Contents

Chapter 5

Generalized Decision Feedback Equalization

Transmit-signal spectrum optimization can simplify the receiver of QAM/PAM transmission systems while sometimes allowing performance to approach the fundamental limit of capacity. Perhaps not suprisingly, the optimized transmit spectrum is the same as with multichannel transmission in Chapter 4. However, the construction of the transmit symbol sequence is different, as is its phase. The spectrum, symbol rate, and carrier frequency for each disjoint band in water-filling must independently carry a separate transmit-optimized QAM (or PAM if baseband) signal for the best performance to be achieved.

Section 5.1 studies mutual information and innovations and then introduces the so-called "CDEF" result that suggests that a minimal number of MMSE-DFEs with appropriate input selection can also be a canonical transmission structure. Section 5.2 investigates some details in the implementation of such a canonical structure and cautions against misinterpreting the CDEF result. Sections 5.3 investigates a general matrix-block or cyclostationary-channel model and the associated canonical channel construction, discovering the Generalized DFE. Section 5.4 then shows how the usual finite-length MMSE-DFE must be altered to attain canonical performance levels, essentially reducing to a periodic set of DFE coefficients with period equal to the packet length when the channel is stationary. Such a packet of symbols is analogous to the block-of-samples symbol of multichannel. There are an infinite number of such periodic DFE structure's that all have the same performance. A very special case of the periodic packet DFE that re-uses the cyclic prefix of DMT leads to the Cyclic DFE of Section 5.4, from which the relationship of GDFE transmission to multicarrier transmission becomes clear. Section 5.5 then shows that Chapter 4's Vector Coding, and in particular the special case of VC known as DMT, is the simplest and most canonical form of the GDFE. Section 5.6 shows that as block lengths tend to infinity, the CDFE tends to a minimum-number-of-members set of the usual DFEs of Chapter 3 if the channel is stationary. Section 5.7 generalizes all equalization results to a block channel with block intersymbol interference where no guard period is necessary and essentially unifies all the DFE theories into a very general format and approach.

5.1 Information Theoretic Approach to Decision Feedback

Chapter 3's signal-to-noise ratio and filter settings for the MMSE-Decision Feedback Equalizer also follow from some basic information-theoretic results for Gaussian sequences. Annex A contains some definitions of information measures that this chapter uses to establish said information-theoretic approach to DFEs. Section 5.1.1 uses a few basic information measures of entropy and mutual information for Gaussian sequences, particularly that MMSE estimation is fundamentally related to conditional entropy for jointly Gaussian stationary sequences. Entropy is a generalization of data rate, while mutual information represents a reduced data rate associated with the transport of a symbol sequence over a distorted channel. Combination of the MMSE results with the information measures then simply relate the "CDEF" result of decision feedback equalization in Section 5.1.2, which suggests that good codes in combination with an appropriate (set of) DFE(s) can reliably allow the highest possible transmission rates, even though the receiver is not MAP/ML. Such a result is surprising as equalization methods as originally conceived are decidely suboptimal, but with care of the input spectrum choice can be made to perform at effectively optimum, or "canoncial" levels.

5.1.1 MMSE Estimation and Conditional Entropy

Given two complex jointly Gaussian random variables, x and y, the conditional probability density $p_{x/y}$ is also a Gaussian density and has mean $E[x/y]$ equal to the MMSE estimate of x given y. (This result is easily proved as an exercise by simply taking the ratio of $p_{x,y}$ to p_y, which are both Gaussian with the general N-complex-dimensional ($2N$ real dimensions) form - see Problem 5.7) The Gaussian vector distribution is $1/(\pi^N |R_{\boldsymbol{xx}}|) e^{-(\boldsymbol{x}-\boldsymbol{u}_x)R_{\boldsymbol{xx}}{}^{-1}(\boldsymbol{x}^*-\boldsymbol{u}_x^*)}$, where $R_{\boldsymbol{xx}}$ is the covariance matrix and $\boldsymbol{u}_x$ is the mean.) The entropy (see Appendix A) of a complex Gaussian random variable is

$$H_x = \log_2(\pi e \mathcal{E}_x) \quad , \tag{5.1}$$

where $\mathcal{E}_x$ is the mean-square value of x. Thus, the conditional entropy of x given y is

$$H_{x/y} = \log_2(\pi e \sigma^2_{x/y}) \quad . \tag{5.2}$$

Entropy can be normalized to the number of real dimensions and complex random variables have the same variance in both real dimensions, which is 1/2 the value of the complex variance. Thus, $\bar{H}_x = \frac{1}{2} \cdot \log_2(2\pi e \bar{\mathcal{E}}_x)$ whether x is real or complex. These results generalize to jointly $\bar{N}$-complex-dimensional Gaussian random vectors (so $N = 2\bar{N}$ real dimensions) as

$$H_{\boldsymbol{x}} = \log_2\{(\pi e)^{\bar{N}} |R_{\boldsymbol{xx}}|\} \tag{5.3}$$

and

$$H_{\boldsymbol{x/y}} = \log_2\{(\pi e)^{\bar{N}} |R^{\perp}_{\boldsymbol{x/y}}|\} \tag{5.4}$$

respectively, where $R^{\perp}_{\boldsymbol{x/y}} = R_{\boldsymbol{xx}} - R_{\boldsymbol{xy}} R^{-1}_{\boldsymbol{yy}} R_{\boldsymbol{yx}}$ is the autocorrelation matrix of the error associated with the vector MMSE estimate of $\boldsymbol{x}$ from $\boldsymbol{y}$. As always $\bar{R}_{\boldsymbol{xx}} = \frac{1}{N} \cdot R_{\boldsymbol{xx}}$. Again, per-dimensional quanties are found by dividing by the number of real dimensions $N = 2\bar{N}$: $\bar{H}_{\boldsymbol{x}} = \frac{1}{2}\log_2\{(\pi e)|R_{\boldsymbol{xx}}|^{1/\bar{N}}\}$. If $\boldsymbol{x} = x$, i.e., a scalar, then $R_{\boldsymbol{xx}} = 2\bar{\mathcal{E}}_x$ in the entropy formula with $\bar{N} = 1$ and all per-dimensional results are consistent.[1]

[1] For the interested in alternative expressions (that provide the same entropy): If $\boldsymbol{x}$ is real, then $H_{\boldsymbol{x}} = \frac{1}{2}\log_2\left[(2\pi e)^N |R_{\boldsymbol{xx}}|\right]$ or

$$\bar{H}_{\boldsymbol{x}} = \frac{1}{2N}\log_2\left[(2\pi e)^N \cdot |N\bar{R}_{\boldsymbol{xx}}|\right] \tag{5.5}$$

$$= \frac{1}{2}\log_2\left[(2N\pi e) \cdot |\bar{R}_{\boldsymbol{xx}}|^{1/N}\right] \quad , \tag{5.6}$$

which checks with one-dimensional formule. If $\boldsymbol{x}$ is complex, then $H_{\boldsymbol{x}} = \log_2\left[(\pi e)^{\bar{N}} |R_{\boldsymbol{xx}}|\right]$ or

$$\bar{H}_{\boldsymbol{x}} = \frac{1}{2\bar{N}}\log_2\left[(\pi e)^{\bar{N}} \cdot |2\bar{N} \cdot \bar{R}_{\boldsymbol{xx}}|\right] \tag{5.7}$$

For stationary Gaussian sequences, the **chain rule** of entropy allows computation of the entropy of a Gaussian sequence. The chain rule for a Gaussian random vector $\boldsymbol{x} = [x_k, x_{k-1}, ..., x_0]$ is

$$H_{\boldsymbol{x}} = H_{x_k/[x_{k-1},...,x_0]} + H_{x_{k-1}/[x_{k-2},...,x_0]} + ... + H_{x_1/x_0} + H_{x_0} = \sum_{n=0}^{k} H_{x_n/[x_{n-1},...,x_0]} \quad . \tag{5.10}$$

($Hx_{0/-1} \stackrel{\Delta}{=} H_{x_0}$.) The first k terms in the sum above are conditional entropies that each equal the logarithm of πe times the MMSE associated with prediction of a component of $\boldsymbol{x}$ based on its past values. The entropy of a stationary Gaussian sequence is defined by the limit of the entropy of the vector $[x_k, ..., x_0]$, normalized to the number of dimensions:

$$H_{X(D)} = \lim_{k\to\infty} \frac{1}{k+1} \sum_{n=0}^{k} H_{x_n/[x_{n-1},...,x_0]} \quad \text{bits/complex dimension.} \tag{5.11}$$

For an infinite-length stationary sequency, essentially all the terms in the sum above must be the same, so

$$H_{X(D)} = \log_2 \left[(\pi e) S_x\right] \quad , \tag{5.12}$$

where S_x is the MMSE associated with computing x_k from its past, which MMSE linear prediction can be recognized as monic causal $V(D) = A(D) \cdot X(D)$ where $A(D) = 1 + a_1 D + a_2 D^2 + ...$ so the product corresponds to $v_k = x_k + a_1 \cdot x_{k-1} + a_2 \cdot x_{k-2} + ...$ Then, the mean-square prediction error is $E[|v_k|^2]$, which is the time-zero value of the autocorrelation function

$$R_{vv}(D) = A(D) \cdot R_{xx}(D) \cdot A^*(D^{-*}) \quad . \tag{5.13}$$

For Gaussian processes, the MMSE estimate is linear, and is best found by canonical factorization (presuming the Paley-Wiener criterion is satisfied for the process) of its autocorrelation function:

$$R_x(D) = S_x \cdot G_x(D) \cdot G_x^*(D^{-*}) \quad , \tag{5.14}$$

as in Chapter 3, where S_x is a positive constant and $G_x(D)$ is causal, monic, and minimum phase. The time-zero value of $R_{vv}(D)$ is then found as

$$E[|v_k|^2] = S_x \cdot \|A/G_x\|^2 \quad , \tag{5.15}$$

which is minimized for monic causal choice of A when $A(D) = 1/G_x(D)$. This MMSE linear prediction filter is then shown in the lower filter in Figure 5.1. S_x is the MMSE. The output process $V(D)$ is generated by linear prediction, and is also Gaussian and called the **innovations sequence.** This process carries the essential information for the process $X(D)$, and $X(D)$ can be generated causally from $V(D)$ by processing with $G_x(D)$ to shape the power spectrum, alter the power/energy, but not change the information content of the process, as in Figure 5.1. When $X(D)$ is white, it equals its innovations.

