R_{xx}(u)$ and $R_{x\bar{x}}(u)$ can equate through

\[
\begin{align*}
2\mathcal{I}_u &= \frac{\left| \bar{H}_u \cdot R_{xx}(u) \cdot \bar{H}_u^* + \mathcal{R}_{nn}(u) \right|}{|\mathcal{R}_{nn}(u)|} \quad (5.370) \\
&= \frac{\left| \bar{H}_u^* \cdot R_{xx}(u) \cdot \bar{H}_u + \mathcal{R}_{\bar{n}\bar{n}}(u) \right|}{|\mathcal{R}_{\bar{n}\bar{n}}(u)|} \quad (5.371) \\
&\triangleq \left| \bar{H}_u \cdot R_{xx}(u) \cdot \bar{H}_u + I \right| \quad (5.372) \\
&\triangleq \left| \bar{H}_u^* \cdot R_{xx}(u) \cdot \bar{H}_u + I \right| \quad (5.373)
\end{align*}
\]

where $\bar{H}_u$, $\bar{x}_u$, and $\bar{\bar{x}}_u$ have definition in the succeeding Lemma 5.5.1, and the last two expressions equivalently diagonalize the noise (including crosstalk) autocorrelation matrix with this **deflected-input channel**. The input-deflection lemma is general, but applies here in the context of the deflecting matrices being the square roots of a $\mathcal{R}_{nn}(u)$ and $\mathcal{R}_{\bar{n}\bar{n}}(u)$ for the vector MAC and BC dual set.

**Lemma 5.5.1 [Input Deflection with Correlated Crosstalk Noise]** For two user-specific channels $\bar{H}$ and $\bar{H}^*$ with different noise autocorrelation matrices, as in Figure 5.38, here more specifically as $\mathcal{R}_{nn}$ and $\mathcal{R}_{\bar{n}\bar{n}}$, and the same mutual information, the related dual channels

\[
\hat{H}_u \triangleq \mathcal{R}_{nn}^{-1/2}(u) \cdot \bar{H}_u \cdot \mathcal{R}_{\bar{n}\bar{n}}^{-1/2}(u) , \quad (5.374)
\]

and $\hat{H}^*$, have the same mutual information with new-noise autocorrelation matrices $I$ and deflected inputs

\[
\begin{align*}
\hat{x}_u &= \mathcal{R}_{\bar{n}\bar{n}}^{1/2}(u) \cdot x_u \quad (5.375) \\
\hat{\bar{x}}_u &= \mathcal{R}_{nn}^{1/2}(u) \cdot \bar{x}_u \quad (5.376)
\end{align*}
\]

Furthermore, from the input deflections,

\[
R_{x\bar{x}}(u) = \mathcal{R}_{nn}^{1/2}(u) \cdot R_{xx}(u) \cdot \mathcal{R}_{nn}^{1/2}(u) , \quad (5.377)
\]

and equivalently

\[
R_{x\bar{x}}(u) = \mathcal{R}_{nn}^{1/2}(u) \cdot R_{\bar{x}\bar{x}}(u) \cdot \mathcal{R}_{nn}^{1/2}(u) . \quad (5.378)
\]
Proof: Both noises are full rank because of the identity matrices in (5.368) and (5.369), and thus they are invertible 1-to-1 mappings and do not change mutual information, whether applied to input or output. Both matrices cause the noise-plus-crosstalk to be white when applied at output, so the mutual information value is then the same (denominator-determinant and numerator-determinant’s added term are identities so the noise also has the same autocorrelation in both). QED.

Figure 5.39 illustrates the fully whitened dual-user channels.

The input deflections can however change the transmitted energy, and indeed duality focuses on the energy sum for all users’ matrix channels, not on equality of individual users’ corresponding energies. Thus, the dual GDFEs’ $G$ matrices deviate, but not $S_0$. The transformations in (5.375) and (5.376) do not introduce correlation between users if applied each separately for a $\mathbf{R}_{nn}(u)$ and/or $\mathbf{R}_{\bar{n}\bar{n}}(u)$, and so these mappings preserve user independence. The remaining duality step sets the user energy sums equal. However, first a scalar-duality reminder follows.

Scalar Duality Revisted By returning to the scalar duality in Subsection 2.8.4, the energy mappings from BC to/from MAC are again:

$$\epsilon_{BC}^1 = \epsilon_{MAC}^1 \cdot \frac{\mathbf{1} + \epsilon_{MAC}^2 \cdot g_2 + \ldots + \epsilon_{MAC}^U \cdot g_U}{\mathbf{1} + \epsilon_{MAC}^1 \cdot g_2 + \ldots + \epsilon_{MAC}^U \cdot g_U}$$  \hspace{1cm} (5.379)

$$\epsilon_{BC}^2 = \epsilon_{MAC}^2 \cdot \frac{\mathbf{1} + \epsilon_{MAC}^1 \cdot g_2 + \ldots + \epsilon_{MAC}^U \cdot g_U}{\mathbf{1} + \epsilon_{MAC}^1 \cdot g_2 + \ldots + \epsilon_{MAC}^U \cdot g_U}$$  \hspace{1cm} (5.380)

$$\vdots \quad \vdots$$  \hspace{2cm} (5.381)

$$\epsilon_{BC}^U = \epsilon_{MAC}^U \cdot \left(1 + \left[\epsilon_{BC}^1 + \ldots + \epsilon_{BC}^{U-1}\right] \cdot g_U\right)$$  \hspace{1cm} (5.382)

(5.383)
These equations are equivalent to writing the deflected inputs as

\[
x^{BC}_1 \cdot \sqrt{1 + \mathcal{E}^{MAC}_2 \cdot g_2 + \ldots + \mathcal{E}^{MAC}_U \cdot g_U} = x^{MAC}_1 \\
x^{BC}_2 \cdot \sqrt{1 + \mathcal{E}^{MAC}_3 \cdot g_3 + \ldots + \mathcal{E}^{MAC}_U \cdot g_U} = x^{MAC}_2 \cdot \sqrt{1 + \mathcal{E}^{BC}_1 \cdot g_2}
\]

(5.384)

(5.385)

(5.386)

These indeed are the input deflection mappings of Lemma 5.5.1, but simpler versions for the scalar case, which can be viewed on either side as noise whitening with corresponding input deflection for any \( u \). The square-root terms include the noise plus the other – uncancelled/precoded – user effects in noise whitening the single scalar output or in deflecting any particular scalar input. The scalar energies sum to the same total on each of the scalar duals. The vector case requires more tedious bookkeeping to preserve the specified \( R_{xx} \) or \( R_{x\bar{x}} \) and the corresponding energy sum. Prior to duality’s vector generalization, an example is revisited to help illustrate scalar duality.

**EXAMPLE 5.5.1 [Revisit of Chapter 2’s scalar BC Example]** Example 5.4.1’s (noise-normalized/whitened) scalar channel is \( H_{MAC} = [80 \ 50] \), but this BC example will start with the dual

\[
H_{BC} = \begin{bmatrix} 50 \\ 80 \end{bmatrix},
\]

(5.388)

for which the dual is \( H_{BC} = J_y \cdot H_{MAC}^* \). User 2 for both MAC and BC corresponds to the 80 (the primary user). The maximum rate sum is 6.322 bits/sub-symbol with all energy \( \mathcal{E}_2 = 1 \) on user 2, so \( \mathcal{E}_1 = 0 \) for user 1 as the secondary user. If the initial BC energies are instead

\[
\mathcal{E}^{BC} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \rightarrow \begin{bmatrix} 1/2502 \\ 2501/2502 \end{bmatrix} = \mathcal{E}^{MAC},
\]

(5.389)

then the dual-channel’s MMSE MAC design follows:

\[
>> H=[50 \ 80];
>> Rxx=diag([1/2502 2501/2502]);
>> A=sqrtm(Rxx) =
\]

\[
0.0200 \quad 0 \\
0 \quad 0.9998
\]

\[
>> [Bu, GU, WU, S0, MSWMFU] = mu_mac(H, A, [1 1], 2)
Bu =
\]

\[
1.0000 \quad 80.0160 \\
0 \quad 1.0000
\]

\[
GU =
\]

\[
1.0008 \quad 0 \\
-0.0125 \quad 0.0003
\]

\[
S0 = \begin{bmatrix} 1.0e+03 * \\ 0.0020 \quad 0 \\ 0 \quad 3.2010 \end{bmatrix}
\]

\[
MSWMFU =
\]

\[
1.0004 \\
0.0125
\]

\[
>> MSWMFU*H*A =
\]

\[
1.0000 \quad 80.0160 \\
0.0125 \quad 1.0000
\]

\[
>> \text{sum}(Bu) = \quad 6.3219
\]

942
The given BC's given energy vector through duality places large energy on MMSE MAC user 2 so thus the rate sum equals very close to the maximum of 6.322 bits/subsymbol. This rate sum decomposes to $b = [0.5 5.8]^*$, equal to Section 2.8's direct calculation for this same $E^{BC} = [0.5 0.5]^*$. The unbiased feedback and biased feedback are the same, because the MAC channel output is a scalar. The two dual-MAC users have the same data rates as the BC with $E^{BC} = [0.5 0.5]^*$, and thus $E^{MAC} = [1/2502 2501/2502]$.

Section 2.8's later worst-case-noise design for this same BC instead only finds the rate sum, because that design's user 1 is a "free-loading" secondary user for which the feedback coefficient's product with the corresponding input is always the square-root energy fraction. If $E_1 = 1$, then both the index-reversed energy-sum MAC and the BC obtain 6.322 by placing all the energy on the better-gain user channel. The

Instead using equal MAC-user energies and duality, scalar duality would find:

$$E^{MAC} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \rightarrow \begin{bmatrix} 2501/2502 \\ 1/2502 \end{bmatrix} = E^{BC}. \tag{5.390}$$

Then, the MAC has

```
>> H=[50 80];
>> A=sqrt(1/2)*eye(2);
>> [Bu, GU, WU, S0, MSWMFU] = mu_mac(H, A, [1 1] , 2)
```

$$Bu = \begin{bmatrix} 5.1444 \\ 0.9155 \end{bmatrix}$$

$$GU = \begin{bmatrix} 1.0000 & 1.6000 \\ 0 & 1.0000 \end{bmatrix}$$

$$WU = \begin{bmatrix} 0.0008 \\ -0.6250 \end{bmatrix} \begin{bmatrix} 0 \\ 0.3909 \end{bmatrix}$$

$$S0 = \begin{bmatrix} 1.0e+03 * \\ 1.2510 \\ 0 \end{bmatrix}$$

$$S0 = \begin{bmatrix} 1.0e+03 * \\ 1.2510 \\ 0 \end{bmatrix}$$

$$0.0036$$

$$MSWMFU = \begin{bmatrix} 0.0283 \\ 0.0177 \end{bmatrix}$$

$$>> MSWMFU*H*A = \begin{bmatrix} 1.0000 & 1.6000 \\ 0.6250 & 1.0000 \end{bmatrix}$$

$$>> sum(Bu) = 6.0600$$

The MAC user 2's smaller energy (0.5 < 2501/2502) reduces the rate sum to 6.06 (below 6.322) because the MAC secondary user has most energy. While the MMSE MAC and GDFE rate sums for any given $R_{xx}$ are order-independent, these sums can vary with $R_{xx}$. This design has less rate sum than the maximum because of the MAC's $R_{xx}$ choice. This rate vector is not a point found in Section 2.8, but still on the $S_{BC}(b)$ border, and as well on the identical $S_{BC}^{-MAC}(b) = S_{BC}(b)$ border.

The feedback section for the original design has coefficient 80 for the receiver’s “subtraction/removal” of user 2 from user 1, which with the order reversal is indeed the same for both the BC and the dual MAC. BC transmitter general design determination from the dual MAC appears later in Subsection 5.5.3.

The designer uses duality to find a dual-MAC’s $E$ that corresponds to a specified rate vector $b$ (and consequent minimized weighted energy sum) or for the specified energy vector and a maximized weighted rate sum. If the point is admissible on the dual MAC, it is also admissible on the BC. The rate sum for a fixed set of $R_{xx}(u)$ may be less on the MAC.
Example 5.5.1’s key point is that input deflection does not change the bit rate \( I(x; y) \) nor the individual user’s rates, although the corresponding energy distributions are different for the BC and its dual. The BC’s maximum rate-sum corresponds to the dual MAC’s order-reversed maximum rate sum (for an energy-sum MAC).

The lack of coordination at the MAC transmitters restricts the input autocorrelation matrices so that reduction in data rate relative to single user is consequent, while the order reversal tacitly imposes worst-case noise rate-sum reduction. The BC loss parameter measures this loss.

\[
\gamma_{BC} = \frac{2^2 - 1}{2^2 - 1} \leq \gamma_{e-sum,dual-MAC} = \gamma_{single-user} = 1.
\]

(5.391)

Lemma 5.5.2 [BC loss relative to single user]

Proof: Duality assures this result. The energy-sum MAC will use best MAC for its maximum rate-sum calculation, which may not correspond to the BC best order. QED

This concept of input deflection and the dual channel applies not to the overall channel \( \tilde{H} \) (where the overall duality preserved sum rate and total sum energy), but instead to the individual user data rates and channel elements in dual vector BC and vector MAC channels.

Duality Mappings: Figure 5.40 illustrates a general vector dual BC and its original vector MAC at user level. The \( \bar{H}_u \) are the MAC columns and the \( \bar{H}_u^* \cdot J_y \) are the BC (order-reversed) rows. Theorem 5.5.1’s translations apply directly to the overall \( \tilde{H} \) channel (both overall noises become white) so that the individual users’ rates and overall rate sum of both channels are equal. However, individual \( R_{xx} \) and \( R_{\bar{x}\bar{x}} \) are now of interest.

The individual user and input autocorrelation matrices \( R_{xx}(u) \) and \( R_{\bar{x}\bar{x}}(u) \) can be chosen so that the data rates on each of the two dual channels are equal as in (5.370) - (5.373). These two individual user channels correspond to the data rates in (5.370) and look almost like duals already, except they have different noises. As in Figure 5.39, the dual design accommodates different noises by input deflection. Each individual channel description deflects its input with equivalent noises that include crosstalk, which is not necessarily white, to equate user \( u \)’s individual MAC and BC data rates. That is, each equivalent-user subchannel deflects their input by the corresponding dual’s square-root noise autocorrelation, where that noise autocorrelation contains the other users’ noises, as in Equations (5.377) and (5.378).
Such input deflection does not change the individual user’s mutual information since it is 1-to-1, so the user’s rate on both dual and original is the same, \( \forall u \). Returning to (5.372) and (5.373) with input deflection, noting trivially that \( \mathcal{R}_{nn}^{1/2} \cdot \mathcal{R}_{nn}^{-1/2} = I \), the equal individual-user mutual information follows if

\[
\left| \hat{H}_u \cdot \mathcal{R}_{nn}^{1/2}(u) \cdot R_{xx}(u) \cdot \mathcal{R}_{nn}^{1/2}(u) \cdot \hat{H}_u^* + I \right| = \left| \hat{H}_u \cdot \mathcal{R}_{nn}^{1/2}(u) \cdot R_{\bar{x}\bar{x}}(u) \cdot \mathcal{R}_{nn}^{1/2}(u) \cdot \hat{H}_u + I \right| , \tag{5.392}
\]

which uses the input deflection mappings from (5.375) and (5.376), to undo the input deflection in the channel model \( \hat{H}_u \). This deflection also absorbs the channel’s other-side noise normalization, thereby setting the denominator determinant to unity.

The (again “economy-mode” SVD) of \( \hat{H}_u \) is

\[
\hat{H}_u = F_u \cdot \Lambda_u \cdot M_u^* . \tag{5.393}
\]

Equation (5.392) requires appropriate choice of \( R_{xx} \) and \( R_{\bar{x}\bar{x}} \). The first determinant’s argument is

\[
F_u \cdot \Lambda_u \cdot M_u^* \cdot \mathcal{R}_{nn}^{1/2} \cdot R_{xx}(u) \cdot \mathcal{R}_{nn}^{1/2} \cdot M_u \cdot \Lambda_u \cdot F_u^* + I . \tag{5.394}
\]

The pre-multiplication and post-multiplication by \( F_u^* \) and \( F_u \) respectively does not change the determinant, nor user \( u \)’s mutual information. Then, if the two input autocorrelation matrices \( R_{xx} \) and \( R_{\bar{x}\bar{x}} \) satisfy

\[
R_{\bar{x}\bar{x}}(u) = M_u \cdot F_u^* \cdot R_{xx}(u) \cdot F_u \cdot M_u^* . \tag{5.395}
\]

Insertion of (5.395) into (5.392), using also (5.368), equates the individual-user mutual information between vector MAC and vector BC. The autocorrelation matrices relate as (BC to MAC)

\[
R_{xx}(u) = \mathcal{R}_{nn}^{-1/2}(u) \cdot M_u \cdot F_u^* \cdot \mathcal{R}_{nn}^{1/2}(u) \cdot R_{\bar{x}\bar{x}}(u) \cdot \mathcal{R}_{nn}^{1/2}(u) \cdot F_u \cdot M_u^* \cdot \mathcal{R}_{nn}^{-1/2}(u) , \tag{5.396}
\]

Figure 5.40: Dual Vector BC and MAC channels.
which reverses to the MAC-to-BC relationship

\[
R_{xx}(u) = R_{nn}(u) \cdot F_u \cdot M_u^* \cdot R_{nn}^{-1/2}(u) \cdot R_{xx}(u) \cdot R_{nn}^{1/2}(u) \cdot M_u \cdot F_u^* \cdot R_{nn}^{-1/2}(u) \ .
\]  

(5.397)

The relationships hold for each and every \( u \). Recursive use of Equation 5.396 must start with user \( U \) and successively pass to \( U-1, \ldots, 1 \) because each successive \( R_{nn}(u) \) in (5.368) depends on higher indices of \( R_{xx}(u) \). Equation (5.397) must start with user 1 and pass successively to 2, \ldots, \( U \) because each successive \( R_{xx}(u) \) in (5.369) depends on lower indices of \( R_{nn}(u) \).