The **mutual information** between two random variables x and y is

$$\begin{aligned} I(x;y) &\stackrel{\Delta}{=} H_x - H_{x/y} && (5.16)\\ &= \log_2\left(\frac{S_x}{\sigma^2_{x/y}}\right) = \log_2(1 + \text{SNR}_{mmse,u}) && (5.17)\\ &= H_y - H_{y/x} && (5.18)\\ &= \log_2\left(\frac{S_y}{\sigma^2_{y/x}}\right) \quad , && (5.19) \end{aligned}$$

$$\begin{aligned} &= \frac{1}{2\bar{N}} \log_2 \left[(\pi e)^{\bar{N}} \cdot (2\bar{N})^{\bar{N}} \cdot |\bar{R}_{\boldsymbol{xx}}|\right] && (5.8)\\ &= \frac{1}{2} \log_2 \left[(2\bar{N}\pi e) \cdot |\bar{R}_{\boldsymbol{xx}}|^{1/\bar{N}}\right] && (5.9) \end{aligned}$$

which also checks with the one dimensional formulae. When a complex vector is modeled as a doubly-dimensional real vector, one can see the two formulae for normalized entropy are the same as they should be.

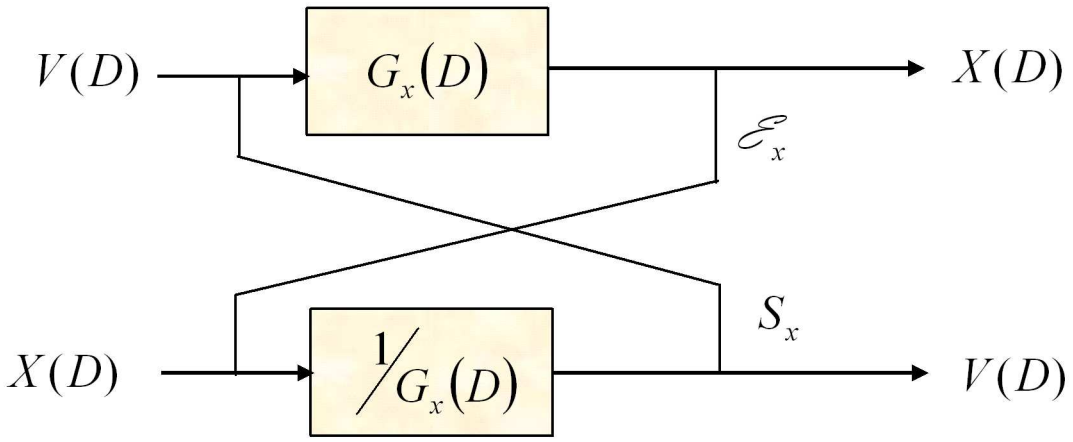


Figure 5.1: Ilustration of Linear Prediction and relation to entropy for stationary Gaussian process.

showing a symmetry between x and y in estimation, related through the common SNR that characterizes MMSE estimation,

$$1 + \text{SNR}_{mmse,u} = \frac{S_x}{\sigma^2_{x/y}} = \frac{S_y}{\sigma^2_{y/x}} \quad . \tag{5.20}$$

Equation 5.17 uses the unbiased SNR. For an AWGN, $y = x + n$, $S_y = \mathcal{E}_x + \sigma^2 = \mathcal{E}_x + \sigma^2_{y/x}$. Since $\text{SNR} \triangleq \bar{\mathcal{E}}_x/\sigma^2$, then Equation 5.20 relates that

$$1 + \text{SNR}_{mmse,u} = 1 + \text{SNR} \quad , \tag{5.21}$$

and thus

$$\text{SNR}_{mmse,u} = \text{SNR} \quad . \tag{5.22}$$

Thus the unbiased SNRcharacterizing the forward direction of this AWGN channel is thus also equal to the unbiased SNR ($\text{SNR}_{mmse,u}$) in estimating the backward channel of x given y, a fact well established in Chapter 3. The result will extend to random vectors where the variance quantities are replaced by determinants of covariance matrices as in the next subsection.

5.1.2 The relationship of the MMSE-DFE to mutual information

In data transmission, the largest reliably transmitted data rate for a given input sequence covariance/spectrum is the mutual information between the sequence and the channel output sequence (see Chapter 8). This result presumes 1-shot maximum-likelihood detection after observing the entire output sequence $Y(D)$ for the entire input sequence $X(D)$. For the ISI channel, this mutual information is

$$\bar{I}(X(D);Y(D)) = \bar{H}_{X(D)} - \bar{H}_{X(D)/Y(D)} \quad . \tag{5.23}$$

The entropy of a stationary Gaussian sequence is determined by the innovations process, or equivalently its MMSE estimate given its past, thus (5.23) becomes

$$\begin{aligned}
\bar{I}(X(D);Y(D)) &= \bar{H}_{x_k/[x_{k-1},\ldots]} - \bar{H}_{x_k/Y(D),[x_{k-1},\ldots]} && (5.24)\\
&= \frac{1}{2}\cdot\log_2(\pi e S_x) - \frac{1}{2}\cdot\log_x(\pi e \sigma^2_{\text{MMSE-DFE}}) && (5.25)\\
&= \frac{1}{2}\cdot\log_2(\text{SNR}_{\text{mmse-dfe}}) && (5.26)\\
&= \frac{1}{2}\cdot\log_2(1+\text{SNR}_{\text{mmse-dfe,u}}) \quad . && (5.27)
\end{aligned}$$

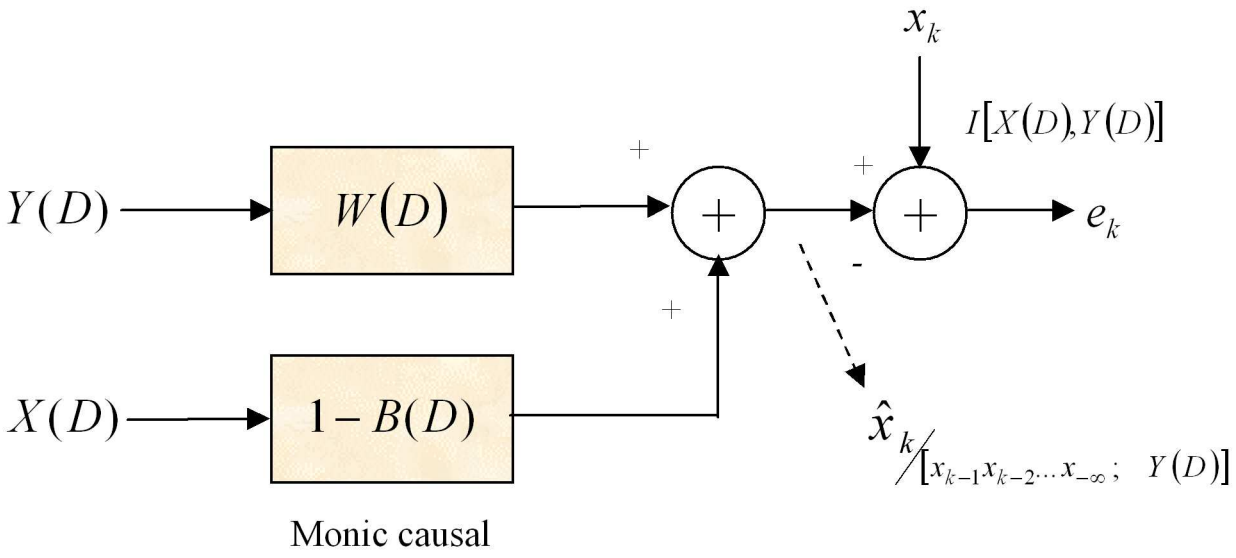


Figure 5.2: MMSE estimation problem associated with $I(X(D);Y(D))$ – the same as the MMSE-DFE.

The main observation used in the last 3 steps above is that the conditional entropy of x_k, given the entire sequence $Y(D)$ and the past of the sequence x_k exactly depends upon the MMSE estimation problem that the DFE solves. Thus the variance associated with the conditional entropy is then the MMSE of the MMSE-DFE. This result was originally noted by four authors[2] and is known as the **CDEF** result (the CDEF result was actually proved by a more circuitous route than in this chapter, with the shorter proof being first shown to the author by Dr. Charles Rohrs of Tellabs Research, Notre Dame, IN).

Lemma 5.1.1 (CDEF Result) *The unbiased SNR of a MMSE-DFE is related to mutual information for a linear ISI channel with additive white Gaussian noise in exactly the same formula as the SNR of an ISI-free channel is related to the mutual information of that channel:*

$$SNR_{mmse\text{-}dfe,u} = 2^{2\bar{I}} - 1 \quad , \tag{5.28}$$

where the mutual information I is computed assuming jointly Gaussian stationary $X(D)$ and $Y(D)$, and only when the MMSE-DFE exists.
Proof: *Follows development above in Equations (5.24)-(5.27).***QED.**

The CDEF result has stunning implications for transmission on the AWGN channel with linear ISI: It essentially states that the suboptimum MMSE-DFE detector, when combined with the same good codes that allow transmission at or near the highest data rates on the ISI-free channel (see Volume II, Chapter 8), will attain the highest possible data rates reliably. This result is the same as the infinite-dimensional MT result of Chapter 4, so another equally good transmission system has apparently been found. In all cases, the form of the mutual information used depends on Gaussian $X(D)$. Since a Gaussian $X(D)$ never quite occurs in practice, all analysis is approximate to within the constraints of a finite non-zero gap, $\Gamma > 0$ dB as always in this Chapter, and one could write $\bar{b} = \frac{1}{2}\log_2\left(1 + \frac{\mathrm{SNR}_{\mathrm{mmse\text{-}dfe,u}}}{\Gamma}\right) \leq \bar{I}$ when the DFE exists. The equality holds when the gap is 0dB.

Figure 5.2 depicts the estimation problem associated with $I(X(D);Y(D))$. This particular interpretation shows that the MMSE-DFE structure, when it exists, can be used to approach capacity with $\Gamma \to 0$ dB with the same codes that are used to approach capacity on the AWGN. This implies the MMSE-DFE could then be canonical, especially if the water-filling energy distribution were used by the transmitter to maximize $\bar{I}(X(D);Y(D))$ to capacity.

[2] "MMSE DFE and Coding: Parts I and II," *IEEE Transactions on Communications*, October 1995, pp. 2582-2604.

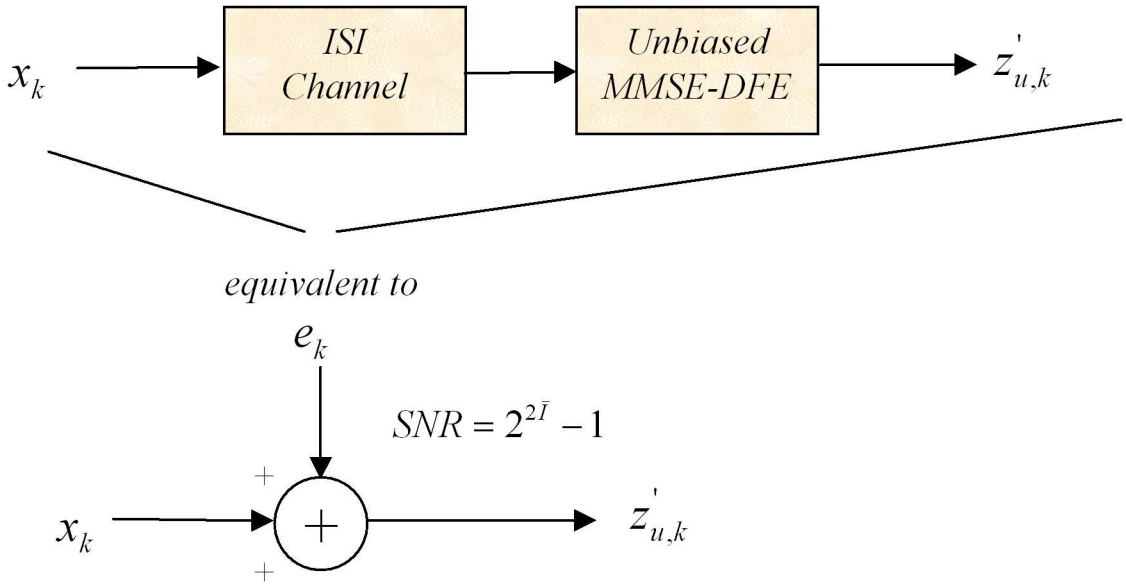


Figure 5.3: CDEF canonical equivalent AWGN channel to MMSE-DFE where codes with gap Γ from capacity on the AWGN channel can be re-used to have the same gap to capacity on the ISI channel.

Definition 5.1.1 (Canonical Performance) *The performance of a transmission design is said to be* **canonical** *if the signal-to-noise ratio of the equivalent AWGN characterizing the system is* $2^{2\bar{I}} - 1$ *when the gap is* $\Gamma = 0$ *dB.*

Clearly then, the MMSE-DFE is canonical when it exists. The mutual information of a set of independent parallel channels is the sum of the mutual information of each of the channels (see the Appendix). Thus, all multichannel systems are canonical if partitioned so that subchannels are independent and free of intersymbol-interference.

Theorem 5.1.1 (Canonical DMT/VC) *For any and all block lengths and any input distribution of energy among the independent subchannels of DMT or VC, DMT and VC are both canonical transmission systems.*
Proof:. Follows trivial from sum of mutual information of the subchannels equals the mutual information of the entire channel. **QED.**

Clearly, then MT and MM are also canonical as the infinite block-length limits of DMT and VC respectively.

However, there are restrictions on the above result that were not made explicit to simplify the developments. In particular, the designer must optimize the transmit filter for the MMSE-DFE to get the highest mutual information, as in Chapter 4. This process can, and almost always does, lead to unrealizable filters. Happily, there are solutions, but the resulting structures are not the traditional MMSE-DFE except in special cases. Sections 5.2 and 5.3 study the necessary modifications of the DFE structure. Section 5.4 finds the exact mirror image(s) of DMT.

5.1.3 Canonical Channel Models

The symmetry of mutual information between $X(D)$ and $Y(D)$ suggest two interpretations of the relationship between $X(D)$ and $Y(D)$, known as the canonical channel models of Figure 5.4. As in Chapter 4, $r(t) = p(t) * p^*(-t)$ (where any minor transmit and receive analog filtering has been absorbed into the channel impulse/pulse response, while any innovations filtering remains separate), and

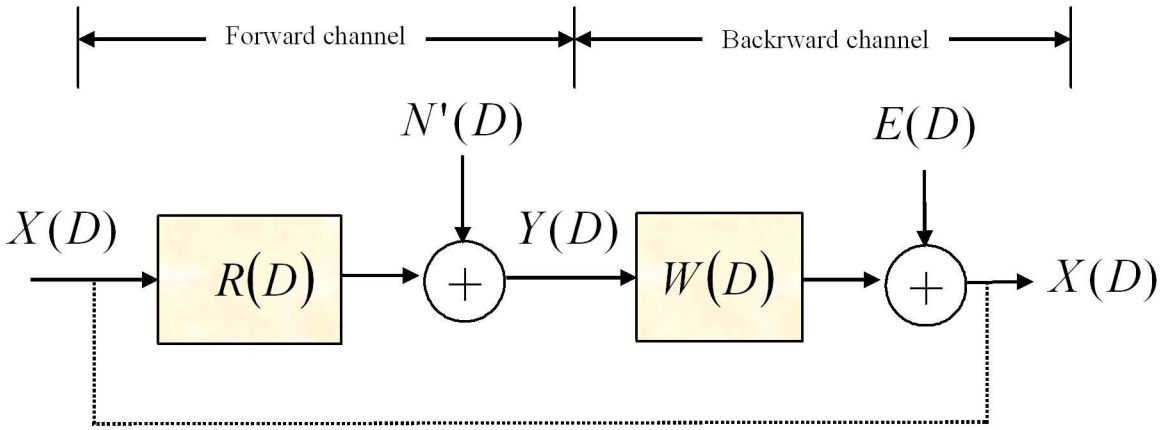


Figure 5.4: Canonical Channel models for same mutual information.

$R(D) = \sum_k r(kT) \cdot D^k$. $Y(D)$ again corresponds to the overall channel shaping at the sampled matched-filter output.

> **Definition 5.1.2 (Canonical Channel)** *A* **canonical channel** *is one in which the linear function or matrix characterizing ISI (or more generally cross-dimensional interference as occurs from Sections 5.3 onward) is equal to the autocorrelation function (or matrix) of the additive independent interference. Interference can be Gaussian noise, or some combination of additive Gaussian noise and residual ISI.*

Canonical channels can lead to canonical performance with the proper design of receiver. There are two canonical channels of interest in this chapter:

The **forward canonical model** has

$$Y(D) = R(D) \cdot X(D) + N'(D) \quad , \tag{5.29}$$

where $N'(D)$ is the Gaussian noise at the output of the matched filter with autocorrelation function $\frac{\mathcal{N}_0}{2} R(D)$. Thus the noise power spectral density and the channel filtering have the same shape $R(e^{-\jmath\omega T})$. The first term on the right in (5.29) is the MMSE estimate of $Y(D)$ given $X(D)$ and $N'(D)$, the MMSE, and $X(D)$ are independent. The **backward canonical model** is

$$X(D) = W(D) \cdot Y(D) + E(D) \quad . \tag{5.30}$$

where (modulo scaling by $\|p\|^{-1}$) $W(D)$ is the MMSE-LE of Chapter 3 and $E(D)$ is the MMSE error sequence for the MMSE-LE. The first term on the right in (5.30) is the MMSE estimate of $X(D)$ given $Y(D)$. The shape of the equalizer and the error-sequence's power spectral density are both $R_{ee}(e^{-\jmath\omega T})$, since $\bar{R}_{ee}(D) = \frac{\mathcal{N}_0}{2} / \{\|p\|^2 \cdot Q(D) + \frac{\mathcal{N}_0}{2} / (\mathcal{E}_{\boldsymbol{x}} \|p\|^2)\} = \frac{\mathcal{N}_0}{2} \cdot W(D)$. Canonical channels always exhibit a response with the same shaping as the power spectral density of the additive Gaussian noise.

It is relatively simple to construct DFE's from canonical models: For the forward model, the receiver processes $Y(D)$ by the inverse of the anticausal spectral factor of $R(D) = (S_r \cdot G_r(D) \cdot G_r^*(D^{-*})$ to obtain

$$\frac{Y(D)}{S_r \cdot G_r^*(D^{-*})} = G_r(D) \cdot X(D) + N''(D) \tag{5.31}$$

where $N''(D)$ is white Gaussian noise with energy per dimension $\frac{\mathcal{N}_0}{2} / S_r$. Since $G_r(D)$ is monic, causal, and minimum-phase, a DFE can be readily implemented. This DFE is the ZF-DFE of Chapter 3. A forward canonical model will always produce a ZF-DFE. The receiver is not optimum in general mainly because the DFE is not optimum – furthermore, the suboptimum DFE implied is not of highest SNR

because the DFE is based on a model that is for the problem of estimating $Y(D)$ from $X(D)$. The DFE from the backward model is the MMSE-DFE. Noting that

$$R(D) + \frac{1}{\text{SNR}} = \|p\|^2 \cdot \left[Q(D) + \frac{1}{\text{SNR}_{\text{mfb}}} \right] = \gamma_0 \|p\|^2 \cdot G(D) \cdot G^*(D^{-*}) \quad , \tag{5.32}$$

with $S_0 = \gamma_0 \cdot \|p\|^2$, then

$$W_{ff}(D) = \frac{1}{S_0 \cdot G^*(D^{-*})} \quad . \tag{5.33}$$

Then also the action of the feedforward filter $W(D)$ on the sampled matched-filter output $Y(D)$ is

$$\frac{Y(D)}{S_0 \cdot G^*(D^{-*})} = G(D) \cdot X(D) - G(D) \cdot E(D) = G(D) \cdot X(D) - E'(D) \quad . \tag{5.34}$$

where $E'(D)$ is the MMSE sequence associated with the MMSE-DFE and is white (and Gaussian when $X(D)$ is Gaussian) and has energy per dimension $\frac{\mathcal{N}_0}{2}/S_0$. The extension of canonical models to the case of finite-length packets allows a generalization of the DFE later in this chapter.