Given:

\[ R_{xx}(u) \text{ for } u = 1, \ldots, U \text{ & } R_{xx}(u) = R_{xx} = 0 \]
\[ R_{nn}(U) = R_{nn}^{-1}(1) = 1 \]
\[ \text{BC: } H_u = R_{nn}(u) \]
\[ \text{MAC: } \bar{H}_u = H_u \cdot R_{nn}^{-1/2}(u) \]

For \( u = 1, \ldots, U-1 \):

\[ R_{nn}(u + 1) = R_{nn}(u) + \bar{H}_u \cdot R_{xx}(u) \cdot \bar{H}_u \]

For \( u = U, \ldots, 1 \)

\[ \bar{H}_u = R_{nn}^{-1/2}(u) \cdot \bar{H}_u \cdot R_{nn}^{-1/2}(u) \]
\[ F_u \cdot \Lambda_u \cdot M_u^* = \text{svd} \left( \bar{H}_u \right) \]
\[ R_{xx}(u) = R_{nn}^{-1/2}(u) \cdot M_u \cdot F_u^* \cdot R_{nn}^{-1/2}(u) \cdot R_{xx}(u) \cdot R_{nn}^{-1/2}(u) \cdot F_u \cdot M_u^* \cdot R_{nn}^{-1/2}(u) \]
\[ R_{nn}(u-1) = R_{nn}(u) + \bar{H}_u \cdot R_{xx}(u) \cdot \bar{H}_u^* \; ; \; \text{skip } u = 1 \]

Figure 5.41: Duality flow chart for BC to MAC.

The dual relationships correspond to a recursion that Figures 5.41’s and 5.42’s flow charts illustrate. For MAC to BC, \( R_{xx}(u) \) forms recursively after each successive duality step for the next user. The overall channels (viewed from single-user perspective) are still duals and have the same rate sum. The energies of the individual users are not the same because of the input deflection that makes individual vector BC and vector MAC users’ bit rates equal, or equivalently these individual users’ energies were never the same in the original overall channel, just their sums were equal. While individual users and rate sums are equal, duals need not otherwise be optimum in any sense, unless the specific rate \( b_0 \) is on the capacity region’s border \( S(b) \)
Given:}\[ R_{xx}(u) \text{ for } u = 1, \ldots, U; \quad R_{xx}(u) = R_{xx} = 0 \]
\[ R_{nn}(U) = R_{nn}(1) = I \]
BC: \[ \tilde{H}_u, R_{nn}(u) \]
MAC: \[ \tilde{H}_u = H_u \cdot R_{nn}^{-1/2}(u) \]

For \( u = U, \ldots, 2; \quad R_{nn}(u - 1) = R_{nn}(u) + \tilde{H}_u \cdot R_{xx}(u) \cdot \tilde{H}_u^* \)

**mac2bc Program:** The mac2bc program computes dual autocorrelation matrices. The 2 inputs are:

1. The input \( R_{xx}(u) \) is an \( L_x \times L_x \times U \) array of autocorrelation-matrices. The \( L_x \) is for the MAC.

2. The input \( H_{vec}(\{ \tilde{H}_u \}) \) is the set of \( L_y \times L_x \) MAC user submatrices specified in an \( L_y \times L_x \times U \) array.

The output \( R_{xx}(u) \) is the \( L_y \times L_y \times U \) array of BC autocorrelation component matrices \( \{ R_{xx}(u) \} \). These may be added over the users to produce the aggregate vector BC transmit autocorrelation \( R_{xx}(u) \).

```
>> help mac2bc
function Rxxb = mac2bc(Rxxm, Hvecu)
```

mac2bc converts Gaussian MAC autocorrelation matrices to the corresponding autocorrelation matrices of its dual Gaussian BC. These autocorrelation matrices achieve the same user rates for a GDFE MAC and a GDFE noiseless precoder. The BC encoding order reverses the MAC decoding order in the input order pi.

\( U \) is the number of users; \( L_x \) (per-user) and \( L_y \) are the number of MAC transmit and receive antennas respectively. These are extracted/inferred from the dimensions of inputs \( H_{vec} \) and \( R_{xx} \). The MAC user order is reversed.

Inputs
Rxxm is an Lx by Lx by U matrix containing the MAC input Rxx matrices.
where Rxxm(:,:,u) is the covariance matrix for MAC user u.
Hvecu is an Ly by Lx by U matrix that contains all MAC user-channel matrices.
Hvecu(:,:,u) is user u’s channel matrix, the transpose of Hvecutrans
in bc2mac prog.

Output

Rxxb is an Ly x Ly x U array of dual-BC autocorrelation matrices.
Rxxb(:,:,u) is user u’s autocorrelation matrix.

When there are unequal numbers of dimensions per MAC-user input, so Lx --> Lxu, the input autocorrelation matrices Rxxm should all use the maximum size Lxmax and thus place zeros in any rows/columns necessary on some users to get to Lxmax. In this situation, the correspond H_u should be extended with zeroed columns in the corresponding zeroed-input Rxxm(u) dimensions.

EXAMPLE 5.5.2 (Return to simple BC channel) Chapter 2’s scalar BC had $\tilde{H}_{bc} = [80 \ 50]^*$, so with order of $u = 1$ at the top/left, then $\tilde{H}_{bc,1} = 80$ and $\tilde{H}_{bc,2} = 50$. The MAC has user $u = 2$ at the top left. Thus, $\tilde{H}_{MAC} = [50 \ 80]$ so user 2 has gain 50 on both the BC and its dual MAC in this labelling convention. Thus BC has one unit of input energy, $E_x = 1$. When the two MAC inputs have equal energy, the data rates were

$$b_{2,MAC} = \frac{1}{2} \cdot \log_2 \left(1 + 80^2 \cdot \frac{1}{2}\right) = 5.8222$$  \hspace{1cm} (5.398)

$$b_{1,MAC} = \frac{1}{2} \cdot \log_2 \left(1 + 80^2 \cdot \frac{1/2}{1 + 50^2 \cdot 1/2}\right) = 0.2378$$  \hspace{1cm} (5.399)

$$b_{sum,MAC} = b_{2,MAC} + b_{1,MAC} = 6.0600 = 0.5 \cdot \log_2 |R_{yy}|$$  \hspace{1cm} (5.400)

Using the mac2bc.m program

Rxxm = zeros(1,1,2);
Rxxm(1,1,2)=0.5 ;
Rxxm(1,1,1)=0.5;
Hmac=zeros(1,1,2);
Hmac(1,1,2)=50;
Hmac(1,1,1)=80;
Rxxb = mac2bc(Rxxm,Hmac)
Rxxb(:,:,1) = 1.5620e-04 \% 1/6402
Rxxb(:,:,2) = 0.9998 \% 6401/6402

Correspondingly the two BC rates are then

$$b_{1,BC} = \frac{1}{2} \cdot \log_2 \left(1 + 50^2/6402\right) = 0.2378$$  \hspace{1cm} (5.401)

$$b_{2,BC} = \frac{1}{2} \cdot \log_2 \left(1 + 80^2 \cdot (6401/6402)/(1 + 80^2/6402)\right) = 5.822$$  \hspace{1cm} (5.402)

$$b_{sum,BC} = b_{2,BC} + b_{1,BC} = 6.0600 = 0.5 \cdot \log_2 |R_{yy}|$$  \hspace{1cm} (5.403)

OR for energy more weighted towards user 2:

Rxxm(1,1,2)=6401/6402 ;
Rxxm(1,1,1)=1/6402;
This second duality example produces a lower sum data rate because its secondary BC user \((u = 1\) in this example) has greater energy \(1/2 > 1/2502\) than the first example. The maximum sum rate occurs when user \(u = 2\) has \(E_2 = 1\) and is \(\log_2(1 + 80^2) = 6.3220\). Energization of the secondary user \(u = 1\) reduces the sum rate in both instances of the example, but for this channel the sum-data rate decreases more rapidly with increase of the BC’s secondary-user energy.

Reversal of the order on the original MAC (which means reversal also of its dual BC’s order) produces a similar sequence but with

\[
R_{xx} = \text{zeros}(1,1,2);
R_{xx}(1,1,2)=0.5;
R_{xx}(1,1,1)=0.5;
H_{mac} = \text{zeros}(1,1,2);
H_{mac}(1,1,2)=80;
H_{mac}(1,1,1)=50;
R_{xb} = \text{mac2bc}(R_{xx},H_{mac})
R_{xb}(:,:,1) = 3.9968e-04 \ % 1/2502
R_{xb}(:,:,2) = 0.9996 \ % 2501/2502
b_1=0.5*\log_2(1+80^2/2501) = 0.9157
b_2=0.5*\log_2(1+50^2*(2501/2502)/(1+50^2/2502)) = 5.1444
b_1+b_2=6.0600
\]

and

\[
R_{xx}(1,1,2)=2501/2502;
R_{xx}(1,1,1)=1/2502;
R_{xb} = \text{mac2bc}(R_{xx},H_{mac})
R_{xb}(:,:,1) = 0.5000
R_{xb}(:,:,2) = 0.5000
b_1=0.5*\log_2(1+80^2/2) = 5.8222
b_2=0.5*\log_2(1+50^2*(1/2)/(1+50^2/2)) = 0.4997
b_1+b_2 = 6.3219
\]

which data rates are the same as in Chapter 2’s investigation of this channel for this order. In this case because the primary user goes into the best order position, the sum rate is higher for the second example basically because the influence of the 80 and 50 factors are reversed and also the energy choices place almost all the energy on the primary user with \(E_2 = 2501/2502\).

**bc2mac Program:** The bc2mac program reversely computes the dual MAC’s autocorrelation matrix set \(\{R_{xx}(u)\}\) from a broadcast channel \(\hat{H}_{dual}\) and the BC autocorrelation matrix set \(\{\bar{R}_{xx}(u)\}\).

```matlab
>> help bc2mac
function Rxxm = bc2mac(Rxxb, Hvecutrans)
This function converts Gaussian BC autocorrelation matrices to the corresponding autocorrelation matrices of a dual Gaussian MAC. These
autocorrelation matrices achieve the same user rates for a GDFE MAC and a GDFE (series thereof) BC noiseless precoder. The MAC decoding order reverses the BC encoding order in the input order pi.

U is the number of users; Lx and Ly (per user) are BC’s respective numbers of transmit/receive dimensions (antennas). U, Lx, and Ly are extracted/inferred from the bc2mac function’s inputs. Lx and Ly correspond to this (dual) MAC. The BC user order is reversed during the course of this program, and then the Rxxm output is reversed also.

Inputs

Rxxb is an Ly by Ly by U array containing the BC input Rxx(u) matrices. where Rxxb(:,:,u) is the autocorrelation matrix for MAC user u. The overall autocorrelation matrix Rxxbsum is sum(Rxxb,3), but not output.

Hvecutrans is an Lx by Ly by U BC channel matrix that contains all BC user-channel matrices. Hvecutrans(:,:,u) is user u's noise-whitened equivalent channel matrix. This bc2mac’s Hvecutrans(:,:,u) is the mac2bc’s Hvecu(:,:,u)' for u=1,...,U.

pi is the BC encoding order, a U x 1 vector. pi(u) is the user that is the u’th encoded up from bottom.

Output

Rxxm contains the MAC autocorrelation matrices, which are Ly x Ly x U. Rxxm(:,:,u) is user u’s autocorrelation matrix.

When there are unequal dimensions per BC-user output, so Ly --> Lyu, the output autocorrelation matrices Rxxm will all be the maximum size Lymax. This computed Lymax value is Ly/U from the H input's dimensions and must be an integer, or a warning message is produced and output is 0. In this case, program inputs should place zeros in Hu’s of those users with Lyu<Lymax to force Lymax on all users. Zeros in the corresponding output Rxxm(u) Lymax-Lyu rows/columns can then be deleted.ding output Rxxm(u) Lymax-Lyu rows/columns can then be deleted.

[Reverse the mac2bc in Example 5.5.2] Continuing with the above Rxxb output

Rxxb(:,:,1) = 1/6402;  
Rxxb(:,:,2) = 6401/6402;  
Rxxm= bc2mac(Rxxb, Hbc)  
Rxxm(:,:,1) = 0.5000  
Rxxm(:,:,2) = 0.5000 (checks)

OR

Rxxb(:,:,1) = 1/2;  
Rxxb(:,:,2) = 1/2  
Rxxm= bc2mac(Rxxb, Hbc)  
Rxxm(:,:,1) = 1.5620e-04 % 1/6402  
Rxxm(:,:,2) = 0.9998 % 6402/6402 (checks)

Thus the bc2mac program reverses both the examples for the first order above.
Now progressing to the more sophisticated example with 3 users:

**EXAMPLE 5.5.3 (3-user 64-tone channel)** Continuing with Example 5.4.8 3-user low-pass/bandpass/highpass channel with $N = 64$, the corresponding dual for the equal-energy-input MAC follows as:

\[
N=64; \\
h=\text{cat}(3,[1 0 .8 ; 0 1 1],[.9 -.3 0 ; .5 -1 -1],[0 .2 0 ; .4 -.63 0],[0 0 0 ; 0 .648 0])*10; \\
H = \text{fft}(h, N, 3); \\
Hbc=\text{zeros}(3,2,64); \\
Rxxm=zeros(1,1,3); \\
Rxxm(1,1,:)= 64/67*[1 1 1]; \\
Rxxb=zeros(2,2,3,64); \\
bbc=zeros(3,64); \\
\]

for n=1:N
    Rxxb(:,:,n)=mac2bc(Rxxm, reshape(H(:,:,n),2,1,3));
    Hbc(:,:,n)=H(:,end:-1:1,n)';
    bbc(1,n)=real(log2(1+Hbc(1,:,n)*Rxxb(:,:,1,n)*Hbc(1,:,n)'));
    bbc(2,n)=real(log2((1+Hbc(2,:,n)*(Rxxb(:,:,2,n)+Rxxb(:,:,1,n))*Hbc(2,:,n)')/(1+Hbc(2,:,n)*Rxxb(:,:,1,n)*Hbc(2,:,n)')));
    bbc(3,n)=real(log2((1+Hbc(3,:,n)*(Rxxb(:,:,3,n)+Rxxb(:,:,2,n)+Rxxb(:,:,1,n))*Hbc(3,:,n)')/(1+Hbc(3,:,n)*Rxxb(:,:,2,n)+Rxxb(:,:,1,n)*Hbc(3,:,n)')));
end

bvec=sum(bbc') = 132.7477 412.8794 445.1264
bsum = 990.7535

The BC places user 3 at the right/bottom, corresponding to the MAC’s placement of user 3 at the top/left, thus explaining why the order above appears reversed. User 3 on the MAC is in the best position (with other users’ crosstalk removed), while it is in the worst position (having other users’ crosstalk as noise) in the BC.

5.5.2.1 Determination of $R_{xx}$ for given rate tuple

The dual vector MAC for a vector BC allows ready determination of a rate vector $b$’s feasibility. This rate tuple can be input, along with $\tilde{H}$, and the weight vector $w = [111...1]$ to minPMAC. The resultant output energy should be summed and compared with the total energy constraint. If the total energy constraint is not exceeded, then the point is feasible (in the capacity region) and a GDFE can be designed for the set of user input energies on the dual vector MAC.

5.5.3 Multi-Step BC GDFE Design

As in Subsection 2.8.3, a GDFE, with lossless precoder for the vector BC, forms recursively through a series of $U$ GDFE designs, without worst-case noise computation, but using the $R_{xx}(u)$ set that duality (mac2bc.m) creates from a MAC design for the dual (viewed as $\tilde{H}$ and thus the BC is $\tilde{H}_{\text{dual}}$). A precoded GDFE BC design cannot occur in one step with nonzero secondary-user energy. Indeed, Section 2.8’s precoder only addressed primary users with secondary users appended to the primary-user dimensions. However, this multi-step design finds a single precoder through extraction of user $u$’s rows only from the $u^{th}$ GDFE design’s unbiased feedback matrix $\tilde{G}_u^{\text{unbiased}}$. This section briefly restates some material from Subsection 2.8.3 The BC matrix is

\[
\tilde{H}_{\text{dual}} = \begin{bmatrix}
\tilde{H}_1^* \\
\vdots \\
\tilde{H}_U^*
\end{bmatrix}. 
\]  \hfill (5.407)

Consequently, receiver $u$’s output is:

\[
y_u = \tilde{H}_u^* x + n_u . 
\]  \hfill (5.408)
Each of the MAC input components relates to its white unit-energy/dimension (innovations) data input as
\[ x_u = A_u \cdot v_u, \] (5.409)
where \( A = [A_1, A_2, \ldots, A_U] \) collects all users’ special square root contributions from duality. While a BC has a single \( R_{xx} \), this \( R_{xx} \) is the sum \( R_{xx} = \sum_{u=1}^{U} R_{xx}(u) \). The GDFE approach expands input dimensionality from \( L_x \) to \( U' \cdot L_x \), allowing detection of \( L_x \) components of each user’s input. Separate Gamma = 0 dB codes could be used for each of a user’s components, depending upon the \( \{R_{xx}(u)\} \).

Then
\[ H_u^* = \tilde{H}_u^* \cdot A, \] (5.410)
The MMSE GDFE receiver for \( y_u \) detects all inputs \( i < u \), but only user \( u \) carries message information of external interest at receiver \( u \). The MMSE GDFE precoder processes all inputs as known side information or as Gaussian crosstalk noise, while the receivers \( w_{unb,u} \) in this design processes only \( y_u \) in (5.408). This GDFE-precoder design process places the index of 1 at the top left to correspond to duality’s order reversal. Receiver \( u \)'s GDFE design then forms
\[ R_{b,u}^{-1} = \hat{H}_u \cdot \hat{H}_u^* + I = G_u \cdot S_{0,u} \cdot G_u^* , \] (5.411)
and leads to unbiased feedback section
\[ G_{u\text{unbiased}} = I + S_{0,u} \cdot (S_{0,u} - I)^{-1} \cdot (G_u - I) , \] (5.412)
which is applicable only to user \( u \) so
\[ \hat{G}_u \triangleq \begin{bmatrix} 0 & \ldots & 0 & I & 0 & \ldots & 0 \end{bmatrix} \cdot G_u , \] (5.413)
and
\[ \hat{S}_{0,u} \triangleq \begin{bmatrix} 0 & \ldots & 0 & I & 0 & \ldots & 0 \end{bmatrix} \cdot S_{0,u} \cdot \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ 0 \end{bmatrix} . \] (5.414)
The corresponding (unbiased) MS-WMF feedforward section that includes matched filtering is
\[ \hat{W}_{unb,u} \triangleq \begin{bmatrix} 0 & \ldots & 0 & I & 0 & \ldots & 0 \end{bmatrix} \cdot (S_{0,u} - I)^{-1} \cdot G_u^* \cdot \hat{H}_u . \] (5.415)
Two examples help illustrate:

**EXAMPLE 5.5.4** [Simple vector BC via duality] Design of reversed order MAC for BC user 1:

```matlab
g Hbc = [80, 50];
g Hmac=[50, 80];
g Rxxm =
    3.9968e-04 0
    0 0.9996
>> Rxxb(:,:,1) = 0.5000;
>> Rxxb(:,:,2) = 0.5000;
```
The dual MAC design itself has

The design for BC user 1 (which corresponds to the 80).
>> H1bc=80;
>> A= [ sqrt(0.5) sqrt(0.5)];
>> [snrGDFEu1, G1U, W1U, S10, MSWMFU1] = computeGDFE(H1bc, A, 2)

snrGDFEu1 = 18.9766 dB
G1U =
    1     1
    0     1
W1U =
   0.0003       0
  -1.0000   1.0003
S10 = 1.0e+03 *
   3.2010       0
       0   0.0020
MSWMFU1 =
   0.0177
   0.0177
>> 0.5*log2(diag(S10)) =
   5.8222
   0.4999
>> B1u=0.5*log2(diag(S10))' =
   5.8222   0.4999
>> sum(B1u) = 6.3220
>> MSWMFU1(1,:)*H1bc*A = 1 1

Not surprisingly the overall pre-equalized channel to user 1 simply adds the two inputs (the receiver absorbs the factor of 80). User 1’s receiver intentionally uses \(L_y,1 = 1\) dimension. Only user 1’s data rate of 5.8222 is of interest at receiver 1, although it is possible to decode user 2 at receiver 1 and have a rate sum to receiver 1 of 6.322. The feedback section/coefficient of 1 also is the known precoder coefficient for this channel from previous studies of it, but obtained very differently here through duality and then a GDFE for receiver 1 that estimates both users. The implementation only uses the top row of G1unb and W1unb (the reason for the W1unb(1,:) above).

Next, the design for BC user 2:

>> H2bc = 50
>> [snrGDFEu2, G2U, W2U, S20, MSWMFU2] = computeGDFE(H2bc, A, 2)

snrGDFEu2 = 16.9028 dB
G2U =
    1     1
    0     1
W2U =
   0.0008       0
  -1.0000   1.0008
S20 = 1.0e+03 *
   1.2510       0
       0   0.0020
MSWMFU2 =
   0.0283
   0.0283
>> B2u=0.5*log2(diag(S20))' =
   5.1444   0.4997
>> sum(B2u) = 5.6441
>> MSWMFU2(2,:)*H2bc*A = 1.0000   1.0000

953
This sum adds the $b_1$ from design 1 to the $b_2$ from user 2’s design. Receiver 2 uses only the bottom row of MSWMF2U and G2U. The lower row of G2 shows that receiver 2 (and user 2) use no precoder (first to be encoded) while the lower row of W2U, when cascaded with the matched-filter matrix is a simple scalar multiply. Again the resultant BC signal is the sum of the BC inputs, and user 1 is crosstalk noise.

This example uses no worst-case noise, nor was there a need to separate primary and secondary users. The combined duality/GDFE design series finds the precoder and receiver processing.

The next example shows a more general example where the precoder does not trivialize.

**EXAMPLE 5.5.5 [ISI and vector BC via duality]** User 2 has a $1 + 0.9D$ channel with some input energy on the dual vector MAC channel. User 1 has a $1 - D$ channel with some input energy on the dual vector MAC channel, with $\bar{N} = 2$ and $\nu = 1$. These energies correspond (through duality) to the BC users each having unit energies $\mathcal{E}_2 = \mathcal{E}_1 = 1$. Both vector BC output noises are white with variance $\sigma^2 = 0.181$. The BC now has user 1 as $1 + 0.9D$ and user 2 as $1 - D$.

\[
H_{ts1} = \left( \frac{1}{\sqrt{0.181}} \right) \begin{bmatrix} 1 & 0.9 & 0 \\ 0 & 1 & 0.9 \\ 0 & 0 & 1 \end{bmatrix}
\]

\[
H_{ts2} = \left( \frac{1}{\sqrt{0.181}} \right) \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}
\]

First, user 1:

\[
H_{ts1} = \left( \frac{1}{\sqrt{0.181}} \right) \begin{bmatrix} 1 & 0.9 & 0 \\ 0 & 1 & 0.9 \\ 0 & 0 & 1 \end{bmatrix}
\]

\[
H_{ts2} = \left( \frac{1}{\sqrt{0.181}} \right) \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 1 \end{bmatrix}
\]

\[
H_{ts1} = \begin{bmatrix} 2.3505 & 2.1155 & 0 \\ 0 & 2.3505 & 2.1155 \end{bmatrix}
\]

\[
H_{ts2} = \begin{bmatrix} 2.3505 & -2.3505 & 0 \\ 0 & 2.3505 & -2.3505 \end{bmatrix}
\]

First, user 1:

\[
A = \begin{bmatrix} \text{eye}(3) & \text{eye}(3) \end{bmatrix};
\]

\[
[\text{snrGDFEu1}, \ G1U, \ W1U, \ S10, \ MSWMFU1] = \text{computeGDFE}(H_{ts1}, \ A, \ 2)
\]

\[
\text{snrGDFEu1} = 2.1606
\]

\[
\begin{array}{cccccc}
1.0000 & 0.9000 & 0 & 1.0000 & 0.9000 & 0 \\
0 & 1.0000 & 0.8006 & 0.1227 & 1.0000 & 0.8006 \\
0 & 0 & 1.0000 & -0.5023 & 0.6591 & 1.0000 \\
0 & 0 & 0 & 1.0000 & 0.4480 & -0.4068 \\
0 & 0 & 0 & 0 & 1.0000 & 0.6579 \\
0 & 0 & 0 & 0 & 0 & 1.0000
\end{array}
\]

\[
\begin{array}{cccccc}
0.1810 & 0 & 0 & 0 & 0 & 0 \\
-0.1227 & 0.1610 & 0 & 0 & 0 & 0 \\
0.5023 & -0.6591 & 0.9558 & 0 & 0 & 0 \\
-1.0000 & -0.4480 & 0.4068 & 1.5842 & 0 & 0 \\
-0.2989 & -1.0000 & -0.6579 & -0.2989 & 1.7244 & 0 \\
0.5262 & -0.6376 & -1.0000 & 0.5262 & -0.6376 & 2.6402
\end{array}
\]

\[
\begin{array}{cccccc}
6.5249 & 0 & 0 & 0 & 0 & 0 \\
0 & 7.2107 & 0 & 0 & 0 & 0 \\
0 & 0 & 2.0463 & 0 & 0 & 0 \\
0 & 0 & 0 & 1.6312 & 0 & 0 \\
0 & 0 & 0 & 0 & 1.5799 & 0 \\
0 & 0 & 0 & 0 & 0 & 1.3788
\end{array}
\]
MSWFU1 =
0.4254   0
0.0522   0.3785
-0.2137   0.4727
0.4254  -0.1923
0.1272   0.3110
-0.2239   0.4727

>> B1u=0.5*log2(diag(S10)) =
1.3530
1.4251
0.5165
0.3530
0.3299
0.2317
>> sum(B1u(1:3)) = 3.2945
upper 3 GU rows are the feedback/precoder for user 1 on user 1, and user 2 on 1; design ignores bottom 3 rows.

>> Wunbias1=inv(S01-eye(6))*inv(G1')*Hgs1' =
0.4254   0
0.0522   0.3785
-0.2137   0.4727
0.4254  -0.1923
0.1272   0.3110
-0.2239   0.4727
bottom 3 rows need not be implemented at receiver 1- only implement top 3.

Now for user 2:

>> [snrGDFEu2, G2U, W2U, S20, MSWMFU2] = computeGDFE(Hts2, A, 2)

snrGDFEu2 = 2.3736
G2U =
1.0000 -1.0000   0   1.0000  -1.0000   0
0  1.0000 -0.8671 -0.1329  1.0000  -0.8671
0  0  1.0000 -0.4585 -0.5415  1.0000
0  0  0  1.0000 -0.5415 -0.4585
0  0  0  0  1.0000 -0.6487
0  0  0  0  0  1.0000
W2U =
0.1810   0   0   0   0   0
0.1329   0.1569   0   0   0   0
0.4585   0.5415   0.7225   0   0   0
-1.0000  0.5415   0.4585  1.7225   0   0
0.3513 -1.0000   0.6487  0.3513  1.7661   0
0.4784   0.5216 -1.0000   0.4784   0.5216  2.2243
S20 =
6.5249   0   0   0   0   0
0  7.3716   0   0   0   0
0  0  2.3841   0   0   0
0  0  0  1.5806   0   0
0  0  0  0  1.5662   0

955
MSWMF2U =

\[
\begin{pmatrix}
0.4254 & 0 \\
-0.0565 & 0.3689 \\
-0.1951 & -0.4254 \\
0.4254 & 0.1951 \\
-0.1494 & 0.2760 \\
-0.2035 & -0.4254 \\
\end{pmatrix}
\]

\[
\begin{align*}
B_{2u} &= 0.5 \log_2(\text{diag}(S_{20})) \\
&= \begin{pmatrix} 1.3530 \\
1.4410 \\
0.6267 \\
0.3302 \\
0.3236 \\
0.2678 \end{pmatrix}
\end{align*}
\]

\[
\text{sum}(B_{2u}(4:6)) = 0.9217
\]

lower 3 rows are feedback from user 2 into user 2.
ignore top 3 rows

first 3 MSWMF2U rows need not be implemented at receiver 2

The previous example illustrates the appropriate setting of the $R_{xx}(u)$ at the output of the special square-root matrix $A$. This process basically absorbs the $A_u$ into the feedback coefficients between different users.

**EXAMPLE 5.5.6 (3-user Design revisited)** The 3-user example with a MAC input of equal energy on all dimensions/users has a dual BC that will have the same data rates. The following commands illustrate the reproduction of the 3 user data rates.

```matlab
N=64;
h=cat(3,[1 0 .8 ; 0 1 1],[.9 -.3 0 ; .5 -1 -1],[0 .2 0 ; .4 -.63 0],[0 0 0 ; 0 .648 0])*10;
H = fft(h, N, 3);
Hbc=zeros(3,2,64);
Rxxm=zeros(1,1,3);
Rxxm(1,1,:)= 64/67*[1 1 1];
Rxxb=zeros(2,2,3,64);
bbc=zeros(3,64);
for n=1:N
Rxxb(:,:,n)=mac2bc(Rxxm, reshape(H(:,:,n),2,1,3));
Hbc(:,:,n)=H(:,end:-1:1,n)';
bbc(1,n)=real(log2(1+Hbc(1,:,n)*Rxxb(:,:,1,n)*Hbc(1,:,n)'));
bbc(2,n)=real(log2((1+Hbc(2,:,n)*Rxxb(:,:,2,n)+Rxxb(:,:,1,n))*Hbc(2,:,n)')/...
(1+Hbc(2,:,n)*Rxxb(:,:,1,n)*Hbc(2,:,n)'));
bbc(3,n)=real(log2((1+Hbc(3,:,n)*Rxxb(:,:,3,n)+Rxxb(:,:,2,n)+Rxxb(:,:,1,n))...
*Hbc(3,:,n)'}/(1+Hbc(3,:,n)*Rxxb(:,:,1,n)*Hbc(3,:,n)'));
end
bvec=sum(bbc') = 132.7477 412.8794 445.1264
bsum=sum(bvec) = 990.7535