The forward and backward canonical channel models have the same mutual information between input and output. In the backward model, $Y(D)$ is considered the input, and $X(D)$ is the corresponding output, with $E(D)$ as the noise. Both channels have the same maximum $\bar{b}$ of $\bar{I}(X(D);Y(D))$, and it is the backward channel that describes the receiver's action. The operation of decoding or slicing in an actual receiver that uses previous decisions to remove the effort of the causal monic $G_r(D)$ or $G(D)$ through decision feedback in (5.33) or (5.34) is information lossy in general, confirming that DFE's are not optimum ML detectors. However, for the backward channel only, the $\text{SNR}_{\text{mmse-dfe,u}}$ is equal to $2^{2\bar{I}(X(D);Y(D))} - 1$, which is the maximum SNR value for the additive white noise/distortion channel created by the MMSE-DFE. Thus, the information loss in the MMSE-DFE does not cause a reduction in achievable data rate and indeed a code with a given gap to capacity on the AWGN channel would be just as close to the corresponding capacity for a bandlimited channel, even when the suboptimum-detector MMSE-DFE is used. This characteristic occurs only for backward canonical models.

5.2 Construction of the Optimized DFE Input

The CDEF result shows a relationship between mutual information $\bar{I}(X(D);Y(D))$ and $\text{SNR}_{\text{mmse-dfe,u}}$ for an AWGN equivalent of the ISI-channel that is the same as the relationship between $\bar{I}(x;y)$ and SNR for the (ISI-free) AWGN channel. Thus, those same good codes that bring performance to within gap Γ of capacity on the AWGN channel can then be applied (ignoring error propagation) to a MMSE-DFE system to achieve the same gap from the ISI-channel's capacity.

However, the MMSE-DFE so far has used an i.i.d. input sequence x_k, which does not usually maximize $\bar{I}(X(D);Y(D))$. Maximization of $\bar{I}(X(D);Y(D))$ in Chapter 4 defines a best water-filling (WF) spectrum. This WF power spectrum maximizes $\bar{I}(X(D);Y(D))$. A designer might then assume that construction of a transmit filter for the MMSE-DFE with the water-filling power-spectral density would then maximize $\text{SNR}_{\text{mmse-dfe,u}} = 2^{2\bar{I}(X(D);Y(D))} - 1$. This is correct if the measure of the water-filling band is equal to the Nyquist bandwidth, i.e. $|\Omega^*| = 1/T$, or equivalently, all frequencies (except a countable number of infinitessimally narrow notches) must be used by water-filling. This rarely occurs by accident[3], and so the designer must be careful in selecting the symbol rates and carrier frequencies of a minimum-size set of QAM/MMSE-DFE channels that can be made almost equivalent[4].

It is important for the reader to distinguish **coding**, which here is restricted to mean the use of codes like trellis codes, block codes, turbo codes, etc. (see Chapters 10 and 11) on an AWGN channel from the concept of spectrum design (sometimes often also called "coding" in the literature in a more broad use of the term "coding"). This chapter focuses on spectrum or input design and presumes use of good known codes for the AWGN channel in addition to the designed spectrum. The two effects are independent. This chapter investigates the designed spectra or input to channels. Volume II investigates codes.

This section address optimization of the transmit signal, which involves optimization of the transmit filter(s), symbol rate(s), and carrier frequency(ies). Subsection 5.2.1 begins with a review of the discrete-time Paley-Wiener (PW) criterion, which is necessary for a canonical factorization to exist for a random sequence's power spectral density. With this PW criterion in mind, Subsection 5.2.2 maximizes $\text{SNR}_{\text{mmse-dfe,u}}$ over the transmit filter to find a desired transmit power spectral density, which has a water-fill shape when it exists. Subsections 5.2.3 then studies the choice of symbol rates, and possibly carrier frequencies, for a continuous-time channel so that the PW criterion will always be satisfied. Such optimization in these 3 subsections for different increasingly more general cases enables implementation of a countable set realizable transmit filters in each disjoint continuous-time/frequency water-filling band of nonzero measure. Subsection 5.2.5 culminates in the definition of the optimum MMSE-DFE.

5.2.1 The Discrete-Time Paley-Wiener Criterion and Filter Synthesis

Theorem 5.2.1 (Discrete-Time Paley Wiener) *The canonical factorization $R_{xx}(D) = S_x \cdot G_x(D) \cdot G_x^*(D^{-*})$ of the power spectral density of a stationary random sequence x_k exists if and only if*

$$\int_{-\pi/T}^{\pi/T} |\ln R_{xx}(e^{-\jmath\omega T})| d\omega < \infty \quad . \tag{5.35}$$

sketch of proof: The integral condition is trivial true when $R_{xx}(D) = S_x \cdot G_x(D) \cdot G_x^*(D^{-*})$ is inserted with $D = e^{-\jmath\omega T}$ and the integral evaluated over one period to obtain $\ln(S_x) < \infty$. Conversely, if the integral is finite, then the integrand can be repeated into a Fourier series of period T that has the same value of $\ln|R_{xx}(e^{\jmath\omega T})|$ over all periods. The Fourier series coefficients essentially then define $G_x(D)$' coefficients. **QED.**

The factor $G_x(D)$ is monic, causal, and minimum phase. There exists a white **innovations** process v_k with energy per sample $\mathcal{E}_v = S_x$, and autocorrelation function $R_{vv}(D) = S_x$. The process $X(D)$ can be

[3] Some designers of single-carrier QAM systems often err in assuming it does occur by accident only to find their system does not perform as expected.

[4] "almost" because error propagation or precoder loss always reduces even the best-designed DFE systems to some small amount of loss.

constructed from $V(D)$ according to

$$X(D) = V(D) \cdot G_x(D) \quad . \tag{5.36}$$

Any process so constructed can be inverted to get the innovations process by (even when $G_x(D)$ has zeros on the unit circle or equivalently the spectrum has a countable number of "notches")

$$V(D) = \frac{X(D)}{G_x(D)} \tag{5.37}$$

as shown in Figure 5.1. There is a causal and causually invertible relationship between $X(D)$ and $V(D)$ that allows realizable synthesis or implementation of the filters relating the two random sequences. Random sequences with power spectra that do not satisfy the PW criterion can not be so constructed from a white innovations sequence. As shown in Subsection 5.1.1, the innovations sequence determines the information content of the sequence. In data transmission, the innovations sequence is the message sequence to be transmitted.

For the MMSE-DFE in Chapter 3, the sequence $X(D)$ was white, and so equals its innovations, $X(D) = V(D)$.

EXAMPLE 5.2.1 (White Sequence) A real random sequence with $R_{xx}(D) = \mathcal{E}_x$ satisfies the Paley Wiener Criterion trivially and has innovations equal to itself $X(D) = V(D)$, $G_x(D) = 1$. This entropy is $\bar{H}_x = .5 \log_2(2\pi \cdot e \cdot \bar{\mathcal{E}}_{\boldsymbol{x}})$ bits/dimension, and $H_x = \log_2(\pi \cdot e \cdot \mathcal{E}_x)$ bits per symbol if this random sequence is complex. The spectrum is shown in Figure 5.5(a).

EXAMPLE 5.2.2 (AMI Sequence) A sequence with $R_{xx}(D) = -1D^{-1} + 2 - D$ clearly factors into $R_{xx}(D) = 1 \cdot (1-D)(1-D^{-1})$ and thus must satisfy the Paley Wiener Criterion. The innovations sequence is $V(D) = X(D)/(1-D)$ and has unit energy per sample.[5] The energy per sample of $X(D)$ is $r_{xx,0} = 2$. The energy per dimension is $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$. The entropy is $\bar{H}_x = .5 \log_2(2\pi \cdot e \cdot \bar{\mathcal{E}}_{\boldsymbol{x}})$ bits/dimension – the same as Example 5.2.1, even though the spectrum in Figure 5.5(b) is different.

The last example illustrates what is obvious or thought about linear prediction: the innovations never has energy/sample greater than the sequence from which it is derived.

EXAMPLE 5.2.3 (Ideal Lowpass Process) A sequence with ($\zeta < 1$ is a positive constant)

$$R_{xx}(e^{-\jmath\omega T}) = \begin{cases} 1 & |\omega| < \zeta \frac{\pi}{T} \\ 0 & |\omega| \geq \zeta \frac{\pi}{T} \end{cases} \tag{5.38}$$

does NOT satisfy the PW criterion and has power-spectral density shown in Figure 5.5(c). This process cannot be realized by passing a white information sequence through a "brickwall lowpass filter" because that filter is always noncausal even with aribtrarily large delay in implementation. An approximatation of such a filter with a causal filter, which would lead to a slightly different power spectral density, could satisfy the PW criterion. The designer might better consider a new sampling rate of ζ/T, and the resulting new sampled random sequence would then satisfy the PW criterion trivially as in Example 5.2.1, rather than incur the complexity of such a filter.

When optimizing a transmit complex transmit filter, an adjustment to the real basis function of Chapter 2 needs to occur. The reader may recall that the QAM basis functions from Chapter 2 were of the form:

$$\varphi_1(t) = \sqrt{\frac{2}{T}} \cdot \varphi(t) \cdot \cos(\omega_c t) \tag{5.39}$$

$$\varphi_2(t) = -\sqrt{\frac{2}{T}} \cdot \varphi(t) \cdot \sin(\omega_c t) \quad , \tag{5.40}$$

[5] This filter is marginally stable in that a pole is on the unit circle - however, the input to it has zero energy at this location (DC).

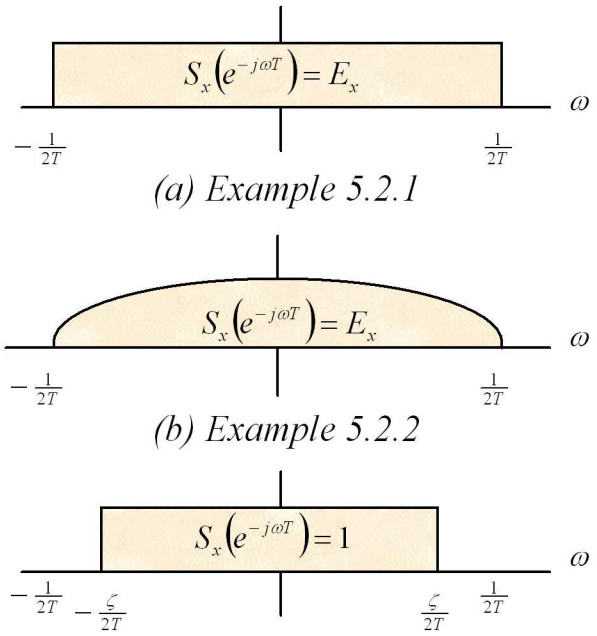


Figure 5.5: Illustration of two spectra that satisfy discrete PW in (a) and (b) while (c) does not satisfy the PW criterion.

where $\varphi(t)$ was real. When optimizing in later sections of this chapter, a complex basis function with orthogonal real and imaginary parts may result. QAM transmission may be generalized to complex basis functions, when needed, by instead generalizing the two basis functions in (5.40) to:

$$\varphi_1(t) = \sqrt{\frac{2}{T}} \cdot \phi_1(t) \cdot \cos(\omega_c t) \tag{5.41}$$

$$\varphi_2(t) = -\sqrt{\frac{2}{T}} \cdot \phi_2(t) \cdot \sin(\omega_c t) \quad , \tag{5.42}$$

which correspond to a complex baseband transmit filter with

$$\varphi(t) = \phi_1(t) + \jmath\phi_2(t) \quad , \tag{5.43}$$

where $\phi_1^2(t) + \phi_2^2(t) = 1$ and have all energy below the frequency ω_c.