A=zeros(2,6,64);
for n=1:N
A(:,:,n)=[ sqrtm(Rxxb(:,:,1,n)) sqrtm(Rxxb(:,:,2,n)) sqrtm(Rxxb(:,:,3,n))];
end
```
This can all be implemented on one call of mu_bc.m

\[ \text{Bu, Gunb, S0, MSWMFunb} = \text{mu_bc}(Hbc, A, [1 1 1], 1); \]

\[ \text{Bu} = \begin{bmatrix} 132.7477 & 412.8794 & 445.1264 \end{bmatrix} \]
\[ \text{sum(Bu)} = 990.7535 \]
\[ \frac{\text{sum(Bu)}}{67} = 14.7874 \]

\[ \text{Gunb}\{:,23\} = \begin{bmatrix} 1.0000 + 0.0000i & 0.1135 - 0.3739i & 6.9871 + 0.9117i & 8.6107 - 12.1481i & -0.5836 - 3.6249i & 1.6766 + 0.5605i \\ 0.0000 + 0.0000i & 1.0000 + 0.0000i & 2.9606 + 17.7883i & 36.1493 + 12.0571i & -0.1570 - 0.6892i & 0.3291 + 0.0869i \\ 0.0000 + 0.0000i & 0.0000 + 0.0000i & 1.0000 + 0.0000i & 0.9887 + 1.8676i & 0.4877 + 0.52861i & 0.1263 + 0.4266i \\ 0.0000 + 0.0000i & 0.0000 + 0.0000i & 0.0000 + 0.0000i & 1.0000 + 0.0000i & -0.2233 + 0.4259i & 0.2315 + 0.0000i \end{bmatrix} \]

\[ \text{A}(\cdot,:,23) = \begin{bmatrix} 0.0964 + 0.0000i & 0.0109 - 0.0360i & 0.2118 + 0.0000i & 0.2094 - 0.3956i & 0.9985 + 0.0000i & -0.2233 + 0.4259i \\ 0.0109 + 0.0360i & 0.0147 + 0.0000i & 0.2094 + 0.3956i & 0.9459 + 0.0000i & -0.2233 - 0.4259i & 0.2315 + 0.0000i \end{bmatrix} \]

\[ \text{MSWMFunb}\{:,23\} = \begin{bmatrix} 0.7049 - 0.2668i \\ 1.1773 + 1.5280i \\ 0.0552 - 0.0807i \\ 0.0460 + 0.0052i \\ 0.0502 - 0.0751i \\ -0.1865 - 0.0200i \end{bmatrix} \]

Where tone \( n = 23 \)'s precoder, linear transmit filter (square root) and corresponding receiver filters appear.

### 5.5.4 Vector DMT and the vector BC

Section 5.5.1's Figure 5.26 illustrates Vector BC broadcast transmitter DMT symbols' synchronization. Vector BC transmit-symbol alignment is easier than for the vector MAC. Each transmitter has the same size DMT symbol and aligns symbol boundaries. However, often systems are bi-directional, so both cyclic prefixes and cyclic suffixes can be used to align both directions of transmission for a downstream (or down-link) or an upstream (up-link) system so that all symbols at the common hub of multiple-access receivers and broadcast transmitters are aligned. Section 4.7.6 also discusses the Zipper method by Isaksson that can be used when a downstream vector BC or an upstream vector MAC share the same users in a bi-directional multi-user channel. Such alignment also ensures that interference from “down back into up” can be easily cancelled at the hub with a single complex coefficient per tone. Such cancellers are often called “NEXT” cancelers (and are not the same as cancellers that exploit spatial correlation of noise in Section 13.3.4). These up-into-down NEXT cancellation must use spatial correlation with multiple antennas. Instead NEXT cancellers remove any effect from the broadcast signal into the received multiple-access signals.

Such alignment will, if the common cyclic extension of DMT partitioning is longer than the length of any of the response entries corresponding to each and all the \( H_u \), which means

\[ \nu T' \geq \max_{u,i} \left\{ \text{length} \left( \hat{h}_{u,i}(t) + (\Delta u) \right) \right\}, \]

lead to no intersymbol interference and to crosstalk on any particular tone \( n \) that is a function ONLY of other users’ signals on that same tone \( n \) of other users. Each tone of the \( \mathcal{L}_y \) receivers’ FFT outputs can then be modeled as

\[ \frac{Y}{\mathcal{L}_y \times 1} = \frac{H}{\mathcal{L}_y \times L_x} \cdot \frac{X}{L_x \times 1} + \frac{N}{\mathcal{L}_y \times 1}, \]

(5.416)
where

\[
H_n = \begin{bmatrix}
H_{1,n} \\
\vdots \\
H_{U,n}
\end{bmatrix}
\]  \hspace{1cm} (5.417)

\[
X_n = \begin{bmatrix}
x_{1,n} \\
\vdots \\
x_{L_x,n}
\end{bmatrix} = \sum_{u=1}^{U} X_{u,n}
\]  \hspace{1cm} (5.418)

\[
Y_n = \begin{bmatrix}
Y_{1,n} \\
\vdots \\
Y_{U,n}
\end{bmatrix}
\]  \hspace{1cm} (5.419)

\[
Y_{u,n} = \begin{bmatrix}
Y_{u,1,n} \\
\vdots \\
Y_{u,L_y,n}
\end{bmatrix} = \hat{H}_{u,n} \cdot V_n + N_{u,n}
\]  \hspace{1cm} (5.420)

\[
\hat{H}_{u,n} = R_{NN}^{-1/2}(n) \cdot H_{u,n} \cdot [A_{1,n} \ A_{2,n} \ldots \ A_{U,n}]
\]  \hspace{1cm} (5.422)

and \( V_n \) is the input vector of unit(identity)-energy user inputs on tone \( n \), with user 1 at the top to maintain vector BC duality-motivated order-reversal. The quantities \( A_{u,n} \) simply form the individual-user autocorrelation matrix components (or energy scalings in the scalar case). The \((l_y,l_x)^{th}\) entry of \( H_{u,n} \) is the DFT of the response from line/antenna \( l_x \) to line/antenna \( l_y \) of user \( u \)’s output. The energy constraint becomes

\[
\sum_n \text{trace} \left\{ R_{XX}(n) \right\} \leq E_x
\]  \hspace{1cm} (5.423)

The input autocorrelation on tone \( n \) is

\[
R_{XX}(n) = \mathbb{E}[X_n \cdot X_n^*] = \sum_{u=1}^{U} R_{XX}(u,n)
\]  \hspace{1cm} (5.424)

This tone-indexed model for DMT leads to tremendous computational reduction with respect to the full precoding (or GDFE) structure. Effectively, \( \bar{N} \) smaller channels of size \( \mathcal{L}_y \times \mathcal{L}_x \) replace a giant channel of size \( \mathcal{L}_y \cdot \bar{N} \times \bar{N} \cdot \mathcal{L}_x \). The GDFE/precoder computational advantage when \( L_x = U \) and \( L_y = 1 \) is a complexity of \( U \cdot \bar{N} \cdot \log_2(\bar{N}) + \bar{N} \cdot U^2 \) versus the much larger \( (\bar{N} \cdot U)^2 \), or if \( \bar{N} = 128 \) and \( U = 4 \), the savings is a factor of about 50 times less computation (262,144 vs 5,632). Figure 5.43 is the result of the modeling.
The input for each tone then decomposes as

\[
R_{XX}(n) = \sum_{u=1}^{U} R_{XX}(u,n) = \sum_{u=1}^{U} A_{u,n} \cdot R_{VV}(u,n) \cdot A_{u,n}^*. \tag{5.425}
\]

Receiver \( u \)'s noise (on any tone) can be found then as

\[
R_{nn}(u,n) = R_{NN}(u,n) + \sum_{i=u+1}^{U} \bar{H}_{u,n} \cdot R_{XX}(i,n) \cdot \bar{H}_{u,n}^*. \tag{5.426}
\]

The noise-equivalent channel is then

\[
\hat{H}_{u,n} = \frac{R_{nn}^{-1/2}(u,n)}{\mathcal{L}_y \times \mathcal{L}_y} \cdot \bar{H}_{u,n} \cdot A_{u,n}. \tag{5.427}
\]

More directly, user \( u \)'s GDFE receiver estimates user \( u \)'s signal by removing users’ \( i = u + 1, \ldots, U \) crosstalk. Then a series of such designs on each tone is done for \( u = 1, \ldots, U \) and the relevant rows of each feedback section \( G_{amb,u,n} \) form the precoder for each tone with known \( R_{XX}(u,n) \).

### 5.5.4.1 Design Assessment

**EXAMPLE 5.5.7 [Vectored VDSL]** Example 5.29’s downstream extension appears here. The downstream direction is a vector BC as in Figure 5.44. The tone spacing is again 4.3125 kHz with a cyclic extension of 640 samples on a sampling clock of \( 2^m \times 2.208 \) MHz, \( m = 4, 5 \). Up to 8192 tones can be used in either direction. Three noise configurations have standards’ specification: The first noise is a flat -125 dBm/Hz noise level (various levels of spatial correlation will be assumed for upstream, but note spatial correlation is of no consequence to downstream or more formally to the vector BC end). This noise is considered to come from outside the coordinated lines and is not “analog front-end” noise. This first noise attempts to model RF noise of radio signals for instance. The second noise is a lower flat level of -140 dBm/Hz and is considered to be “analog front-end” noise and so would always have no spatial correlation (which again is only of interest for the upstream vector MAC that will
also be shown in curves to follow). The third noise is the same as the second, but adds large unknown noise crosstalkers that are outside the vectored group with varying degrees of spatial correlation.

Two frequency plans have been used for a frequency-division separation of upstream and downstream bands. The North American so-called 998 plan allows up and down transmission below 138 kHz (tone 32), and also up-only transmission between 4 MHz and 5.2 MHz and between 8.5 MHz and 17.6 MHz. These are the same as those in Example 5.29, except they expand to 7 bands with the highest additional cut-off frequencies set at 17 and 25 MHz. The results allow FDM of up and down so there is no up-into-down crosstalk of concern. Europe uses a 998 plan with larger upstream frequency capability.

Figures 5.45 - 5.47 illustrate the achievable data rates with vectoring (up and down) over no vectoring. From this example and Example 5.29, vectoring in either direction provides an enormous gain over expectations. G.fast and G.mg.fast use tone spacing 12 × 4.3125 kHz, and vectored DMT on shorter copper lines for data rates to 5 Gbps on a single phone line.

5.5.5 Generation of the Vector BC Capacity Rate Region

The steps for tracing the vector BC Capacity Region are:

1. Create a dual vector MAC channel (with coefficients \( \tilde{H} \) and noise autocorrelation \( I \)).

2. for each \( b' \) with \( b'_1 = 0, ..., b_{1,\text{max}}, ... \) \( b'_U = 0, ..., b_{U,\text{max}} \) with increments selected appropriately and maximums chosen sufficiently large to be outside the rate region (i.e., equal to the single user capacity for all other users zeroed)

   (a) Find the energy vector \( \mathcal{E} \) for a given \( b \) on the dual vector MAC using the minPmac program of Section 5.4.

   (b) if \( \sum_u \mathcal{E}_u \leq \mathcal{E} \), then the point is in the region, so \( c_{\text{new}}(b) = \{b' \cup c_{\text{old}}(b)\} \).

3. Trace the boundary for of \( b \) in Step 2 for which \( \sum_u \mathcal{E}_u = \mathcal{E} \).
Figure 5.45: VDSL2 Data rates for vector BC and vdmt for Configuration 1.

Figure 5.46: VDSL2 Data rates for vector BC and vdmt for Configuration 2.
Noise 3, 7-band plan, Option 1

Figure 5.47: VDSL2 Data rates for vector BC and vdmt for Configuration 3.
5.6 Gaussian IC with GDFE

As in Section 2.6.7 and now focusing on the Matrix-AWGN IC, allocation of energy and information to IC users divides into situations of Section 5.6.1’s central and Section 5.6.2’s distributed management. Section 2.6.7’s consensus management will be viewed here as intermediate and allowing some information passing between users in overhead messages and effectively part of an enhanced distributed management in Section 5.6.3. IC random-access management largely reverts to Section 2.6.7.4’s collision avoidance (already studied and not repeated here) that increasingly yields to Section 5.6.2’s consensus management alternatives. This section further refines central management description versus distributed management before proceeding to optimization within each situation.

The Centrally Controlled Gaussian IC (CIC) Figure 5.48 describes the CIC. The CIC situation includes the Distributed Antenna System (DAS) concept. The CIC permits central assignment of user data rates, codes, bit distributions and corresponding user input autocorrelation matrices $R_{xx}(u)$ through the central controller that knows all noise autocorrelation matrices $R_{nn}(u)$ and channel $H_{ij}$ entries. This CIC is Chapter 2’s IC. For instance, Vector DMT and multi-user GDFE’s are possible at each user’s CIC receiver. All CIC users synchronize to a common symbol rate, $1/T$.

CIC DAS versions may have $L_{x,u} > 1$ and/or $L_{y,u} > 1$. DAS uses space-division multiplexing of different users’ transmitted energy to improve spectrum efficiency. Similarly, frequency- or time-division multiplexing of these users’ signals is also possible. IC receivers can employ GDFE receivers for the known codes of all users; these receivers may decode some or all other users first wherever possible.

---

76The word distributed in DAS refers to the antennas’ locations and not to the control; indeed most DAS systems presume and exploit central coordination of their various transmissions, particularly in space.
according to order $\pi_u$ at each receiver $u$. Users' transmitters may employ individual matrix filters that a central controller supplies. The controller may separate dimensional energization per user or may allow users to share dimensions.

The Locally Controlled Gaussian IC (LIC) Theoretically, local control is not truly an extra IC constraint because a designer could presumably guess the correct codes (and receivers if given sufficiently long time could presumably ascertain the constellations and codes used on all other users if those codes were not quite, but almost, Gaussian). However, the LIC refines the constraints where the central design of spectra and bit distributions may not be feasible. This may appear then “random” to all other users, possibly stationary or non-stationary.

The LIC may presume only that the transmitter and receiver $u$ know only the following 4 items (see Figure 5.49):

1. the received noise autocorrelation $R_{\text{noise}}(u)$ (presumed Gaussian including contributions in aggregate from all other users and the Gaussian noise)
2. its own channel $H_{uu}$ (but not any $H_{iu}$ nor $H_{ui}$ where $i \neq u$)
3. The transmit autocorrelation $R_{\text{xx}}(u)$ of its own channel
4. the bit distribution and thus total rate of its own channel $b_{u,n}$ and $b_u$ respectively.

Furthermore, multuser GDFE/successive-decoding will not occur in the LIC, except possibly in single-user form for any particular user, with any decisions only within the dimensions that user energizes.
5.6.1 Centrally Managed Gaussian IC Optimization

Section 2.9.1’s MAC-set approach views the IC as \( U \) parallel MACs. Thus, this subsection builds upon Section 5.4.3’s MAC optimization methods. Basically, the \( \theta \) expands to \( U^2 \) entries \( \Theta \) (a length \( U \) order for each of \( U \) receiver/detectors) and the \( R_{XX}(u) \) optimization corresponds to a sum of \( U \) quantities (each with their own \( U \)-element subset of \( \Theta \) but a common \( R_{XX}(u) \) user set), each term of which remains convex as does consequently their sum.

5.6.1.1 Optimization of the CIC

The minimum IC energy sum\(^{77}\) corresponds to the set of \( U \) input energies that minimize a weighted sum for a MAC-set of canonical (GDFE) designs to that target a common fixed rate vector \( b_{\text{min}} \):

\[
\min \left\{ R_{XX}(u) \right\} \quad \sum_{u=1}^{U} w_u \cdot \text{trace} \{ R_{XX}(u) \} \tag{5.428}
\]

\[
ST: \quad b_i \succeq [b_{i,1,\text{min}}, b_{i,2,\text{min}}, \ldots, b_{i,U,\text{min}}]^* = b_{\text{min}}^* \succeq 0 \quad \forall \ i \in U
\]

\[
\mathcal{E} \succeq 0
\]

\[
\text{present order } \Pi
\]

where \( i \in U \) is an IC receiver index. The individual IC receiver \( i \) data rate for any user \( u \) is equal to \( b_{\text{min},u} \) if the constraint is active (which corresponds to a decoding order \( \Pi \) where user \( u \) must be reliably decoded at receiver \( i \)). If the constraint is inactive, then the data rate is arbitrary positive number (which corresponds to a decoding order where \( u \)’s crosstalk is added noise and user \( u \) need not be reliably decoded at receiver \( i \)). There are \( U^2 \) constraints always, and they may be active (Lagrange multiplier is nonzero) or inactive (Lagrange multiplier is zero). When more than one Lagrange multiplier is zero at receiver \( i \), the corresponding remaining users may have any order subsequent to the decoding of user \( i \) at receiver \( i \). A subgradient descent method for such non-decoding users will observe the current order (as with the MAC indicated by the Lagrange multiplier relative amplitudes) and target a \( b_{i,u} = 0 \) for this user instead of the nominal \( b_{\text{u,min},u} > 0 \) for this step.

The vector \( w = [1 \ldots 1]^* \) corresponds to the energy sum; more generally \( w \succeq 0 \). Theoretically, (5.428) always has a solution for a given \( b \). A difference with the MAC is that there are actually \( U^2 \) IC constraints on the \( U \) receiver rate vectors \( b \). Similar to the MAC, (5.428) may not produce a \( b \in C_{IC}(b) \) because the designer may not know a priori the capacity region in selecting \( b \). When this happens, (5.428)’s minimized energy vector solution \( \mathcal{E} \) may exceed the energy limit \( \mathcal{E} \not\succeq \mathcal{E}_x \). (5.428) describes the capacity region point with lowest \( w^* \mathcal{E} \). When \( w = [1 1 \ldots 1]^* \), this is the minimum-sum-energy point on the capacity-energy-region boundary \( \mathcal{C}_{b,IC}(\mathcal{E}) \). As in Subsection 5.4.4, an extension determines if \( b \) is admissible, that is \( b \in C_{IC}(b) \) for the given \( \mathcal{E} \). IC individual users’ energies satisfy \( \sum_{n} \text{trace} \{ R_{XX}(u,n) \} \leq \mathcal{E}_u \), \( \forall u \in U \) when for at least one user \( L_x > 1 \). When \( L_{x,u} = 1 \forall u \in U \), these terms reduce to the individual-user energies \( \mathcal{E}_u \) that arise from summing energy over all individual user’s tones. Equation (5.428) specializes with vector DMT to

\[
\min \left\{ R_{XX}(u,n) \right\} \quad \sum_{u=1}^{U} \sum_{n=0}^{N} w_u \cdot \text{trace} \{ R_{XX}(u,n) \} \tag{5.429}
\]

\[
ST: \quad b_i = \sum_{n=0}^{N} [b_{i,1,n} b_{i,2,n} \ldots b_{i,U,n}]^* \succeq b_{\text{min}} \succeq 0 \quad \forall \ i \in U
\]

\[
b_{i,u,n} \geq 0 \quad \forall \ i, u, n
\]

Each tone has an independent MAC-set of GDFEs with the set of minimum-sum energies \( \mathcal{E}_{\text{min}} \) for \( b_{\text{min}} \) over all tones. Equation (5.428) follows from Equation (5.429) by setting \( N = 1 \) mathematically, but that ignores the vector-DMT/tonal-GDFE’s large simplification with temporal ISI. This appears almost identical to the MAC, except that there are now many more side constraints (\( U \) sets of \( U \) constraints each

\(^{77}\)Assumes that all users employ capacity-achieving codes individually.