5.2.2 Waterfilling and the Paley-Wiener Citerion

The mutual information $\bar{I}(X(D); Y(D))$ is a function only of the power spectral densities of the processes $X(D)$ and $Y(D)$ if they are Gaussian. It is thus perhaps too easy to maximize $\bar{I}(X(D); Y(D))$ over the power spectral density of $X(D)$, thus producing a water-fill spectrum for the transmit filter, which presumably processes a white (and "near-Gaussian") message sequence.

Figure 5.6 shows the implementation of a transmitter that includes a digital filter ϕ_k (with transform $\Phi(D)$). This filter preceeds the modulator that converts the symbols at its output x_k into the modulated waveform $x(t)$, typically QAM or PAM. The discrete-time filter input is v_k, which now becomes the message sequence. Recalling that $q(t) \triangleq p(t) * p^*(-t)/\|p\|^2$, the mutual information or maximum data

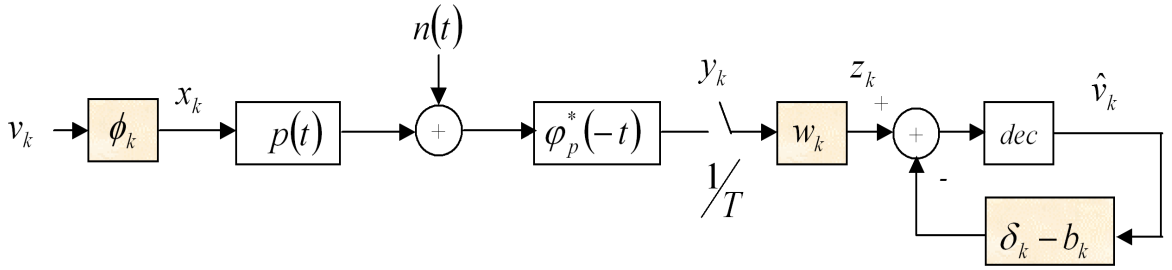


Figure 5.6: MMSE-DFE system with digital transmit filter.

rate from Section 4.4.3 is

$$\bar{I}(X(D);Y(D)) = \frac{T}{2\pi}\int_{-\pi/T}^{\pi/T} \log_2\left(\frac{\|p\|^2\cdot\bar{\mathcal{E}}_v}{\frac{\mathcal{N}_0}{2}}\cdot|\Phi(e^{-\jmath\omega T})|^2\cdot Q(e^{-\jmath\omega T})+1\right)d\omega \quad . \tag{5.44}$$

The transmit energy constraint is the $\Gamma = 0$dB-sum of data rates on an infinite number of infinitessimally small tones:

$$\frac{T}{2\pi}\int_{-\pi/T}^{\pi/T}\bar{\mathcal{E}}_v\cdot|\Phi(e^{-\jmath\omega T})|^2 d\omega = \bar{\mathcal{E}}_x \quad . \tag{5.45}$$

The maximization of $\bar{I}(X(D);Y(D))$ is then achieved by classic water fill solution in

$$|\Phi(e^{-\jmath\omega T})|^2 + \frac{\frac{\mathcal{N}_0}{2}}{\bar{\mathcal{E}}_v\cdot\|p\|^2\cdot Q(e^{-\jmath\omega T})} = K \quad . \tag{5.46}$$

As always, energy at any frequency ω must be nonnegative, so

$$|\Phi(e^{-\jmath\omega T})|^2 \geq 0 \quad . \tag{5.47}$$

There is a set of frequencies Ω^* for which transmit energy is nonzero and for which the discrete-time transmit filter output energy satisfies water-filling. The corresponding capacity is then

$$\bar{C} = \max_{\|\Phi(e^{-\jmath\omega T})\|^2=1}\bar{I}(X(D);Y(D)) \quad . \tag{5.48}$$

The **measure** of Ω^* is again

$$|\Omega^*| = \frac{T}{2\pi}\int_{\Omega_*} d\omega \quad . \tag{5.49}$$

If $|\Omega^*| = 1$, then a realizable transmit filter exists by the PW criterion. In this case, $\text{SNR}_{\text{mmse-dfe,u}} = 2^{2\bar{C}} - 1$ and the MMSE-DFE transmission system with water-fill transmit power spectral density can achieve, with powerful known AWGN-channel codes, the highest possible rates, so the MMSE-DFE in this case only is effectively optimal or "canonical." **If the water-fill band does not have unit measure, the transmitter is not realizable, and the MMSE-DFE is not optimal.** The non-unit-measure case is very likely to occur unless the sampling rate has been judiciously chosen. This is the single most commonly encountered mistake of QAM/PAM designers who attempt to match the performance of DMT systems in Chapter 5 - only in the rare case of $|\Omega^*| = 1$ can this design work.

The **continuous-time** capacity in bits per second for a transmission channel where the symbol rate has not yet been selected from Chapter 4 (Subsection 4.4.2) is (in bits/second)

$$\mathcal{C} = \frac{1}{2\pi}\int_{\Omega^*}\frac{1}{2}\log_2\left(1+S_x(\omega)\cdot g(\omega)\right)d\omega \quad , \tag{5.50}$$

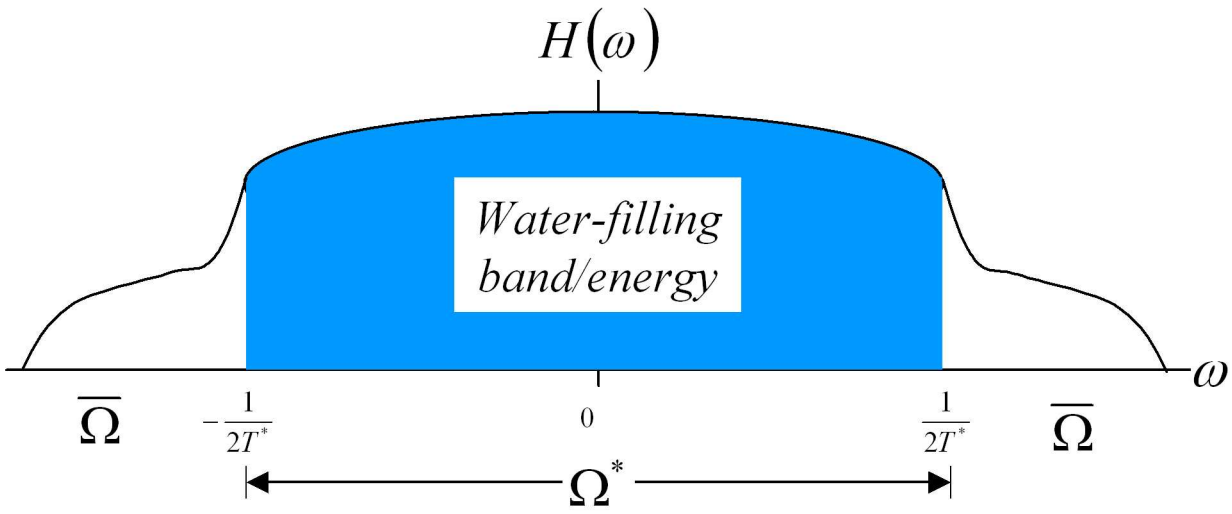


Figure 5.7: Illustration of water-filling and transmit optimization for real baseband lowpass channel.

where $S_x(\omega)$ is the transmit power spectral density with total power constraint

$$P_x = \frac{1}{2\pi}\int_{\Omega^*} S_x(\omega)d\omega \tag{5.51}$$

and $g(\omega)$ (in the continuous-time case) includes the effects of any continuous-time transmit filtering. The continuous water-filling solution is

$$S_x(\omega) = \lambda' - \frac{1}{g(\omega)} \quad , \tag{5.52}$$

where λ' is determined from the power constraint in (5.51). Clearly this continuous capacity in bits per second determines an upper attainable bound on the capacity of any discrete time approximation with symbol rate $1/T$. The next 3 subsections investigate the determination of best symbol rate(s) and best corresponding carrier frequencies.

5.2.3 The baseband lowpass channel with a single contiguous water-fill band

The baseband lowpass channel has the transfer characteristic of Figure 5.7. The continuous-time waterfill band Ω^* is also shown. Clearly sampling of this channel at any rate exceeding $1/T^* = |\Omega^*|$ should produce a discrete-time channel with capacity

$$\bar{C}(T < T^*) = \mathcal{C} \cdot T \quad , \tag{5.53}$$

where the explicit dependency of $\bar{C}$ in bits/dimension on T is shown in the argument $\bar{C}(T)$. When the channel is complex, then the relation is $\bar{C}(T < T^*) = \mathcal{C} \cdot T/2$. The capacity in bits/second remains constant while the capacity in bits per symbol decreases with increasing symbol rate to maintain the constant value $\mathcal{C} = C(\bar{T})/T$. At symbol rates below $1/T < 1/T^* = |\Omega^*|$, capacity $\mathcal{C}$ may not (and usually is not) be achieved so

$$\bar{C}(T > T^*) \leq \mathcal{C} \cdot T \quad . \tag{5.54}$$

To achieve the highest data rates on this lowpass channel with good codes, the designer would like to choose $1/T \geq 1/T^*$. However, the transmit filter is not realizable unless $1/T = 1/T^*$, so there is an optimum symbol rate $1/T^*$ for which

$$\mathrm{SNR}_{\text{mmse-dfe,u}}(T^*) = 2^{2\mathcal{C}T^*} - 1 \quad . \tag{5.55}$$

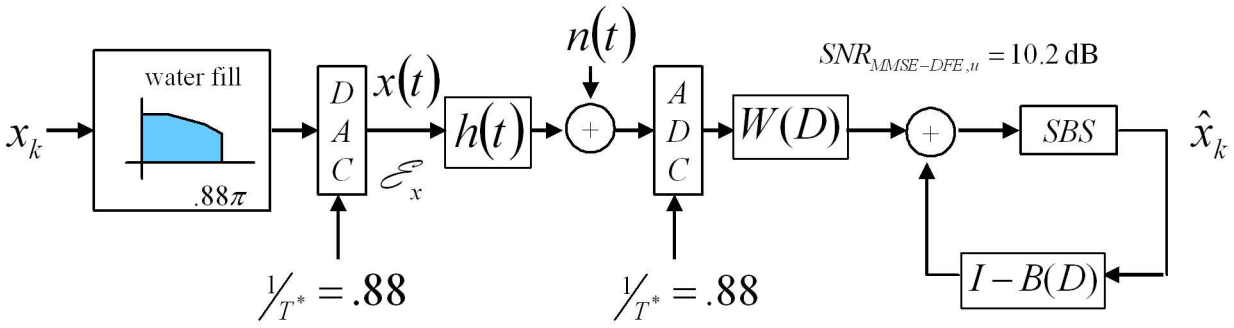


Figure 5.8: Block diagram of optimized DFE system for the $1 + .9D^{-1}$ channel.