Section 5.4.3’s comments on vertex-touching for the fixed-$R_{XX}$ generalize only in that the hyperplane touch/intersection will be at a vertex corresponding to an $\mathbf{z}_{\min}$ point for the order implied by the Lagrange multipliers ($\Theta$), which will imply a possibly different order for each receiver. Equation (5.429)’s Lagrangian is

$$
L(R_{XX}(n), b, w, \Theta) = \sum_{u=1}^{U} \left( w_u \cdot \left[ \sum_{n=0}^{N} \text{trace} \{ R_{XX}(u, n) \} \right] - \theta_{i,u} \cdot \left\{ \left[ \sum_{n=0}^{N-1} b_{i,u,n} \right] - b_u \right\} \right) \forall i \in U \tag{5.430}
$$

(presuming no single-side-band uses of tone $n = N$ tone if baseband real). Each side-constraint term tacitly implies that a $\{b_{i,u,n}\}$ set for tone $n$ must be within tonal-MAC-set matrix-AWGN ($\widetilde{H}_{i,n}$) achievable region, or equivalently the intersection of these convex capacity regions (over $i$), which remains convex, is

$$
b_n \in A_n(R_{XX}(u,n), b_n) = \bigcap_{i=1}^{U} A_{i,n}(R_{XX}(u,n), b_{i,n}), R_{XX}(u,n), u = 1,...,U. \tag{5.431}
$$

The non-negative Lagrange side-constraint multipliers $\theta_{i,u}$ are rate-control constants that contribute nothing to the Lagrangian’s value when the rate constraints are met, but can cause a suboptimal Lagrangian value when attempting to meet any tone’s corresponding side constraint (5.431). Equation (5.431)’s minimization for maximal $\Theta \succeq 0$ elements solves (5.429). Any solution has $b_n \in A_n(b_n)$, because energy can increase in a way that minimizes the unbounded weighted energy sum and thus be larger than the desired $E$. This achievable-region constraint, however, simplifies a complete solution’s algorithmic development. The solution to the dual MAC set of weighted rate-sums problem

$$
\max_{\{R_{XX}(u,n)\}} \sum_{u=1}^{U} \theta_{i,u} \cdot \left\{ \sum_{n=0}^{N} b_{i,u,n} \right\} \forall i \in U \tag{5.432}

ST : \ E \geq \sum_{n} \text{trace} \{ R_{XX}(u,n) \} \geq 0 ,
$$

has essentially the same Lagrangian (differs only by constants that disappear under differentiation). Thus the same algorithm can address:

**The original min weighted energy sum**: is the sum with specified energy weights $w \succeq 0$, where $\Theta \succeq 0$ are the side-constraint multipliers to minimize (maximize with negative sign) rate-constraint impact and determine all receivers corresponding orders, or

**max weighted rate sum** for the order implied by the specified $\Theta \succeq 0$, where $w \succeq 0$ as the side-constraint multipliers to minimize (maximize with negative sign) energy-constraint impact. Essentially this rate is $U$ times the rate sum, because all IC users target the same rate sum $\bar{b} = \sum_u b_u$, but that constraint is not evident in the dual solution.

### 5.6.1.2 Tonal Decomposition of the Lagrangian

Interchange of Equation (5.430)’s finite sums over indices $u$ and $n$ obtains

$$
L(R_{XX}, b, w, \Theta) = \left( \sum_{i=1}^{U} \sum_{u=1}^{U} \theta_{i,u} \cdot b_{i,u} \right) + \sum_{n=0}^{N-1} \left[ \sum_{u=1}^{U} \sum_{i=1}^{U} w_u \cdot \text{trace} \{ R_{XX}(u,n) \} - \theta_{i,u} \cdot b_{i,u,n} \right] \tag{5.433}
$$

---

78The frequency-space $R_{XX}$ notation will use non-subscription arguments to include user and tonal dimensions, e.g.

$R_{XX}(u,n)$ is an $L_{x,u} \times L_{z,u}$ smallest matrix, while $R_{XX}(u)$ includes all tones as a 3-dimensional tensor and similarly $R_{XX}(n)$ includes all users as a 3-dimensional tensor, while $R_{XX}$ includes all tones and all users for a 4-dimensional tensor. $R_{XX}$ is often a 4-dimensional array/tensor in vector MAC software programs.
where $L_n(R_{XX}(n), b_{i,n}, w, \Theta)$ is a tonal Lagrangian term. For a given $\Theta$, (5.433)’s first right-side sum term is not a function of $R_{XX}(u, n)$, because $b_u$ is given even though its exact constituency $b_{i,u,n}$ is not ($N > 1$). $L_n(R_{XX}(n), b_{i,n}, w, \Theta)$ for any given $\Theta$ and $w$ depends only on tone $n$ quantities. Thus, the overall “max-min” is

$$L^* = \max_{\Theta} \left\{ \sum_{n=0}^{N-1} L_{\min}(\Theta, n) + \sum_{i=1}^{U} \sum_{u=1}^{U} \theta_{i,u} \cdot b_{i,u} \right\} \left(\text{independent of } R_{XX}(u,n), n\right) \tag{5.434}$$

where for each $n$ and $\Theta$, noting $R_{XX}(n) = \text{blkdiag} \{ R_{XX}(U,u), ..., R_{XX}(1,n) \}$,

$$L_{\min}(\Theta, n) \overset{\Delta}{=} \min_{\{R_{XX}(u,n)\}, b_{i,u,n}} L_n(R_{XX}(n), b_{i,n}, w, \Theta) \quad . \tag{5.435}$$

The last term in (5.434) has the given fixed $b_i, u$ (from the given $b$) values in the sum. (5.313)’s minimization is thus $N$ independent problems for a given $\Theta$. The outer $\Theta$ maximization proceeds after these $N$ inner minimizations.

There is also an achievable-region constraint that relates $b_{i,n}$ and $R_{XX}(n)$, or in this text’s context GDFE constraint, according to the achievable region:

$$b_{i,n} \in \left\{ b_{i,n} \mid 0 \leq \sum_{u \leq U} b_{i,u,n} \leq \log_2 \left( \sum_{i=1}^{U} \sum_{u=1}^{U} \tilde{H}_{i,u,n} \cdot R_{XX}(u,n) \cdot \tilde{H}_{i,u,n}^{-1} \right) + I \right\} = A_{i,n} \left( \{ R_{XX}(n) \}, \tilde{H}_{i,n} \right) \quad , \tag{5.436}$$

where

$$\tilde{H}_{i,u,n} = R_{-\frac{1}{2}} (n) \cdot H_{i,u,n} \quad . \tag{5.437}$$

Summing the solutions of (5.435) over the tones provides

$$L_{\min}(\Theta) = \sum_{n=0}^{N-1} L_{\min}(\Theta, n) \quad . \tag{5.438}$$

This $L_{\min}(\Theta)$ has maximum at $\theta^*$, as in (5.434), for

$$L(\theta^*) = \sum_{i=1}^{U} \sum_{u=1}^{U} \theta_{i,u}^* \cdot \frac{b_{i,u}}{L_{\min}(\Theta^*)} \quad \text{point in } A(R_{XX}, \tilde{H}) \quad \tag{5.439}$$

**Computation of tonal mutual information:** The relationship between the bits per user/tone $b_{i,u,n}$ and tone autocorrelation $R_{XX}(u, n)$ follows from the known best order (that is at given $\Theta$), and the tonal-GDFE/achievable-region relation, which is the difference between two chain-rule mutual information terms, becomes (with zero gap)

$$b_{i,u,n} = \log_2 \left| \sum_{j=u}^{U} \tilde{H}_{i,\pi_j^{-1}(j),n} \cdot R_{XX}(\pi_j^{-1}(j), n) \cdot \tilde{H}_{i,\pi_j^{-1}(j),n}^* + I \right| - \log_2 \left| \sum_{j=u-1}^{U} \tilde{H}_{i,\pi_j^{-1}(j),n} \cdot R_{XX}(\pi_j^{-1}(j), n) \cdot \tilde{H}_{i,\pi_j^{-1}(j),n}^* + I \right| \quad . \tag{5.440}$$
When computing the sum \( \sum_{i=1}^{U} \sum_{u=1}^{U} \theta_{i,u} \cdot b_{i,\pi_{i}^{-1}(u)} \), each “log” term in (5.440) appears twice with a different value of \( \theta_{i,u} \), so with simplification (and \( \theta_{i,\pi_{i}^{-1}(0)} = 0 \)) the sum simplifies to

\[
\sum_{i=1}^{U} \sum_{u=1}^{U} \theta_{i,u} \cdot b_{i,\pi_{i}^{-1}(u)} = \sum_{i=1}^{U} \sum_{u=1}^{U} \left\{ \left[ \theta_{i,\pi_{i}^{-1}(u)} - \theta_{i,\pi_{i}^{-1}(u-1)} \right] \cdot \log_{2} \left( \sum_{j=u}^{U} \tilde{H}_{i,\pi_{i}^{-1}(j),\pi_{i}^{-1}(j)},n \cdot R_{XX}(\pi_{i}^{-1}(j),n) \cdot \tilde{H}_{i,\pi_{i}^{-1}(j),\pi_{i}^{-1}(j)},n + I \right) \right\}.
\]

Equation (5.441) relates the weighted rate sum to the tonal autocorrelation matrices the order; \( \pi_{i}^{-1}(u) \geq \pi_{i}^{-1}(u-1) \) for each receiver \( i \) and ensures that (5.323) remains concave in \( R_{XX}(u,n) \). Degraded MACs have linearly dependent \( \tilde{H}_{i,u} \), or equivalently two users combine effectively into a macro-user. Such degraded channels have two successive equal \( \theta \) values in (5.323). This situation corresponds to a need to time-share if the given desired \( b \) is not coincidentally a vertex already; however if only the maximum weighted rate sum is of interest, any point on the face produces that maximum. This concave function can then be maximized to find the best set of autocorrelation matrices for this tone and the given value of \( \Theta \), as in the next subsection.

Section 5.4.3’s \( R_{XX} \) step now modifies the gradient and Hessian to be respectively

\[
\nabla L_{R_{XX}(u)} = \frac{1}{\ln(2)} \sum_{i=1}^{U} \sum_{j=1}^{U} \theta_{i,j} \cdot \left[ \tilde{H}_{i,u} \cdot R_{Y_{i}Y_{i}}^{-1}(u) \cdot \tilde{H}_{i,u} \right] - w_{u} \cdot I + \sum_{j=1}^{L_{x,u}} \lambda_{R_{XX}}^{-1}(j) \quad (5.442)
\]

where for \( u = 1 \ldots U \)

\[
R_{Y_{i}Y_{i}}^{-1}(u) = \left[ R_{Y_{i}Y_{i}}^{-1}(u) - R_{Y_{i}Y_{i}}^{-1}(u-1) \cdot \tilde{H}_{i,u} \cdot \left( R_{X_{i}X_{i}}(u) + \tilde{H}_{i,u} \cdot R_{Y_{i}Y_{i}}^{-1}(u-1) \right) \right] \cdot \tilde{H}_{i,u}^{-1} \cdot R_{Y_{i}Y_{i}}^{-1}(u-1) \quad (5.443)
\]

The \( \sum_{j=1}^{L_{x,u}} \lambda_{R_{XX}}^{-1}(j) \) term corresponds to a simplified non-negative user-energy side constraint in the form of individual \( \sum_{j=1}^{L_{x,u}} \ln(\lambda_{u}(j)) \) singular values’ logarithms penalizing infinitely any user’s negative energy. The Hessian has \( U^{2} \) components (\( \ell = 1, \ldots, U, j = 1, \ldots, U \))

\[
\nabla^{2} L_{u}(\ell, j) = \frac{1}{\ln(2)} \sum_{i=1}^{U} \sum_{k=\max(u,\ell)}^{U} \theta_{i,k} \cdot \text{trace}\left\{ \tilde{H}_{i,u} \cdot \tilde{H}_{i,u}^{-1} \cdot R_{Y_{i}Y_{i}}^{-2}(k) \cdot \tilde{H}_{i,u}^{-1} \cdot \tilde{H}_{i,u}^{-1} \right\} - \left\{ \lambda_{R_{XX}}^{2}(u) \right\} \quad (5.444)
\]

This Hessian’s inverse will exist when at least one channel is non-degraded; otherwise a pseudoinverse is necessary and there is ambiguity that does not appear in the weighted energy sum, but would occur with an energy-vector constraint.

The order step appears the same as the MAC, but with \( \Theta \) replacing \( \theta \) so a larger dimensionality on the Lagrange multipliers. The \( \Delta b \) consequently has \( U^{2} \) terms, and will use the elements of the fixed \( b \) for each receiver if the corresponding user and order mandate that user be reliably decoded at that receiver. If not, the corresponding elements of \( \Delta b \) then target 0, that is \( \Delta b_{i,u} = 0 - b_{u} \). The ellipsoid descriptions enlarge as:

\[
O_{k}(\Theta) \triangleq \left\{ \Theta \mid (\Theta - \Theta_{k})^{*} \cdot A_{k}^{-1} \cdot (\Theta - \Theta_{k}) \leq 1 \right\} \quad (5.445)
\]

The matrix \( A \)’s eigenvalues \( \lambda_{A,u} \) specify the semi-axis lengths as \( \sqrt{\lambda_{A,u}} \). Defining \( \Delta \Theta_{k} \triangleq (\Theta - \Theta_{k}) \), the new ellipsoid is then

\[
O_{k+1}(\Theta) = O_{\text{min}} \left\{ O_{k}(\Theta) \cap \Delta \Theta_{k}^{*} \cdot \Delta b_{k} \geq 0 \right\} \quad (5.446)
\]

When \( U' > U \), equivalently at least one \( L_{x,u} > 1 \), the matrix \( \tilde{H}_{i,u} = F_{i,u} \cdot A_{i,u} M_{i,u}^{*} \) redefines as \( \tilde{H}_{i,u} \rightarrow \tilde{H}_{i,u} \cdot M_{u} \) where user \( u \) has autocorrelation matrix \( M_{u} \cdot \text{Diag}(E_{u}) \cdot M_{u}^{*} \).
The ellipsoid-normalized rate-difference sub-gradient vector is

\[ \Delta \hat{b}_k \triangleq \frac{\Delta b_k}{\sqrt{\Delta b_k^* \cdot A_k^{-1} \cdot \Delta b_k}} , \]  

which simplifies (5.446) and (5.445) to produce a new ellipsoid that covers the intersection of the half plane and the old ellipsoid:

\[
\Theta_{k+1} = \Theta_k - \frac{1}{U+1} \cdot A_k \cdot \Delta \hat{b}_k \text{ new center (5.448)}
\]

\[
A_{k+1} = \frac{U^4}{U^4 - 1} \left[ A_k - \frac{2}{U^2 + 1} \cdot A_k \cdot \Delta \hat{b}_k \cdot \Delta \hat{b}_k^* \cdot A_k \right] \text{ new axes . (5.449)}
\]

These steps execute once during each order step after the \( R_{XX} \) step updates the current autocorrelation matrix. The first step in (5.448) is a steepest descent (with \( A \) as the inverse of the Hessian matrix \( A^{-1} \) weighting the direction by the ellipsoid axes’ lengths together) with step-size \( 1/(U^2 + 1) \) that moves so that \( \Delta \theta_k \) moves into the half-plane where \( \Delta \theta_{k+1} \cdot \Delta \hat{b}_k \geq 0 \). The second update in (5.335) follows through insertion of (5.334) into (5.331) and solving for \( A_{k+1} \) in terms of \( A_k \), \( U \), and \( \Delta \hat{b}_k \). The new ellipsoid’s volume decreases as

\[
\text{vol}(O_{k+1}) \leq e^{-\frac{1}{2k^2}} \cdot \text{vol}(O_k) \ , \quad (5.450)
\]

implying exponential convergence in terms of the number of iterations through the two successive steps.

**Initialize Ellipsoid:** To determine the initial ellipsoid that contains \( \Theta^0 \) on any order step, this step’s initialization selects a random order and a user-bit/symbol distribution \( b_i^0 \) such that

\[
\begin{aligned}
   b_{j,i \neq u} &= b_i^0 , \forall j \in U \\
   b_{j,u} &= b_u^0 + 1 , \forall j \in U 
\end{aligned} \quad (5.451)
\]

Initialization makes a single-pass\(^79\) through Chapter 4’s fixed-margin iterative water-filling, for the selected order and bit distribution in (5.337) and (5.338), generates the bit distribution \( b^0 \), thus generating a set of \( \{ R_{XX}(u,n) \} \). This set of autocorrelation functions substitutes into the Lagrangian equation, which must always be non-negative for \( \Theta^0 \geq 0 \), so

\[
0 \leq L_{\min}(\Theta^0) \leq \left[ \sum_{u=1}^{U} \sum_{n} w_u \cdot \text{trace}(R_{XX}(u,n)) \right] + \left[ \sum_{j=1}^{U} \sum_{u=1}^{U} \left( b_{j,u} - \sum_{n} b_{j,u,n} \right) \right] \cdot \theta_u^0 \quad (5.453)
\]

must hold. Rearranging this equation using (5.338) leads to

\[
0 \leq \theta_u^0 \leq \sum_{u=1}^{U} \sum_{n} w_u \cdot \text{trace}(R_{XX}(u,n)) = \sum_{j=1}^{U} \theta_{j,u,max} \ , \quad (5.454)
\]

The step in (5.452) repeats for each user (that is incrementing each user successively by one bit as in (5.451) and (5.452)) to generate a \( \theta_{j,u,max} \) for each, as in (5.454). Thus by executing \( U \) single-pass FM IWs for each of the \( U \) bit distributions (each incrementing one user by one bit while others are held constant) and each of the \( U \) possible IC receivers, a box containing \( \Theta^0 \) is obtained. The eigenvector-axised ellipsoid that covers this box has diagonal Hessian with squared lengths at least \( \theta_{j,u,max}^2 \) on each of these axes.

\[
A_0^{-1} = \begin{bmatrix}
0 & \cdots & 0 \\
\vdots & \ddots & \vdots \\
0 & \cdots & 0
\end{bmatrix} \quad (5.455)
\]

This step’s algorithm runs until \( \theta_{j,k+1} - \theta_{j,k} < \epsilon \).

\(^79\)The water-filling need be executed only once for each user because a single pass using \( R_{\text{noise}}(u) \) for some order will produce a solution, which is all that is necessary for initialization.
5.6.2 Distributed Management Strategies and Optimization

Distributed management essentially means that each IC receiver $u$ and transmitter $u$ can measure the channel as if all other users were noise. This of course reduces performance with respect to the optima in Section 5.6.2. Nonetheless, it may be a practical necessity. This section then describes some methods, including the optimum spectrum balancing (OSB) method that is best for distributed management, which admits no GDFE or successive decoding but does presume a central management allocates spectra (dimensional energy more generally) to all transmitters.

5.6.2.1 Iterative Water-filling

Chapter 2’s Iterative water-filling (IW) addresses the vector MAC’s rate sum. However, IW can also apply well to the distributed-management IC, where IW improves overall IC performance in achieving what is sometimes called a “Nash equilibrium,” a point where all users inputs are such that none can change individually without degradation. Distributed-management’s use of IW has all receivers treat all other users’ signals as noise, Luo and Pang\(^{80}\). IW convergence for the IC situations largely occurs. Perhaps of greater interest is that of diagonally dominant channels. Such channels imply that $|H_{uu}(f)| >> |H_{ui}(f)|$ when $i \neq u$. In effect the transfer function of the user’s channel is significantly larger than the transfer function from any other user into this same user. There may be many points to which IW can converge on the IC (but any may be acceptable improvements over some less adaptive design approaches), unlike IW’s unique convergence point of the vector MAC.

Essentially, each user being as polite as possible is about the only acceptable solution for a distributed-management IC, given a situation where no successive decoding is used by any of the IC receivers. Local/central-management constraints relax slightly to the improvements in Subsections 5.6.3 and 5.6.3.3.

---

5.6.2.2 The IW Algorithm for the distributed-management IC

In Figure 5.50’s IW for the IC, each user water-fills energy by treating all other crosstalking signals as Gaussian noise. User $u$’s spectrum water-fills using the noise-referred to channel input spectrum curve $\left[\sum_{i \neq u} H_{ui}^2 \cdot S_i(f) + S_n(f)\right] / |H_{uu}|^2$ as (more generally with coding gap $\Gamma$ and presuming energy-per-dimensional quantities)

$$\lambda_u = \mathcal{E}_{u,n} + \frac{\Gamma \sigma_n^2 + \sum_{i \neq u} |H_{ui}|^2 \cdot \mathcal{E}_{i,n}}{|H_{uu,n}|^2}.$$

(5.456)

In actual use, each user would presumably implement locally a water-fill-based loading algorithm that treats all other users as noise. All users may adjust simultaneously. Simulations instead usually hold all other users constant and implement a water-fill loading algorithm for the user of interest. This simulation then iterates through all users several times until the spectra of all users have converged. Field use simply converges if the $\mathbf{b}$ is within the FM IW achievable region.

Figure 5.51: Simple IW Example.

Figure 5.50 also illustrates IW’s simulation algorithm. Basically each user successively water fills
as if all others are noises. After a few to several passes through the procedure (as indicated by $j_{\text{max}}$ - typically $j_{\text{max}} = 5$ is sufficient), it converges to stable $\{R_{xx}(u)\}$ for all users.