At this symbol rate **only** can the MMSE-DFE can achieve the best possible performance and indeed match the MT (or DMT/VC) performance levels of Chapter 4. This example correctly suggests that an ideal symbol rate exists for each channel with "PAM/QAM-like" transmission and a MMSE-DFE receiver. For DMT/VC, any sampling rate $1/T' \geq 1/T^*$ is feasible for implementation, but only $1/T^*$ is feasible for the MMSE-DFE approach.

For rates below capacity where a code with gap $\Gamma > 1$ is used, the continuous-time water-filling can be solved with the transmit power P_x reduced by Γ. This would produce a slightly lower optimum symbol rate $1/T_\Gamma^* \leq 1/T^*$. This slightly lower symbol rate should then be used for transmission with gap Γ, and the bit rate of the MMSE-DFE system would be

$$R = \frac{\bar{b}}{T_\Gamma^*} = \frac{1}{2T_\Gamma^*} \log_2 \left(1 + \frac{\text{SNR}_{\text{mmse-dfe,u}}(T_\Gamma^*)}{\Gamma} \right) \quad . \tag{5.56}$$

Two examples are provided to illustrate the effects and terminology of symbol-rate optimization:

EXAMPLE 5.2.4 ($1 + .9D^{-1}$ channel) The $1 + .9D^{-1}$ ISI-channel has been revisited often, and is again revisted here again with $\bar{\mathcal{E}}_x \|p\|^2 / \frac{\mathcal{N}_0}{2}$=10 dB. This channel corresponds to the real baseband lowpass channel discussed above. The capacity in bits per real sample/dimension has been previously found in Example 4.4.1 to be $C(T = 1) = 1.55$ bits/dimension, so that $\mathcal{C} = 1.55$ bits/second with $T = 1$. This capacity corresponds to an overall MM SNR of 8.8 dB, but could not be directly realized with a MMSE-DFE, which instead had an $\text{SNR}_{\text{mmse-dfe,u}} = 8.4$ dB with the symbol rate of of $1/T = 1$. The optimum water-fill band has width $|\Omega^*| = .88\pi$ (positive frequencies). Then

$$1/T^* = .88 \quad . \tag{5.57}$$

Thus, the capacity in bits/dimension for the system with optimum symbol rate is

$$\bar{C}(T^*) = \frac{1.55}{.88} = 1.76 \text{ bits/dimension} \quad . \tag{5.58}$$

The resultant SNR is then

$$\text{SNR}_{\text{mmse-dfe,u}}(T^*) = \text{SNR}^*_{\text{mmse-dfe,u}} = 10.2 \text{ dB} \quad . \tag{5.59}$$

This optimum MMSE-DFE at this new symbol rate is the only MMSE-DFE that can match the MT system of Chapter 4 in performance (if there is no loss from error propagation or precoding). The SNR_{mfb} of 10 dB in previous invocations of this example is no longer a bound unless the symbol rate is $1/T = 1$ since that bound was derived for $T = 1$ and needs rederivation if another symbol rate is used. For verfication, the capacity in bits/second

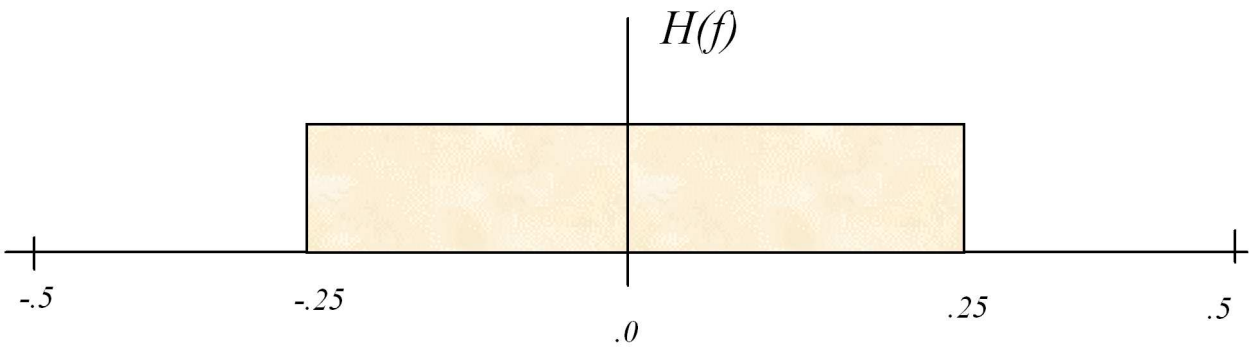


Figure 5.9: Example channel.

is consistently at both sampling rates, but $1/T^* = .88$ can be realized. The capacity in bits/second remains

$$\mathcal{C} = \frac{.88}{2}\log_2(1+10^{1.02}) = \frac{1}{2}\log_2(1+10^{.88}) = 1.55 \text{ bits/second} \quad . \tag{5.60}$$

EXAMPLE 5.2.5 (The "half-band" ideal lowpass channel) A brickwall lowpass channel has cut-off frequency .25 Hz as shown in Figure 5.9. Clearly the water-filling bandwidth is $\Omega^* = 2\pi(-.25,.25)$. The capacity is given as $\mathcal{C} = 2$ bits per second for this channel. Thus, at $T^* = 2$, then $\bar{C}(T^*) = 4$ bits/dimension and by the CDEF result, a MMSE-DFE with this symbol rate will have performance given by $\text{SNR}_{\text{mmse-dfe,u}}(T^* = 2) = 2^{2\cdot 4} - 1 \approx 24$ dB. The MMSE-DFE in this case is clearly trivial and the equalized channel is equivalent to the original ISI-free channel at this symbol rate.

Suppose instead, a designer not knowing (or not being able to know for instance in broadcast or one-directional transmission) the channel's Fourier transform or shape instead used $T = 1$ for a DFE with flat transmit energy. While clearly this is a poor choice, one could easily envision situations that approximate this situation on variable channels. An immediate loss that is obvious is the 3 dB loss in received energy that is outside the passband of the channel, which in PAM/QAM is equivalent to roughly to .5 bit/dimension loss. This represents an upper bound SNR for the MMSE-DFE performance.[6] The mutual information (which is now less than capacity) is approximately then

$$\bar{I}(T=1) \approx \frac{\bar{I}(T^*=2) - .5}{2} = \frac{3.5}{2} = 1.75 \text{ bits/dimension} \quad . \tag{5.61}$$

The .5 bit/dimension loss in the numerator $\bar{I}(T^* = 2) - .5 = \frac{1}{2}\log_2(1 + \frac{\text{SNR}}{2}) = 3.5$ approximates the 3 dB loss of channel-output energy caused by the channel response. More exactly $\bar{I}(T=1) = \frac{1}{4}\log_2\left(1 + \frac{\text{SNR}_{\text{mmse-dfe,u}}{}^{(T^*=2)}}{2}\right)$=1.746. Clearly, the channel has severe ISI when $T = 1$, but the CDEF result easily allows the computation of the MMSE-DFE SNR according to

$$\text{SNR}_{\text{mmse-dfe,u}}(T=1) = 2^{2\cdot 1.75} - 1 = 10.1 \text{ dB} \quad . \tag{5.62}$$

[6] In fact, this upper bound can be very closely achieved for very long filters in MMSE-DFE design, which could be verified by DFE design for instance according to the DFE program in Chapter 3.

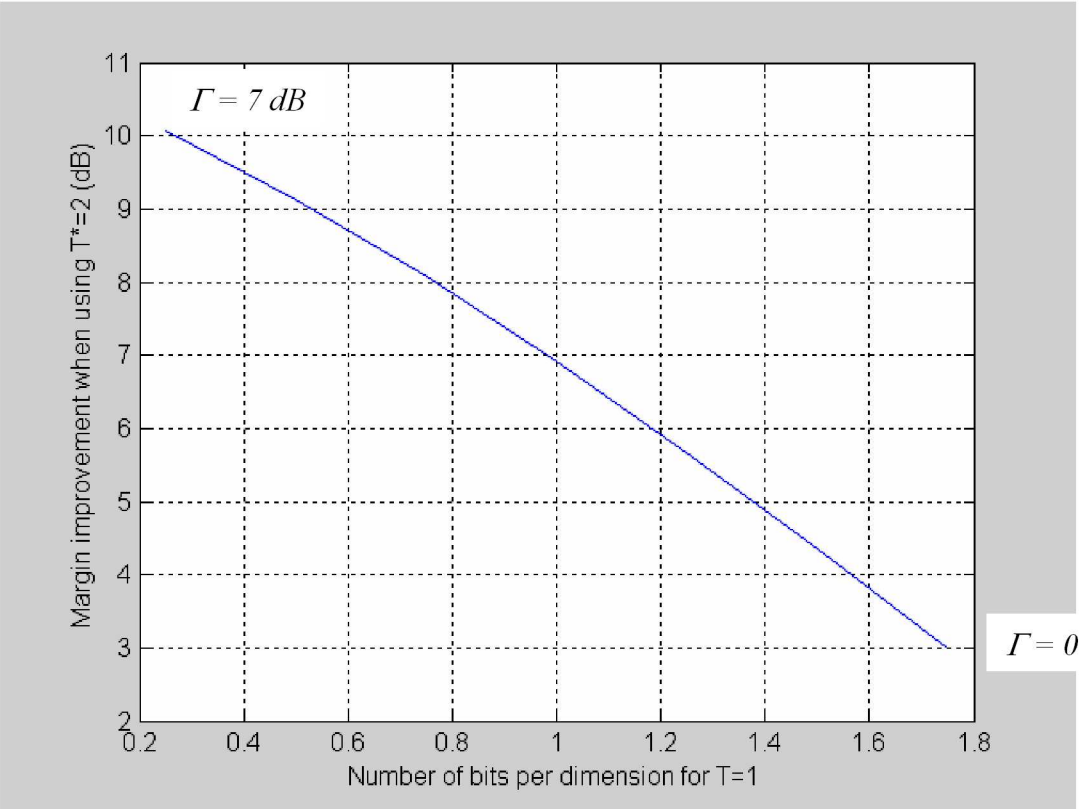


Figure 5.10: Margin difference (improvement) for optimized transmission over unoptimized MMSE-DFE on a "half-band" channel.

The equalized system with $T = 1$ is equivalent to an AWGN with SNR=10.1 dB, although not so trivially and with long equalization filters in this $T = 1$ case because ISI will be severe and filter lengths will be very long.

The system with $T = 1$ then attempts to transmit some $\bar{b}(T = 1)$ with an SNR of 10.1 dB, while the optimized system transmits $\bar{b}(T^* = 2) = 2\bar{b}(T = 1)$ with SNR of 24 dB. The margin for $T = 1$ is then for instance

$$\gamma_m(T = 1) = 10 \cdot \log_{10}\left(\frac{2^{2\cdot 1.75} - 1}{2^{2\bar{b}(T=1)} - 1}\right) = 10.1 \text{ dB} - 10 \cdot \log_{10}\left[2^{2\bar{b}(T=1)} - 1\right] \quad . \tag{5.63}$$

For $T^* = 2$, the margin is

$$\gamma_m(T^* = 2) = 10 \cdot \log_{10}\left(\frac{2^{2\cdot 2} - 1}{2^{2\bar{b}(T^*=2)} - 1}\right) = 24 \text{ dB} - 10 \cdot \log_{10}\left[2^{4\bar{b}(T=1)} - 1\right] \quad . \tag{5.64}$$

The ratio of the margins for $T = 1$ and $T^* = 2$ is then plotted in Figure 5.10 as a function of the number of bits per dimension. The difference decreases as the systems approach capacity, meaning codes with smaller gap are used, but is always nonzero in this case because the system with $T = 1$ essentially wastes 3 dB of bandwidth. Indeed, it is not possible for the $T = 1$ system to transmit beyond 1.75 bits/second and is thus at best 3 dB worse than the $T^* = 2$ system because half the energy is wasted on a channel band that cannot pass signal energy. One could infer that at fixed P_e, the gap decreases with increasing $\bar{b}(T = 1)$ on the horizontal access until the $\Gamma = 0$ dB limit is reached at 1.75 bps.

Some designers might attempt transmission with an approximation to the correct transmit filter, thus regaining some of the minimum 3 dB energy loss when $\Gamma = 0$ dB. As the transmit filter becomes more tight, it becomes difficult to implement - in the limit, such a filter is not realizable because it does not meet the PW criterion. However, leaving $T = 1$, no matter how complex the transmit filter, forces ISI and makes both transmitter and receiver very complex with respect to using the correct $T^* = 2$. As the gap increases, the performance difference is magnified between $T = 1$ and $T^* = 2$, so even a very good brickwall transmit filter at $T = 1$ loses performance when $\Gamma > 0$. The transmit filter is simple flat passband at $T^* = 2$. Also as the capacity/SNR increases, the curve in Figure 5.10 will show larger difference at reasonable data rates, but more rapidly fall to again 3dB at the point where $T = 1$ system achieves capacity with $\Gamma = 0$ dB. It is thus, very inadvisable design to use the incorrect symbol rate from the perspectives of performance or complexity.

5.2.4 The single-band bandpass case

The channels of the previous subsection (5.2.3) were baseband (i.e., real one-dimensional) lowpass channels. A complex lowpass channel tacitly includes the effect of a carrier frequency. Figure 5.11 illustrates the passband channel and the choice of best carrier (or "center") frequency f_c and then consequently the best symbol rate $1/T^*$. Clearly, to select the optimum symbol rate, the QAM carrier frequency must be in the center of the water-filling band. Any other choice of carrier frequency results in an asymmetric use of water-filling band frequencies around DC, which means that the PW criterion will not be satisfied.[7] Thus, for the MMSE-DFE on a passband channel, the carrier frequency must be centered in the water-filling band and the symbol rate is chosen to be the measure of the resulting (postive-frequency for passband case) continuous-time water-filling band. Again, a gap $\Gamma > 0$ dB causes a slight decrease in the water-fill band's measure and will also alter the carrier/center frequency slightly (unless the band is conjugate symmetric with respect to the choice of carrier/center frequency).

EXAMPLE 5.2.6 (V.34 Voiceband Modems) The International Telecommunications Standardized v.34 modem, better known as the "28.8 kbps" modem uses optimized decision feedback equalization. This modem initially trains by using multitone transmission

[7] Note Carrierless AMPM (CAP) systems of Chapter 1 do not use a carrier, but have a "center" frequency. This center frequency is optimized in exactly the same way as the carrier frequency in this subsection.

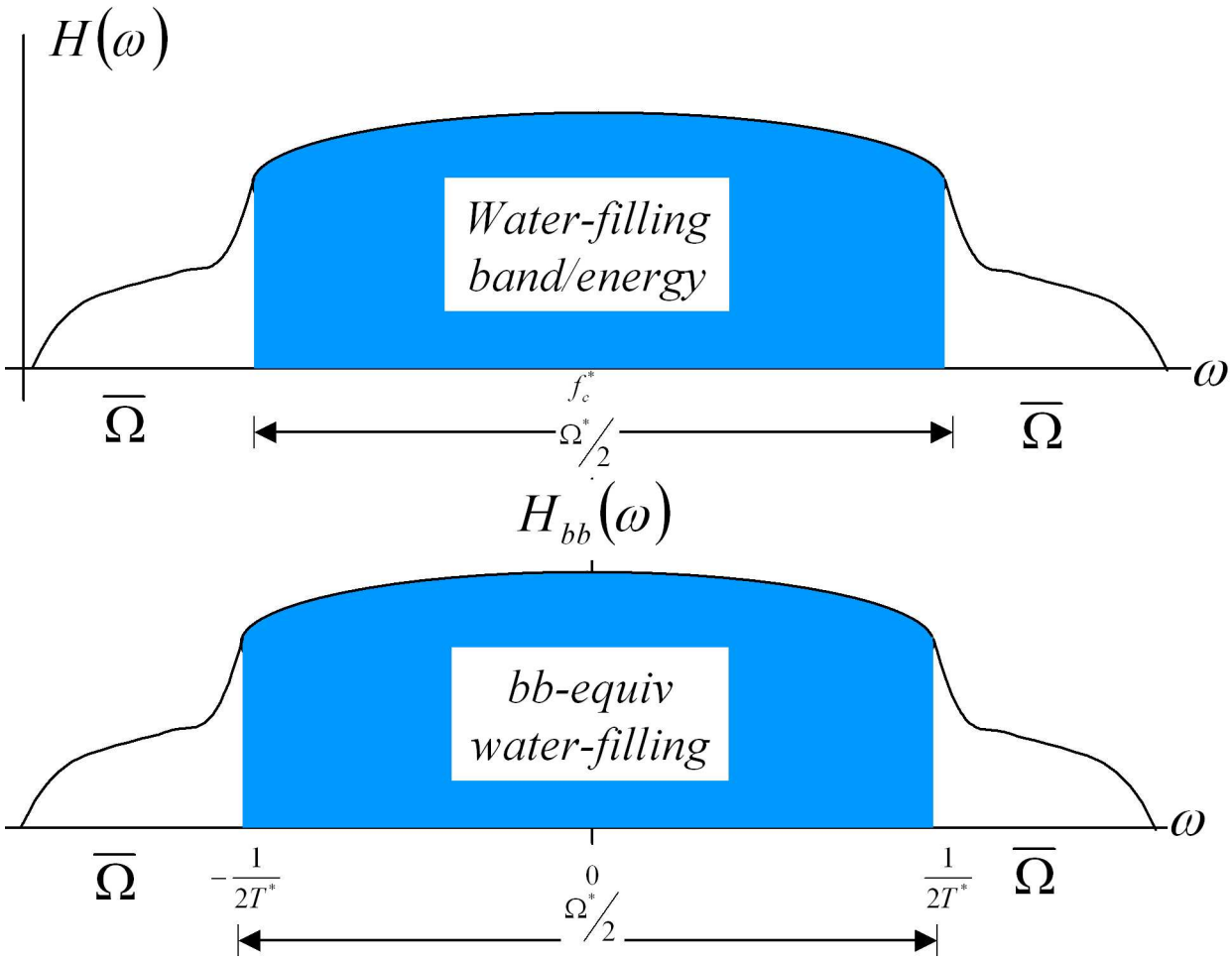


Figure 5.11: Illustration of water-filling and transmit optimization for ("complex") passband channel.