The use of FM water-filling is important because it corresponds to polite energy use where no user has excessive margin. In particular, the non-unique convergence point to which IW converges is usually then a good one. It is not necessarily optimum in all cases, but it is usually close and almost always by definition improves over a static-spectrum choice. Figure 5.51 illustrates a hypothetical 5-dimensional channel (Dimensions have indices A, B, C, D, and E to emphasize that dimensions may be space, time, and/or frequency). Each of two users can use any of the 5 bands. In step one, user 1 water-fills (dark blue), which creates a crosstalk/interference to user 2 on Figure 5.51 step 1’s right spectrum (light blue), changing the channel-input referenced spectrum ($1/g$, red color) for user 2. User 2 then waterfills in step 2 (right side) with the dark-blue user 2 energy. However, this then causes user 1 (light blue) to have interference/crosstalk that renders its dark blue energy sub-optimum. User 1 then waterfills again against the new combined noise, but causes user 2 previous water-fill to be slightly suboptimum. At each step, especially with FM IW, the deviation from optimum for both users reduces, eventually leading to a simultaneous waterfill spectra for both users with the other as noise.

5.6.2.3 Near/Far ICs

Figure 5.52 illustrates the near/far” problem where signals from a near transmitter overwhelm signals from a far transmitter at a receiver location. This situation has a $H_{\text{near} \rightarrow \text{far}}$ that approaches or exceeds $H_{\text{far} \rightarrow \text{far}}$ in magnitude. The far user’s data-rate reduction can be very large.

The near-far problem occurs often when different systems’ users share a common bandwidth. For instance, an “unlicensed” wireless frequency band may allow many systems’ simultaneous use (See also Section 2.6.7 on Scheduling and CSMA). It also occurs in wireless systems uplink when a user in one cell may simultaneously transmit on a downlink frequency band used by another cell. Wireline systems with electromagnetically-radiated crosstalk often also experience near-far when one or more transmitters are located much closer to receivers than other transmitters. While the methods of Section 5.6.1 might best address the common-dimension interference, distributed management imposes limitations. Different MIMO systems may spatially target a given direction with intense radiation that overwhelms another systems co-radial user (or one of the spatial sidelobes that always exist with $L < \infty$ might just accidentally be strong at another systems’ user-receiver location. Fixed-Margin (FM) water-filling
minimizes energy emission to achieve a certain data rate, thereby reducing also interference to other systems. Thus, most practical IW systems use FM. Power Control systems are essentially FM IW with a single dimension optimized.

EXAMPLE 5.6.1 [2-dimensional IW IC example]

Figure 5.53 provides a simple $U = 2$-user IC example helps illustrate the convergence of IW as well as the effects in the following Example. Table 5.1 and Figure 5.51 further detail the steps of IW for the channel of Figure 5.53. Initially user 1 places equal energy on both dimensions. Since the crosstalk transfer into both bands of user 2 is unit gain on this channel, the calculation of the inverse (noise plus crosstalk/ referenced by the unit channel gain) is simply $1 + .9^2 = .91$ for dimension A and $1 + .1^2 = .11$ for dimension B. Waterfilling sets user 2’s energy on both these bands, plus the normalized noise, equal., which leads to user B energies of 0.6 and 1.4 in the respective dimensions. These now impose crosstalk on user 1. Table 5.1’s 4th row then executes water-fill for user 1 and zeros dimension B and places all energy on dimension A. User 2 reacts and places almost all its energy in dimension B (1.81), but still some in dimension A (.19). Subsequent waterfill steps provide no further improvement. User 1 has .89 bits/symbol and user 2 has 4.4 bits/symbol, with sum 5.29 bits/symbol. If instead both users equally energized each dimension with 1 unit of energy, the data rates are lower, summing to .7 bits for user 1 and 1.07 (A) and 3.18 (B) = 4.25 bits/symbol for user 2. There is a small improvement here, but improvements can be much larger with larger dimensionality (more degrees of freedom and many users). The IW solution is also better than all user 1 energy on dimension B and all user 2 energy on dimension A, which leads to a sum of almost exactly 5 bits/symbol.
### Table 5.1: Simple IW Example

<table>
<thead>
<tr>
<th>User 1</th>
<th>Band A</th>
<th>Band B</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>E_{x1}</strong></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>E_{x2}</strong></td>
<td>1 + (9)^2 = 81</td>
<td>1 + (1)^2 = 11</td>
</tr>
</tbody>
</table>

**Example 5.6.2 Bi-directional DSL near/far Example**

Figure 5.54 illustrates a situation where 25 users’ channels all have high crosstalk into one another if they try to transmit up and down on the same dimension. Otherwise, all have the same channel and symmetric crosstalk into one another. In this simulation, the number of users was actually 50 because the lines are all used bi-directionally. Echo cancelation is used on any loop (see Chapter 4) to isolate upstream and downstream transmissions, but crosstalk into all other loops is presumed. The lower curve uses a 64PAM system with equal energy on all users’ dimensions. The upper curve system used IW with up to 4096 4.3125 kHz tones, with no restriction on frequencies used up or down. All links individually and locally use FM IW and the bandwidth is determined adaptively for each. As is clear, the IW doubles to triples the rate. Such an improvement is solely caused by the better adaptive polite determination of spectrum use.
Figure 5.54: Improvement of IW over current static design in symmetric DSL transmission.

Figure 5.55 shows the up and down spectrum of one user (almost all are the same), which clearly attempts to be an “FDM” solution, but not completely. It is one that is achieved only through distributed management. Coordinated management systems will perform yet much better in this situation, providing another factor of 3 roughly in data rate. Nonetheless, the distributed solution is considerably better than the “flat everywhere” without the IW. The lower frequencies are used in both directions since crosstalk for lower-frequency tones is lower for the transmission lines used in this example (twisted-pairs) at lower frequencies. As the frequencies increase, the IW produces a frequency-division-like separation of up and down transmissions because of the stronger crosstalk at those frequencies.

**Brady’s Worst-Case Distributed-Management IC Interference:** While the IC capacity region is an outer bound for the performance, and therefore also on the distributed-management IC, an interesting question is a worst-case bound, namely one in which the users are hostile to one another and mutually cost largest degradation. A version of such a worst-case would then be prudent in design of a DCIC if possible.

M. Brady has developed an approach to determining the worst-case LIC noise. Such a worst-case problem is somewhat ill-posed in that all users could simply be obnoxiously impolite transmitting as much power as they could everywhere. However, under a power constraint for the other $U - 1$ users individually, the problem shifts to a game in which user $u$ tries to maximize their data rate, while all the others try to minimize this same data rate of user $u$ with offensive choices of power spectra that however must satisfy individual power constraints. The other users employ a strategy to reduce this one victim user’s data rate and thus create a “worst case.” Mathematically, this problem is written as

$$
\max_{R_{xx}(u)} \min_{R_{xx}(i \neq u)} \log_2 \frac{|H_{uu} \cdot R_{xx}(u) \cdot H_{uu}^* + R_{\text{noise}}(u)|}{|R_{\text{noise}}(u)|},
$$

$$
ST : \quad (5.457)
$$

$$
975
$$
where as usual
\[ R_{\text{noise}}(u) = \sum_{i \neq u} H_{ui} \cdot R_{xx}(i) \cdot H_{ui}^* + R_{nn}(u) \quad . \] (5.459)

The constraints are that the user autocorrelations are valid (positive semi-definite) matrices with bounded traces (or sum traces when formulated in frequency domain for set of tones) by the energy constraints. Power spectral density constraints may also be applied to each tone in “loading” for the worst-case interference.

The analysis of Brady’s method is somewhat involved and beyond the scope [?]. Nonetheless, a Matlab subroutine that solves the above problem (all users have \( L_x = L_y = 1 \)) exits.

The program has 4 inputs (with user 1 considered to be the victim and all others to be the offenders):

1. \( H \) the \( U \times U \times N \) tensor of all the channel responses
2. \( P \) a \( U \times 1 \) vector of individual energy per-user constraints
3. \( \Sigma \) is the \( U \times 1 \) vector of noise variances
4. \( \text{Gap} \) is the linear scale gap

and 4 outputs

1. \( R \) is the number of bits per symbol for the victim user 1
2. \( Y \) is the \((U - 1) \times N\) matrix of PSDs of the offending modems
3. \( X \) is victim user 1’s power spectral density
4. \( \text{int. profile} \) is the sum of offending users’ channel output spectra into user 1.

The program provided by Mark Brady is described here and the program listing is in Appendix G:
function [Rate Y X int_profile]=wci(H,P,Sigma,Gap)

%function [Rate Y X int_profile]=wci(H,P,Sigma,Gap)
%Compute the worst-case interference for IC user 1
%Inputs: H is a UxUxN matrix of channel gains
% where H(m,p,n) is channel from user p into user m on tone n
% P is a Ux1 vector of power constraints.
% Sigma is a Nx1 vector of AWGN noises (per tone noise)
% Gap is the Gap-to-Capacity in LINEAR scale (not dB)
%Outputs: Rate is the guaranteeable rate under WCI
% Y is the worst-interference inducing power allocations
% X is the victim modem response to the worst-interference
% int_profile is WCI interference profile
%Restrictions: *H of user 1 should not be zero for all tones
% *H must have at least 2 users
% *P must have each element strictly positive
%rev mhbrady 9/23/05

%Test the input dimensionality

5.6.3 Dynamic Spectrum Allocation (DSA)

DSA is a form of consensus management. It allows central spectrum-policy guidance for the distributed-management IC, slightly violating the constraint of no central control, but not allowing the specification of receiver coefficients, channel $H_{ij}$ sharing, or other more coordinated IC management. Subsection 5.6.3.1 introduces\textsuperscript{81} “Optimal” spectrum balancing (OSB), a method that is optimum for a synchronized IC-DMT-user set and can calculate best bounds for policy-based DSA. OSB’s complexity is exponentially high and it is not feasible for field use. Iterative Spectrum Balancing (ISB) replaces the OSB’s intensive step with an approximation as in Subsection 5.6.3.2 and often produces at a lesser (but still high) complexity a result. Papandriopoulos’ SCALE method for the IC reduces complexity to a level comparable to the MAC’s minPMAC program. They the SCALE approximations provide a nice analogy with iterative water-filling, lead ultimately to the iterative multi-level water-filling methods of Section 5.6.3.3 that will obtain essentially the highest level of performance at a cost essentially no greater than that of IW and with a highly distributed implementation (although a very small amount of central policy appears, which is the message-passing of consensus-based management).

5.6.3.1 Optimum Spectrum Balancing (OSB)

OSB presumes a DMT user system that synchronizes all users’ symbol boundaries\textsuperscript{82}. OSB maximizes a weighted rate sum with $\mathbf{w} \succ 0$ and traces an achievable region, when each IC receiver has access only to its own signal. However, all the users’ spectra are calculated centrally under the assumption that no user has a GDFE - that is no successive decoding. The name “optimum” is probably a misnomer in that the system is not overall optimum for an IC, but is optimum under certain restrictions. OSB otherwise is a theoretical concept that helps develop more practical consensus-management approaches.

The “optimization” is\textsuperscript{83}

$$\max_{\{R_{xx}(u,n)\}} \sum_{u=1}^{U} \theta_u \cdot b_u$$

ST : \begin{align*}
0 \leq \sum_n \text{trace} \{R_{xx}(u,n)\} &\leq C_{u_{\text{max}}} u = 1, \ldots, U \end{align*} (5.461)

\textsuperscript{81}Introduced by Dr. Raphael Cendrillon and Wei Yu when Raphael was a visiting student at Stanford in 2003 and Wei Yu was a PhD student at Stanford [3].

\textsuperscript{82}Such synchronization may use GPS timing references outdoors or some other common or learned network clock source generally.

\textsuperscript{83}This problem is the same as maximizing one users’ rate while all others are each lower-bounded at some desired rate.
The presumed relationship between user bit distribution and autocorrelation matrices is, specifically noting that all users crosstalk is included in distortion and there is no successive decoding nor GDFE of any type,
\[ b_u = \sum_n \log_2 \left( \frac{|H_{uu,n} \cdot R_{xx}(u, n) \cdot H_{uu,n}^* + R_{noise}(u, n)|}{|R_{noise}(u, n)|} \right). \] (5.462)

This OSB derivation uses \( L_x = L_y = 1 \). Later simplified algorithms generalize to larger numbers of antennas. Thus, \( R_{xx}(u, n) \to E_{u,n} \). The Lagrangian forms as a tonal sum:
\[ L_n(R_{xx}(u, n), b_u, w, \theta) = \sum_n w_u \cdot E_{u,n} \cdot \theta_u \cdot b_{u,n}. \] (5.463)

The overall Lagrangian is, with user energy constraints being the diagonal elements of \( E \),
\[ L = \sum_{u=1}^U \left[ \sum_n L_n \right] - w_u \cdot E_u. \] (5.464)

The Lagrangian problem is not convex because each user depends on all other users’ spectra. However, it does have a solution. The achievable rate region, with no successive decoding, is the convex hull for \( \sum_{u=1}^U \theta_u = 1 \) with \( \theta \succeq 0 \) and each’s implied constraint \( w \succeq 0 \). Since there are no GDFE receivers, there is no decoder order \( \pi \). Each user’s bits/symbol can include a gap (with \( SNR_{u,n} \Delta = |H_{uu,n}|^2 \cdot E_{u,n} \))
\[ b_u = \sum_n \log_2 \left( 1 + \frac{\text{SNR}(u, n)}{\Gamma} \right). \] (5.465)

Since each user employs its own separate code that treats all others as noise, then the gap approximation directly applies.

The same Lagrangian also applies with fixed user data rates to the energy-sum minimization where the trailing term in (5.464) becomes \( \theta_u \cdot b_u \) instead of \(-w_u \cdot E(u)\). Equivalently, the doubly constrained problem can be checked for energy-constraint satisfaction at any given \( b \) (the admission problem). This check can be used to generate a new \( \theta \), which then can be subsequently used again in the original problem. Thus the algorithm has two steps:

**energy step** minimize \( L_n \) for fixed \( w \) and \( \theta \) by using the known capacity relation in (5.465).

**constraint step** Optimize using sub-gradient descent (or the elliptical method) for the value of \( w \) (or both \( \theta \) and \( w \)) in the admission-problem context.

The second step mirrors the vector MAC’s order step\(^{85}\), uses the same (elliptical or sub-gradient) algorithms. The first step differs because of the \( E_{u,n} \) interdependencies without successive decoding. This step instead exhaustively searches all users’ possible energy settings.

The first step evaluates all energy increments up to a maximum:
\[ M = \max_u \frac{\mathcal{E}(u)}{\Delta \mathcal{E}} \] (5.466)
for some energy search increment \( \Delta \mathcal{E} \); then this step evaluates \( M^U \) values of \( L_n \) for each tone. This is a high complexity for more than 2 or 3 users. The actual complexity has order \( O(NUM^U) \) because each \( R_{noise}(u, n) \) calculation itself requires \( U \) operations.

The second step can use sub-gradients (let \( \mathcal{E}_{\max} \) be a vector of the diagonal elements of \( \mathcal{E} \) and \( \mathcal{E}_n \) the vector of energies for the users on any tone \( n \))
\[ \Delta b = b_{\min} - \sum_n b_n \] (5.467)
\[ \Delta \mathcal{E} = \mathcal{E}_{\max} - \sum_n \mathcal{E}_n. \] (5.468)

\(^{84}\) Thus, \( \theta \) does not in the OSB case determine an order.

\(^{85}\) Although the \( \theta \) element relation no longer reflects a decoding order, but such could be an interesting investigation even if OSB presumes no GDFE use.
These sub-gradients permit a direct update of the Lagrange multipliers according to

\[ \theta \leftarrow \theta + \epsilon \cdot \Delta b \]  

\[ w \leftarrow w + \epsilon' \cdot \Delta E. \]

Typically, a more sophisticated elliptical procedure is necessary for reasonable convergence in practice, where the initial condition for \( \theta \) (and \( w \) in the admission problem) is in the first orthant.

OSB really is a suboptimum central management system, but focuses only on spectrum setting (and not maximums with best decoders and signal processing/modulation). In so narrowing focus, however, OSB paves a path to simpler methods that ultimately can be largely distributed in implementation and will be effectively optimum for such systems.

### 5.6.3.2 Iterative Spectrum Balancing (ISB)

Iterative Spectrum Balancing (ISB) was simultaneously introduced by Yu and Liu [2]. ISB replaces OSB’s exhaustive energy-search step by an iterative approximate algorithm. The approximation optimizes each user separately in a sub step where \( M \) values of energy for that user are compared in terms of \( L_n \) values while the energies for all other users are held constant. The algorithm cycles through all users (each holding all the others constant). Convergence is assured because the method reduces \( L_n \) at each step, and usually in far less than \( M^U \) steps. Thus, the complexity reduces from \( O(NU^2M) \) in OSB to \( O(NU^2M) \). By comparison, IW requires \( O(NU) \) and thus is still considerably less complex yet than ISB, which however is much less complex than OSB and approximates it. However ISB is approximating a central-management system that was already suboptimum (despite its name).

Papandriopoulos introduced another method called SCALE that allows a distributed implementation with message passing, so more closely approximating consensus management. It approximates the performance of OSB.

Observation in practice of OSB and its approximations suggest that the best spectra of all users look like water-filling, but in separate bands with different water levels. This makes a strong intuitive theoretical sense since there is no successive decoding and indeed a designer would expect simultaneous water-filling appearance if all other users are noise. However, the water-levels are different in different frequency bands, or more generally across different dimensional sets or clusters.

### 5.6.3.3 Multi-Level (Iterative) Water Filling

The multi-level iterative water-filling introduced first in this text simply observes that water-filling indeed would appear optimum in certain clusters of dimensions between all users energizing those dimensions. The water-levels can vary, and the clusters of dimensions (frequency bands when there is only one antenna per receiver) can vary, so MLIWF focuses on finding the levels and the clusters (which are cut-off frequencies between the different bands in the single-antenna-receiver case). Summarizing the realization difficulties with all the methods, except iterative water-filling:

1. OSB and ISB require central spectra control and synchronization (as with the MAC or BC) so that crosstalk only occurs between systems independently on each tone. Such synchronization is unlikely and thus the crosstalk will be a function of other adjacent tones (and the spectra of unsynchronized tones only falls as \( 1/f \) so the other tones’ crosstalk will be significant – even windowing can only reduce this effect slightly as in Section 4.9).

2. In unsynchronized systems, the power-spectral density roll-off prevents very large reductions in power-spectral density on adjacent tones, which is exactly the type of spectra that OSB methods typically produce.

These difficulties encourage the development of more distributed autonomous spectral balancing among the multiple users.
5.6.3.4 The Multi-Level IW method

Ultimately, the key DSA parameter is a separation or “cut-off” frequency $f_{\text{cut},u}$ between a band of tones in which loading to the maximum level is desired more for politeness and other bands that are preferred less for politeness. Figure 5.56 illustrates the basic concept where a control unit informs users that can easily meet their rate targets are registered and informed to be extra polite. In effect ML IW imposes a maximum (flat) PSD level (which is not quite water-fill but often close as in Chapter 4’s discussion around the Separation Theorem) at two levels $S_{\text{low},u}$ and $S_{\text{high},u}$. In effect, each user water-fills in two bands with different water levels. Users that have no ability to be polite are not so asked to be extra polite (and so execute water-filling with one level).

An extra-polite user might have two water-filling bands (the extension to more than two bands is trivial):

$$\lambda_{u,1} = \mathcal{E}_{u,n} + \frac{\Gamma}{g_{u,n}} \forall n \in \mathcal{F}_1$$

(5.471)

$$\lambda_{u,2} = \mathcal{E}_{u,n} + \frac{\Gamma}{g_{u,n}} \forall n \in \mathcal{F}_2$$

(5.472)

The DSA regulator determines two bands $\mathcal{F}_1$ and $\mathcal{F}_2$ and tells a (or some) user(s) to be extra polite, on the $S_{\text{low}}$ side of its cut-off frequency. The local user transmitter itself determines the two water-filling levels $\lambda_{u,1}$ and $\lambda_{u,2}$ and includes any specified power-spectral density limits. The band with the higher
index 2 is preferred for loading and its water-level is determined by first solving an overall water-filling problem (with one level), and then moving bits one-by-one from band 1 to band 2 until any power spectrum density limit constraints in band 2 would be violated (or until the overall power constraint would be exceeded). This preferred band then effectively water-fills (approximately to the degree that LC approximates water-filling) with a higher water-filling level. The polite band also water-fills but to a lower water-filling level. The energy (and information/coding with gap \( \Gamma_u \)) allocation continues until the \( b_u \) and any margin \( \gamma_u \) is achieved. If the system must use the polite band above the requested PSD, then it does and the DSA regulator observes this on a next pass. The cut-off frequency is important. Minimal central control occurs, because the controller would run the multi-level IW for each user for various estimates of the cut-off frequency (or exhaustive search) and then recommends only each user’s cut-off accordingly. The users otherwise retain full autonomy. If noises change such that two different water levels cannot be maintained, then the local user strategy returns normal water-filling.

The concept easily extends to 3 or more bands with the most preferred band being filled first, then the next highest index and so forth. A simple method to communicate preferences simply sends slight differences in the power-spectral density (PSD) mask to all users (via central control or even by standards, as in DSA). A change in PSD mask from one tone \( n \) to \( n + 1 \) simply implies that the preference index changes and the band with higher PSD mask should have a higher index. Such a method leaves all bands water-filling (and since we know IW converges under wide conditions, then this ML IW will also converge in those same situations). Inspection of the final results for power spectra in Section 5.6.3 certainly confirms that a simple cut-off could easily have been used to obtain the same results with IW. Thus, the only central control required is an indication to the user to load extra polite favoring those bands for which its power spectrum density limit is higher. The power spectrum density limit can be preset in all modems or might possibly have been distributed in a quasi-static upon-start type initialization within the network.

5.6.3.5 ML IW examples and results

The key advantages of the ML IW approach are:

1. distributed low-complexity implementation. There are consequently also no convergence issues, choice of thresholds, choice of elliptic versus sub-gradient, etc complexities even at the controller if one exists.

2. The individual user modems retain the ability to react to changes in the noise or channel. Thus, rapid direct reaction to changes allows robust operation in the presence of any kind of situation not originally anticipated by the controller and/or modems.

3. OSB’s tacit “all-are-synchronized” assumption is no longer necessary. The modems react according to the actual noise present and not some presumed synchronized-crosstalk presumption in an optimization algorithm.

4. In the next example, the performance of ML IW matches OSB.

5. If \( L > U \), it’s optimal.

A few examples will illustrate the advantages.

EXAMPLE 5.6.3 (xDSLs) A more generic example will replace this later. A two-user DSL simulation is appears in Figure 5.57. A noise floor of -140dBm/Hz models background noise. Four RT (remote terminal) DSL signals constitute a strong interference to user 2.
Figure 5.57: ADSL mixed binder IC example.

Figure 5.58 provides the optimal spectrum levels for this channel. From this example, it is clear that the masks are very sharp in transition and nearly impossible for implementation. They also illustrate the basic concentration of short-line energy at higher frequencies. The crosstalk transfer function for the channels into one another used here was

\[
| H_{ij}(f) | = 9 \times 10^{-20} \cdot \left( \frac{1}{49} \right)^{0.6 \cdot l \cdot f^2} \cdot | H_{ii}(f) |^2 ,
\]

(5.473)

(where \( l \) is the channel length of the line in 1000’s of feet, with one meter equaling 3.28 feet) which increases with frequency, so all lines would experience dramatic crosstalk at higher frequencies in addition to the typical attenuation with frequency of the lines. Nonetheless, the shorter lines best occupy the very lowest frequencies where the crosstalk is low and the very highest frequencies where they do not inflict harm on the high frequencies that could not be used by the long line.
By contrast, the use of ML IW would place 3 bands, low, medium, and high where low and high can have the same power spectrum density limit levels and the medium band should have slightly lower power spectrum density limit. The low and high bands are then preferred on the 4 short loops. The cut-offs are roughly 300 kHz and 700 kHz.

Figure 5.59 shows the ML IW and OSB achievable rate regions for the situation in Figure 5.57. These two regions are the two largest, which are virtually equal. IW with no power spectrum density limits also appears for reference. Each of the intermediate curves corresponding to relaxing the OSB power spectrum density limit in Figure 5.58 by successive increments of 6 dB to allow for both robustness to channel change and possibility of implementation with less sharp transition bands. ML IW has a power spectrum density limit in the middle band that is just 1 to 2 dB lower than the low and high bands, and thus is very feasible. These show that reasonable relaxation of the central control is highly sensitive. However, the ML IW works without need for such sensitivity.
Figure 5.59: Rate Regions with various levels of tolerance on the OSB power spectrum density limits.
5.7 Statistically Characterized GDFEs

not yet added.
Exercises - Chapter 5

5.1 channel singularity elimination (8 pts)
(Subsection 5.1.1.1) Given a binary antipodal (2 PAM) white input with $\bar{E}_x = 1$ transmitted through the channel with $H$ matrix

$$H = \begin{bmatrix} .9 & 1 & 0 & 0 \\ 0 & .9 & 1 & 0 \\ 0 & 0 & .9 & 1 \end{bmatrix},$$

(5.474)

and with the input-modulator-matrix choice of $C = I$:

a. (2 pts) Find the projection matrix $\tilde{P}_M$.

b. (2 pts) Determine the $4 \times 3$ matrix $\tilde{C}$ and the input $\tilde{u}$ that characterize the pass-space description of this channel.

c. (2 pts) Determine the autocorrelation matrix $R_{\tilde{x}\tilde{x}}$.

d. (2 pts) How much energy is lost into the null space of this channel?

5.2 Singularity Elimination short problems. (12 pts)
(Subsection 5.1.1.1) For each of the following channels and inputs, determine $N$, $\tilde{N}$, $N^*$, and $\nu$.

a. (2 pts) $H = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$ and $R_{xx} = I$.

b. (2 pts) $H = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$ and $R_{xx} = I$.

c. (2 pts) $H = \begin{bmatrix} 5 & 6 & 7 & 8 \\ 1 & 3 & 2 & 4 \\ 4 & 3 & 5 & 4 \end{bmatrix}$ and $R_{xx} = I$.

d. (2 pts) $H = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$ and $R_{xx} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$.

e. (2 pts) $H = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$ and $R_{xx} = \begin{bmatrix} 1 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$.

f. (2 pts) $H = \begin{bmatrix} 5 & 6 & 7 & 8 \\ 1 & 3 & 2 & 4 \\ 4 & 3 & 5 & 4 \end{bmatrix}$ and $R_{xx} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$.

5.3 MMSE Estimation (8 pts)
(Subsection 5.1.2) Two zero-mean complex Gaussian random variables have probability distributions $p_x$ and $p_y$ with joint probability distribution $p_{x,y}$ and nonsingular autocorrelation matrix

$$R_{x,y} = \begin{bmatrix} \mathcal{E}_x & r_{xy} \\ r_{xy}^* & \mathcal{E}_y \end{bmatrix}$$

(5.475)

The minimum mean-square estimate of $x$ given $y$ has variance $\sigma^2_{\hat{x}/y}$. The orthogonality principle of Appendix 3.A may be useful throughout.
a. (1 pt) Find $\sigma^2_{x/y}$ in terms of $\mathcal{E}_x$, $r_{xy}$ and $\mathcal{E}_y$.

b. (1 pt) Relate the conditional entropy $H_{x/y}$ to $\sigma^2_{x/y}$ and therefore to the MMSE estimate.

c. (2 pts) Interchange the roles of $x$ and $y$ in the results in Parts a and b and compare the SNRs for the two results.

d. (1 pt) Relate the mutual information to the SNR of Part c.

e. (3 pts) Suppose $y$ becomes a complex vector $y$ but $x$ remains a scalar with $y = \begin{bmatrix} 1 \\ 1 \end{bmatrix} x + n$ where 

$n$ has independent zero-mean Gaussian components all with variance $\sigma^2$. Does your answer to part d change? How might you interpret SNR$x$ in this case in terms of equalizers you know from Chapter 3?

5.4 DFE - MT connection. (8 pts)

(Subsection 5.2.3 ; Section 2.5 may also be helpful, as well as Subsection 3.12.1) This problem explores the connection between DFE and multi-tone-like concepts for transmitter optimization, essentially extrapolating the Circulant DFE’s limit to Chapter 3’s MMSE-DFE and DMT’s limit to MT.

a. (4 pts) Show that the following are all equivalent expressions for $\mathcal{I}$ for a fixed sampling period $1/T$ equal to the $1/N$ times a limiting infinite-length symbol period. (Complex baseband is assumed so the extra factor of $1/2$ in front of the log does not appear.) $\bar{N}$ can be assumed to always be an even integer.

$$
\bar{\mathcal{I}} = \frac{T}{2\pi} \int_{-\frac{\pi}{T}}^{\frac{\pi}{T}} \log_2(1 + \text{SNR}(\omega)) \cdot d\omega = \frac{T}{2\pi} \int_{-\frac{\pi}{T}}^{\frac{\pi}{T}} \log_2(1 + \text{SNR}_MFB \cdot |Q(e^{-j\omega T})|) \cdot d\omega
$$

$$
= \frac{T}{2\pi} \int_{-\frac{\pi}{T}}^{\frac{\pi}{T}} \log_2(1 + \text{SNR} \cdot |H(e^{-j\omega T})|^2 \cdot |\Phi(e^{-j\omega T})|^2) \cdot d\omega
$$

where $H(e^{-j\omega T})$ is the transform of the sampled channel. This problem assumes that the sampling rate is high enough so that there is no aliasing. Relate these formulas to each of MT and the MMSE-DFE.

b. (4 pts) The integral in Part a can be approximated by the following sum for large $\bar{N}$:

$$
\bar{\mathcal{I}} = \frac{T}{2\bar{N}T} \sum_{-\bar{N}/2+1}^{\bar{N}/2} \log_2 \left[ 1 + \text{SNR} \cdot \left| H(e^{j2\pi n/2\bar{N}T}) \right|^2 \cdot \left| \Phi(e^{j2\pi n/2\bar{N}T}) \right|^2 \right]
$$

Optimize the above approximation and show that the water-fill equations for this problem are:

$$
\frac{1}{\text{SNR} \cdot |H(e^{j2\pi n/2\bar{N}T})|^2} + \left| \Phi(e^{j2\pi n/2\bar{N}T}) \right|^2 = \text{a constant}
$$

5.5 Input Singularity Elimination. (7 pts)

(Subsection 5.1.1.2) The real baseband channel input autocorrelation after elimination of this channel’s singular components and design of the input autocorrelation is

$$
R_{\tilde{x}\tilde{x}} = \begin{bmatrix} 1 & .5 & 1 \\ .5 & 1 & .5 \\ 1 & .5 & 1 \end{bmatrix} \quad \text{(5.476)}
$$

987
a. (1 pts) What is $N^*$ for this channel input? Is $\bar{N}^* = N^*$ on this channel?

b. (2 pts) Find $\tilde{Q}$ and $P_{\tilde{Q}}$.

c. (4 pts) Find at least two $A$ matrices that accept $N^*$-dimensional (white, so $R_{vv} = I$) input $v$ such that the output autocorrelation is $R_{\hat{x}\hat{x}} = A \cdot A^*$. In each case, relate the components of $v$ to those of $\hat{x}$.

5.6 Complex Baseband Canonical Triangles (13 pts)

(Subsection 5.1.2) This problem illustrates easy derivation of a number of important mathematical relationships in the GDFE through the canonical triangles associated with the canonical forward and backward channel models, which are found in Figure 5.7.

a. (2 pts) The forward channel in Figure 5.7 has two right subtriangles embedded within it. From the one describing a decomposition of the noise vector $n'$, one notes that the MMSE estimation filter for $n'$ given $z$ is $R_{n'z}^{-1} R_{zz} = (I - R_f R_b)$. Use this relation, and note that $R_{n'z} = R_{n'n} = R_f$, to prove that $R_f^{-1} = R_b + R_{zz}^{-1}$. 

b. (2 pts) Use the triangle for decomposition of the error vector in the backward triangle to note that $R_b = -R_{en} R_{n'n}^{-1}$ to prove easily that $R_{ee} = R_b$.

c. (2 pts) Follow parts a and b to prove that $R_b^{-1} = R_f + I$.

d. (2 pts) Use a convenient triangle to show that $R_{en}^* = -R_f R_b$.

e. (2 pts) Prove that $R_b R_f = (I - R_b)$.

f. (3 pts) The matrix $SNR$ for a matrix MMSE-LE is $SNR = R_{vv} \cdot R_{ee}^{-1}$. Use the Pythagorean relationship $I = R_b \cdot R_{zz} \cdot R_b + R_{ee}$ to prove the “similar” triangles relationship $SNR$ is also equal to $R_{n'n}^{-1} R_{zz}$.

5.7 Complex Baseband Matrix Bias. (8 pts)

(Subsection 5.1.2) For the matrix additive Gaussian noise channel, the relationships for bias in the matrix MMSE-LE and the GDFE generalize easily from their scalar counterparts. Let $\hat{v} = R_b \cdot z$ and $SNR = R_{ee}^{-1}$. The forward and backward are as in Section 5.1.

a. (1 pt) Show that $E[\hat{v}/v] = R_b \cdot R_f \cdot v \neq v$ unless $v = 0$.

b. (1 pt) Show that $E[e/v] = (I - R_b \cdot R_f) \cdot v$.

c. (2 pts) Show that unbiased matrix $SNR$ defined as $SNR_{unb} \triangleq R_{v\hat{v}} \cdot R_{ee}^{-1}$ is equal to $SNR - I$.

d. (2 pts) Show that another expression for the unbiased matrix $SNR$ is $SNR_{unb} = R_{n'n}^{-1} R_{zz}$ where $\hat{z} = R_f v$.

e. (2 pts) Show that $\mathcal{I}(v; z) = \log_2 |SNR|$ for complex signals.

5.8 Non-white and white inputs to the $1 + .9D^{-1}$ channel. (16 pts)

From Example 5.1.2, assume that the $N + \nu$-dimensional binary PAM input symbol possibilities are mutually independent and the eight possible 3-dimensional channel-input vectors are $x = [\pm 1, \pm 1, \pm 1]$. The noise-equivalent channel is

$$
\frac{1}{\sigma} \cdot \vec{H} = \frac{1}{\sqrt{1.81}} \cdot \begin{bmatrix} .9 & 1 & 0 \\ 0 & .9 & 1 \end{bmatrix} = \begin{bmatrix} 2.1155 & 2.3505 & 0 \\ 0 & 2.1155 & 2.3505 \end{bmatrix},
$$

with singular value decomposition:

$$
\frac{1}{\sqrt{2}} \cdot \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} = \begin{bmatrix} 3.8694 & 0 & 0 \\ 0 & 2.2422 & 0 \end{bmatrix} \begin{bmatrix} .3866 & -.6671 & -.6368 \\ .8161 & -.0741 & .5731 \\ .4295 & .7412 & -.5158 \end{bmatrix}^*.
$$

(5.477)
The first two columns of $M$, or $\tilde{M}$, span the pass space of the channel, so $P_{\tilde{M}} = \tilde{M} \cdot \tilde{M}^\ast$. The original PAM modulator corresponds to $C = I$, so

$$
\tilde{C}_{\text{temp}} = P_{\tilde{M}} \cdot C = P_{\tilde{M}} = \begin{bmatrix}
0.5945 & 0.3649 & -0.3285 \\
0.3649 & 0.6715 & 0.2956 \\
-0.3285 & 0.2956 & 0.7340
\end{bmatrix},
$$

(5.478)

which in this special case of white PAM modulation corresponds to a rank-2

$$
R_{\tilde{x}\tilde{x}} = P_{\tilde{M}} \cdot I \cdot P_{\tilde{M}} = \tilde{C}_{\text{temp}}.
$$

(5.479)

a. (4 pts) Complete the steps in Figure 5.4 for this input to find $R_{uu}$.

b. (4 pts) Compute a formula for the channel inputs $\tilde{x} = \tilde{C} \cdot u$ using the $O$ matrix from Part a. Use it to compute the 8 $u$ and associated $\tilde{x}$ values that correspond to the 8 possible $\pm 1$ 2-PAM inputs? Are the equal? Why or Why not?

c. (2 pts) How much of $R_{\tilde{x}\tilde{x}}$’s energy is lost into null space?

d. (2 pts) Continuing, what are the 8 possible noise-free channel outputs?

e. (2 pts) What kind of detector would achieve lowest $P_e$ for this (backward canonical) channel? Please provide detector-type answers for both uncoded 2-PAM and a more general description for the presumed use of a zero-gap code.

f. (2 pts) Compare $2^{2\mathcal{E}(x,y)}$ to $\text{SNR}_{\text{GDFE}}$ for some $v$ that whitens this $u$. What can you say about error probability if the original 2-PAM $x$ is used in this GDFE?

5.9 GDFE’s and White Inputs. (15 pts)

(Subsection 5.1.3) A real baseband linear-ISI channel with discrete-time response $1 + D$ has AWGN with $\frac{N_0}{2} = 0.001$. The transmission scheme is real baseband and initially uses 8PAM inputs on all dimensions that are independent with $\overline{E}_x = 1$.

a. (4 pts) For $N = 3$ and $\nu = 1$, find the CDFE on this channel, including $G$, feedforward filter matrix, $\text{SNR}_{\text{cdf.e,u}}$ and $\overline{P}_e$ with equal-energy 16-level PAM on each of the 3 used subchannels’ dimensions. For this problem, the geometric SNR can be considered the same on all subchannels, although that is only true asymptotically with CDFE. (Error propagation may be ignored.)

b. (2 pts) Compare Part a’s error probability with the asymptotic error probability for an 8 PAM MMSE-DFE with $N_f = 3$ and $N_b = 1$ on this same channel. The program dfecolorsnr.m from Chapter 3 may be helpful. Comment on why the latter appears better.

c. (4 pts) Repeat Part a for a GDFE (with a guard band of $\nu = 1$ but not necessarily a cyclic prefix) that instead places 4/3 units of energy on each of the 3 useful input dimensions. Why does this work better than Part a? Compare with part b and comment as to which system is actually better.

d. (5 pts) Repeat Part c for $N$ increasing to 10, 20, 100, 200 except only compute the unbiased SNR (and not all the matrices). Estimate to which value this SNR converges as $N$ becomes infinite. What is the number of levels of PAM that would be used as $N$ increases at $\overline{P}_e = 10^{-6}$. Which system in this problem is best and why?

5.10 Comparing the GDFE and the MMSE-DFE. (14 pts)

(Subsection 5.1.3) For the $1 + 0.9 \cdot D^{-1}$ with $\text{SNR}_{\text{mfb}} = 10$ dB, the GDFE performs worse than the MMSE-DFE for $N = 2$ and $\nu = 1$. One might conjecture that this is caused by a large $(\nu/(N+\nu) = 1/3)$ bandwidth loss due to too small $N$. This problem investigates increasing $N$ and assumes the student has access to the dfecolor.m Matlab program.

Initially let $R_{xx} = I$ and let $N$ increase.
a. (5 pts) Use Matlab to compute the GDFE performance for variable $N$ and $\nu = 1$. Plot $SNR_{gdf,e,u}$ versus $N$ for $N = 2, \ldots, 40$. (Recall $SNR_{gdf,e} = 2^2 \|X,Y\|$). Is there an easy way to compute $\|X,Y\|$?, and think of vector coding)

b. (2 pts) Repeat Part a for Chapter 3’s MMSE-DFE and plot $SNR_{mmsedfe,u}$ versus $N$ with the number of feedback taps fixed at $\nu = 1$. Compare with part a results by plotting on the same graph.

c. (2 pts) Plot Vector Coding’s performance with waterfilling, which equals the performance of the GDFE with optimized input, for $N = 2, \ldots, 40$ with $\nu = 1$ and compare to parts (a) and (b).

d. (5 pts) Draw conclusions from your plots in parts (a), (b), (c). This part has a large number of points for a reason, so perhaps think about any initial transmission-system conditions in terms of what was being plotted.

5.11 Cyclic Antennas (7 pts)
(Subsection 5.2.3)

Figure 5.11’s 6-real-dimensional cyclic channel has $H = \begin{bmatrix} 1 & .5 & .5 \\ .5 & 1 & .5 \\ .5 & .5 & 1 \end{bmatrix}$ with noise variance (2-sided power spectral density) equal to .01 and a gap of 0 dB, the energy per real dimension is 1. Hint: Cholesky factorization of a matrix that is nearly white is trivial.
a. (2 pts) Find a water-filling input covariance matrix \( R_{xx} \). Also find the corresponding Cholesky factorization \( R_{xx} = G_x \cdot S_x \cdot G_x^* \).

b. (2 pts) Find the forward and backward canonical-channel-characterizing matrices \( R_f \) and \( R_b^{-1} \).

c. (2 pts) Find the feedback matrix \( G \) and the combined matched-filter feedforward filter matrix.

d. (1 pt) What is the SNR of this CDFE-equalized channel and what is the corresponding largest number of bits per dimension that can be transmitted at \( P_e = 10^{-6} \)?

5.12 MAC-like Vector DMT (9 pts)

(Section 5.2.1) Two transmission lines share a common cable and are not shielded with respect to one another and so thus experience crosstalk as well as ISI, as in Figure 5.61. The two transmitters are not co-located, but each uses DMT with the same clocks and cyclic-prefix lengths - that is the symbols are aligned on the two channels. The inputs to the transmit IFFT’s are \( X_{1,n} \) and \( X_{2,n} \), respectively, and each such component is independent of all others (including all frequency indices on the other input). The input energy per dimension for each tone of these two transmitters on each such input is denoted \( E_{1,n} \) and \( E_{2,n} \). The two outputs are received in the same box at the end of the cable, so that coordinated signal processing/detection of the two input streams (as if they were one) is possible. The discrete-time matrix impulse response of the channel is

\[
\mathbf{h}(D) = \begin{bmatrix}
h_{11,k} & h_{12,k} \\
h_{21,k} & h_{22,k}
\end{bmatrix}
\]  

\( (5.480) \)

where each entry is an FIR channel and

\[
h_{ij,0} + h_{ij,1} \cdot D + \ldots + h_{ij,\nu_{ij}} \cdot D^{\nu_{ij}}.
\]  

\( (5.481) \)

and \( h_{ij,0} \neq 0 \) and \( h_{ij,\nu_{ij}} \neq 0 \). The channel output is thus

\[
y(D) = \begin{bmatrix}
y_1(D) \\
y_2(D)
\end{bmatrix} = \mathbf{h}(D) \begin{bmatrix}
x_1(D) \\
x_2(D)
\end{bmatrix} + \mathbf{n}(D),
\]  

\( (5.482) \)

where \( R_{nn}(D) = \sigma^2 \cdot I \).

a. (1 pt) What is the shortest-length cyclic prefix that can be used to ensure there is no intersymbol interference on or between any of the discrete-time paths?

b. (2 pts) Write a relationship for each DMT-output (i.e. tone \( n \) for both of the outputs \( Y_{1,n} \) and \( Y_{2,n} \)) of a receiver in terms of ALL the tone inputs to both channels. What is the dimension of the channels created?
c. (4 pts) For the resultant channels in part b, determine GDFE settings in terms of the given quantities above and $H_n$ and/or its elements? What happens if any of the inputs have zero energy on a given frequency? You should use the defined quantities $r_{1,n} = |H_{11,n}|^2 + |H_{21,n}|^2$, $r_{2,n} = |H_{22,n}|^2 + |H_{12,n}|^2$, and $r_{12,n} = H_{11,n}^* H_{21,n} + H_{21,n}^* H_{22,n}$ to simplify derived expressions. The total feedforward-filter/matched-filter may be expressed as a product of 3 matrices, and not simplified further.

d. (2 pts) If the two inputs have mutual information $I_1(x_1; y)$ and $I_2(x_2; y/x_1)$, what is the overall SNR of the DMT/GDFE (bonded DMT) system?

5.13 Generalized Cholesky. (10 pts)

(Subsection 5.2.4.1) Given the discrete-time channel $H(D) = 1 + .8 \cdot D$ with $\bar{E}_x = 1$ and $N_0^2 = .164$ and a gap of 0 dB:

a. (4 pts) Find the best $R_{xx}$ and corresponding SNR$_{gdfe,u}$ for $N = 4$ and $\nu = 1$.

b. (5 pts) Compute all the matrices in the GDFE for your answer in part a if the designer is allowed to construct a white input and wants a triangular or “causal” implementation. Provide a block diagram of both transmitter and receiver.

c. (1 pt) Why can’t a CDFE or DMT obtain quite this same performance with $N = 4$ and $\nu = 1$?

5.14 Some review questions. (6 pts)

(Section 5.3)

a. (1 pt) For what types of $R_{xx}$ can a CDFE be implemented? Equivalently, for what types of channel matrices $H$?

b. (1 pt) For what types of $R_{xx}$ can a GDFE be implemented? Equivalently, what are the restrictions on $H$?

c. (1 pt) What is the characteristic that defines a “canonical channel”?

d. (1 pt) What is the difference between “canonical factorization” and a “canonical channel”?

e. (1 pt) Why does the appropriate use of the CDEF result allow suboptimum equalizers to play at highest performance levels?

f. (1 pt) Generally on a channel with a prior unknown or severe intersymbol interference, if you had the option to implement DMT or a CDFE/MMSE-DFE, what would you choose if complexity/cost was your dominant concern? Why? (Give two reasons).

5.15 A multi-band design.

A channel has $H(D) = 1 - D^3$ with $\bar{E}_x = 1$ and $N_0^2 = .01$ and $\bar{E}_x = 1$.

a. (5 pts) Suppose this channel were modelled as 3 independent $1 - D$ channels in parallel. Find the CDFE for $N = 8$ and $\nu = 1$ and the best data rate with $\Gamma = 3$ dB.

b. (1 pt) Now instead model this channel directly with $N = 24$ and $\nu = 3$. How may independent CDFE bands are there, $M =$? .

c. (5 pts) Design the CDFE and compute its SNR for the overall design in part b. You need not provide the feedforward and feedback matrices, but do provide the set of SNR’s for all the subchannels implemented. Use a water-filling rate-adaptive input design for the best covariance.

d. (2 pts) Can you guess an eventual setting for the CDFE’s feedforward and feedback sections from part c. If so, provide it.

e. (1 pt) What is the SNR of a DMT with $N = 24$ and $\nu = 3$?
f. (3 pts) Draw a block diagram of the DMT implementation and of the CDFE implementation and compare (you can label boxes with letters and do not have to provide the settings for each and every matrix entry).

5.16 A Tonal Channel (16 pts)

A complex baseband $2 \times 2$ channel $D$-Transform is

$$H(D) = \begin{bmatrix} 1 + D & -0.5 - 0.4D \\ 0.9 & 1 - D \end{bmatrix}$$

with white (spatially and in frequency) noise (per-tone) variance $N_0 = 0.01$ and the energy per input subsymbol (=sample) is $E_x = 2$ per symbol.

a. (5 pts) What is $\nu$ for this matrix channel for the time-domain symbols? For $\bar{N} = 8$ using the same sampling rate/bandwidth for the channel, find each tonal noise-whitened channel $\tilde{H}_n$. To what level does the symbol energy increase when $\bar{N} = 8$? What is the transmit energy per each tone with $\bar{N} = 8$? (The answer should include $\nu$ value that adjusts for prefix-lost input energy.)

b. (3 pts) For Vector DMT on this channel, first as a single-user channel, find the maximum data rate possible in bits/symbol. Can the gap approximation be well used on this design? Comment on bits/tone and bits/real-dimension.

c. (4 pts) For a $U = 2$-user MAC with the energy per user from Part a, find a GDFE solution/realization with the order such that $\pi(2) = 2$ (top of matrix) and $\pi(1) = 1$ (thus the order is as shown). That is, find the 8 unbiased feedback sections and associated (unbiased) feedforward matrices, along with the total data rates for users 2 and 1. Describe heuristically what you might do if the gap were increased from 0 dB to 3 dB. Also find the design for the other vertex with the same sum data rate.

d. (1 pt) Find the loss of using only a linear receiver for this channel in dB.

e. (1 pt) Find $\gamma_{MAC}$.

5.17 A GDFE Design Problem 5.16’s Channel (8 pts)

This problem continues the previous Problem 5.17. The complex baseband $2 \times 2$ channel $D$-Transform is again

$$H(D) = \begin{bmatrix} 1 + D & -0.5 - 0.4D \\ 0.9 & 1 - D \end{bmatrix}$$

with white noise (per-tone) variance $N_0 = 0.01$ and $E_x = 2$.

a. (5 pts) Is the rate vector $b = [60 50]^*$ feasible if the energy is equally split between the two users? If so, find a design for $\bar{N} = 8$. If not, find a design for $\bar{N} = 8$ with minimum sum energy. The answer need only include the energy and bit distributions for this problem (no need to design the 8 GDFE’s, although an implementation would need them).

b. (3 pts) Repeat Part a for $b = [65 55]^*$. If infeasible, find the best design with minimum energy sum.

5.18 GDFE BC Design with Duality (20 pts)

This exercise returns to Problem 5.16’s MAC

$$H(D) = \begin{bmatrix} 1 + D & -0.5 - 0.4D \\ 0.9 & 1 - D \end{bmatrix}$$

with white (spatially and in frequency) noise (per-tone) variance $N_0 = 0.01$ and $E_x = 2$ with one energy unit for each MAC user. This noise whitened MAC channel is now the dual of another BC, $H_{BC}(D)$. 993
a. (3 pts) Find \( \nu \). Find the number of (complex) channel input dimensions (per symbol) on \( \tilde{H}_{BC} \) if \( \tilde{N} = 8 \)? How many (complex) input dimensions are there for the \( R_{XX} \) used in the tonal channel, so after any transmit linear matrix \( (A) \)? How many dimensions total for all users at the input to any \( A \) matrix and/or lossless precoder input/output on this channel?

b. (3 pts) Using the MAC’s FFT, \( \tilde{H}_{n} \), find the (almost) dual \( \tilde{H}_{BC,n} \) that corresponds to reversing the channel \( \tilde{H}_n \)’s input order (so it corresponds to \( \tilde{H}_{BC,n} = (\tilde{H}_n \cdot J^*) \)). List the \( \tilde{N} \) values for \( \tilde{H}_{BC,n} \). The second reversal on the BC channel input that remains occurs automatically in Part c’s duality program use.

c. (6 pts) Find the set of dual-BC autocorrelation matrices \( \{R_{XX}(u,n)\} \) that correspond to a white (equal energy on all dimensions) input over all users on the MAC, and the corresponding dual channel’s bit distribution, users’ bits/symbol, and overall rate sum. (Hint: use the mac2bc.m program).

d. (2 pts) Find the maximum sum rate for this channel. (Hint: use the bcmax program.) Show how this equals the max energy-sum rate sum.

e. (6 pts) Find the lossless precoders \( (G_n) \) and linear-matrix \( (A_n) \), along with the receivers (unbiased) MSWMF \( (W_n) \) for the BC design corresponding to Part c. (hint: use the mu_bc.m program)

5.19 Specific rate-point GDFE BC Design with Duality (20 pts)

For Problem 5.17’s channel

\[
H(D) = \begin{bmatrix}
1 + D & -0.5 - AD \\
.9 & 1 - D
\end{bmatrix}
\]

with white noise (per-tone) variance \( \frac{N_0}{2} = 0.01 \) and \( \mathcal{E}_x = 2 \). The target data BC rate vector for \( \tilde{N} = 8 \) is \( \mathbf{b} = [50 \ 40]^* \).

a. (6 pts) Find a design that for this data rate that minimizes the BC transmit energy. Find a design using admMAC to set the energy levels used. Show the \( A \) matrix and the precoder \( G \), as well as the (unbiased) MSWMF \( (W) \) for each tone.

b. (5 pts) Are there other designs that achieve the same rate point? For instance, consider a design using admMAC for the rest of this problem.

c. (2 pts) Find the loss relative to a single-user best design for this channel in dB.

d. (2 pts) Find the margin for the design in Part a.

5.20 An Interference Channel Comparison (8 pts)

For the (memoryless) IC

\[
H(D) = \begin{bmatrix}
50 & 50 \\
0 & 1
\end{bmatrix}
\]

with white (spatially and in frequency) noise (per-tone) variance \( \frac{N_0}{2} = 1 \) and \( \mathcal{E}_x = 2 \):

a. (4 pts) Find an iterative (simultaneous) water-filling solution with no GDFE’s and each user as crosstalk to the other.

b. (4 pts) Find the best rate sum when both receivers may use a GDFE for successive decoding and compare to Part a.
Bibliography


Index

admissible rate vector, 904, 962
admission
  problem, 923
backward channel model, 816
bias, 821
CDEF
  finite-length, 831
channel
  dual, 933
  partitioning, 802
channel singularity
  elimination, 806
deflection
  input, 936
DFE
  circulant, 836
  Generalized, 799
diagonal dominance, 901
diagonally dominant, 840
discrete modulator
  GDFE, 816
dual
  channel, 934
dual channel, 933
duality, 933
  single-user, 933
fairness
  proportional, 904
forward channel model, 816
gap
  multiuser, 904
GDFE, 799
  discrete modulator, 816
  nested, 888
  precoded, 828
  tonal, 843, 895
  zero-forcing, 827
GLE, 827
GDFE, 816
  GDFE, 799
  discrete modulator, 816
  nested, 888
  precoded, 828
  tonal, 843, 895
  zero-forcing, 827
IC
  weighted energy sum, 963
  weighted rate sum, 963
  input singularity
    elimination, 811
MAC
  weighted energy sum, 905
  weighted rate sum, 905
MIMO
  massive, 839
  minPmac, 908
  modulator
    discrete, 803, 805, 855
Nash
  Equilibrium, 967
noise-whitening
  scaling, 844
null space, 803
pass space, 803
power control, 970
scattering
  rich, 840
Toeplitz Distribution, 879
tonal Lagrangian, 906, 964
VDSL
  vectored, 901
water-filling
  multi-level, 977
waterfill
  simultaneous, 899
waterfilling
  iterative
    IC, 968