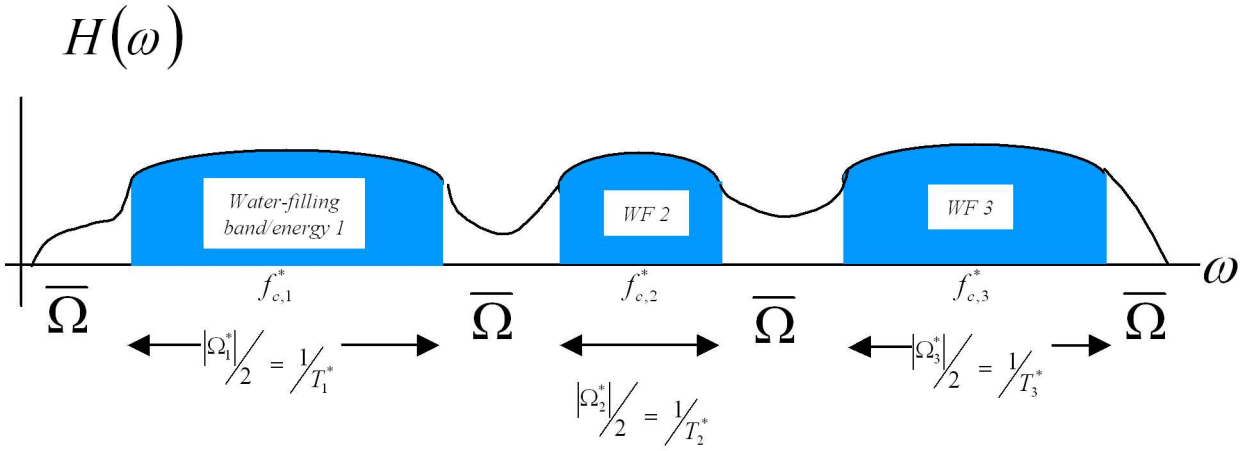


Figure 5.12: Illustration of water-filling and transmit optimization for a general channel with more than one passband.

technology, specifically 25 equally spaced tones are sent with fixed amplitude and phase at the frequencies $n(150Hz)$ where $n = 1...25$. The frequencies 900, 1200, 1800, and 2400 are silenced, leaving 21 active tones for which the SNR's are measured (and interpolated for 900, 1200, 1800, and 2400) Water-filling or other RA loading-like procedures determine an optimum bandwidth, which is reduced to a choice of symbol rate and carrier frequency for QAM. The choices for symbol rate and carrier must be from the attached table:

$1/T^*$	f_{c1}	f_{c2}
2400	1600	1800
2743	1646	1829
2800	1680	1867
3000	1800	2000
3200	1829	1920
3429	1959	1959

Only the symbol rates in the table may be selected and after one is selected, only two carrier frequencies are then allowed as denoted in the same row. there is never more than one disjoint WF band, and examination of the choices with respect to optimum has revealed a loss of less than 1 dB. Voiceband modems rarely have $M > 1$, so the issue of multiple band MMSE-DFE is not addressed in v.34. This special case of a single band being sufficient does not extend to DSL for instance, see Example 5.2.5.

5.2.5 The general multiple band case and the Optimum MMSE-DFE

Examples 5.7 and 5.11 were overly simple in that a single band of transmission was presumed. On ISI channels that exhibit notches (i.e., multipath fading or bridged-taps, multiple antennae, etc.) and/or band-selective noise (crosstalk, RF noise, etc.), Ω^* may consist of many bands. This subsection addresses these more practical cases to find a minimal set of MMSE-DFEs that can be analyzed as a single system.

Practical ISI-channels often have water-filling solutions that consist of a countable number of band-pass (and perhaps one lowpass) water-fill regions, which have nonzero transmit energy over a continuous set of frequencies within each region. To satisfy the PW criterion, the MMSE-DFE then must become multiple MMSE-DFE's, each with its own independent water-filling solution. This situation appears in Figure 5.12 for 3 disjoint water-fill regions. Correspondingly, there would be 3 MMSE-DFE's each operating at an $\text{SNR}_{\text{mmse-dfe,u}}$ for its band and at a data rate R_i for each band.

An overall SNR can be computed in this case of multiple bands. An overall symbol rate, presuming

carriers centered in each band, is defined

$$\frac{1}{T^*} = \sum_{i=1}^{M} \frac{1}{\bar{T}_i^*} \quad , \tag{5.65}$$

where $\bar{T}_i^* = (1 \text{ or } 2) \cdot T_i^*$ for complex passband and real baseband respectively. The SNR in each band is $\text{SNR}_i(T_i^*)$. Each band has data rate

$$R_i = \frac{1}{\bar{T}_i^*} \cdot \log_2 \left(1 + \frac{\text{SNR}_i(T_i^*)}{\Gamma}\right) \quad . \tag{5.66}$$

Then, the overall data rate is in bits per second is

$$\begin{aligned} R &= \sum_{i=1}^{M} R_i & (5.67)\\ &= \sum_{i=1}^{M} R_i = \sum_{i=1}^{M} \frac{1}{\bar{T}_i^*} \cdot \log_2 \left(1 + \frac{\text{SNR}_i(T_i^*)}{\Gamma}\right) & (5.68)\\ &= \frac{T^*}{T^*} \log_2 \prod_{i=1}^{M} \left(1 + \frac{\text{SNR}_i(T_i^*)}{\Gamma}\right)^{1/\bar{T}_i^*} & (5.69)\\ &= \frac{1}{T^*} \log_2 \prod_{i=1}^{M} \left(1 + \frac{\text{SNR}_i(T_i^*)}{\Gamma}\right)^{T^*/\bar{T}_i^*} & (5.70)\\ &= \frac{1}{T^*} \log_2 \left(1 + \frac{\text{SNR}^*_{\text{mmse-dfe,u}}(T^*)}{\Gamma}\right) & (5.71)\\ &= \frac{b^*}{T^*} \quad , & (5.72) \end{aligned}$$

where

$$\text{SNR}^*_{\text{mmse-dfe,u}} \triangleq \Gamma \cdot \left\{ \left[\prod_{i=1}^{M} \left(1 + \frac{\text{SNR}_i(T_i^*)}{\Gamma}\right)^{T^*/\bar{T}_i^*} \right] - 1 \right\} \tag{5.73}$$

and

$$\bar{b}^* \triangleq \frac{1}{2} \cdot \log_2 \left(1 + \frac{\text{SNR}^*_{\text{mmse-dfe,u}}}{\Gamma}\right) \text{ bits/dimension} \quad . \tag{5.74}$$

When all symbol rates and carrier frequencies are optimized, clearly this SNR is the same as MM and MT systems of Chapter 4 because the rates/capacities for each water-filling band are the same. However, the MMSE-DFE now is often many MMSE-DFE's, and each has a distinct variable carrier frequency and symbol rate. The reader is cautioned against the statements often encountered by as-yet-uninformed engineers who misinterpret the CDEF result to mean that a "single-carrier system performs the same as a multicarrier system" – misquoting the CDEF result. Such a statement is only true in the simplest cases where a single water-fill band is optimum, and the symbol rate and carrier frequencies have been precisely chosen as a function of the specific channel's water-filling solution and coding gap. Otherwise, multicarrier will outperform single carrier – and further, single carrier can never outperform multicarrier.

Lemma 5.2.1 (The Optimum MMSE-DFE) *The optimum MMSE-DFE is a set of M independent DFE's with M equal to the number of disjoint water-filling bands. Each of the MMSE-DFE's must have a symbol rate equal to the measure of a continuous water-fill band $1/T_m^* = |\Omega_m^*|$ and a carrier frequency where appropriate set exactly in the middle of this band ($f_{c,m} = f_{max,m} + f_{min,m}/2 \; \forall \; m = 1, ..., M$). The number of bits per dimension for each such MMSE-DFE is*

$$\bar{b}_m = \frac{1}{2} \cdot \log_2 \left(1 + \frac{SNR_m(T_m^*)}{\Gamma}\right) \quad . \tag{5.75}$$

The overall SNR is provided in (5.73) and the (5.74)

A final simplified VDSL example provided to illustrate both proper design and a performance loss that might be incurred when designers try "to simplify" by only using one fixed band.

EXAMPLE 5.2.7 (Simple VDSL and lost "dead" bands) Figure 5.13 illustrates equivalent transmitters based on a multitone design of 8 equal bandwidth modulators. All tones are presumed passband uncoded QAM with gap $\Gamma = 8.8$dB. This system choses symbol rates to approximate what might happen on a short telephone line of 1 km in length, and is simplified with respect to what actually happens in practice (where more than 2 bands might occur in water-filling). The transmit filter for each band is denoted by $H_n(f)$ in set A and $G_n(f)$ in set B. The g_n's for these example subchannels are given in the table below.

Application of water-filling with 11 units of energy produces channel SNR's appear in the following table for a Gap of $\Gamma = 8.8$dB and a water-filling constant of $\lambda' = 2$:

n	g_n	$\mathcal{E}_n$	b_n
1	15.2	1.50	2
2	30.3	1.75	3
3	121.4	1.9375	5
4	242.7	1.97	6
5	2	0	0
6	60.7	1.875	4
7	242.7	1.97	6
8	2	0	0

Channel characteristics are such in this system that the optimum bandwidth use is only 6 of the 8 subchannels. 4 of the used bands (set A) are adjacent and another 2 of the bands (set B) are also adjacent. Sets A and B, however, are separated by an unused band. That unused band might be for example caused by a radio interference or perhaps notching from a bridged-tap, and its position is likely to vary depending on the line. The corresponding receiver is shown in Figure 5.14.

First Multitone Design:
The tones in Set A can carry ($2\bar{b}_n$ =) 2, 3, 5, and 6 bits respectively corresponding to increasing signal-to-noise ratios in the corresponding channel, while the two bands in Set B can carry 4 and 6 bits respectively. The unused bands carry 0 bits. The average number of bits per tone in Set A is 4 while the average number in Set B is 5. If the symbol rate for each subchannel is 1 MHz, then the data rate is (2+3+5+6+4+6) bits transmitted one million times per second for a total data rate of 26 Mbps. Equivalently, from Equation 5.65, the overall symbol rate is $1/T^* = 6$ MHz, and there are thus $\bar{b}(1/T^* = 6\text{MHz}) = \frac{26\,\text{Mbps}}{2(1/T^*)} = 26/[2(6)] = 2.16$ bits/dim or 4.33 bits per equivalent tone. For an uncoded-$\Gamma = 8.8$dB multitone system $\bar{b}(1/T' = 8\text{MHz}) = 26/16 = 1.625$. This rate is equivalent to $\bar{b}(1/T^*) = 6\text{MHz}) = 2.16$ since both correspond to the same overall data rate of 26 Mbps. The SNR at a symbol rate of $1/T = 8$ MHz is

$$\text{SNR}_{mt}(1/T' = 8\text{MHz}) = 10 \cdot \log_{10} \Gamma \cdot \left(2^{3.25} - 1\right) = 18.1\text{dB} \ . \tag{5.76}$$

The SNR at a symbol rate of $1/T^* = 6$ MHz is

$$\text{SNR}^*_{\text{mmse-dfe,u}}(1/T* = 6\text{MHz}) = 10 \cdot \log_{10} \Gamma \cdot \left(2^{4.33} - 1\right) = 21.6\text{dB} \tag{5.77}$$

The two multitone SNRs are equivalent in that both correspond to a $P_e = 10^{-6}$ at the data rate of 26 Mbps.

2nd Design: Equalivalent Two-tone QAM Design:
The corresponding best QAM receiver uses the same bands and is actually 2 QAM systems in this example. The first system has a symbol rate of $1/T_1^* = 4$ MHz and carries 4 bits per

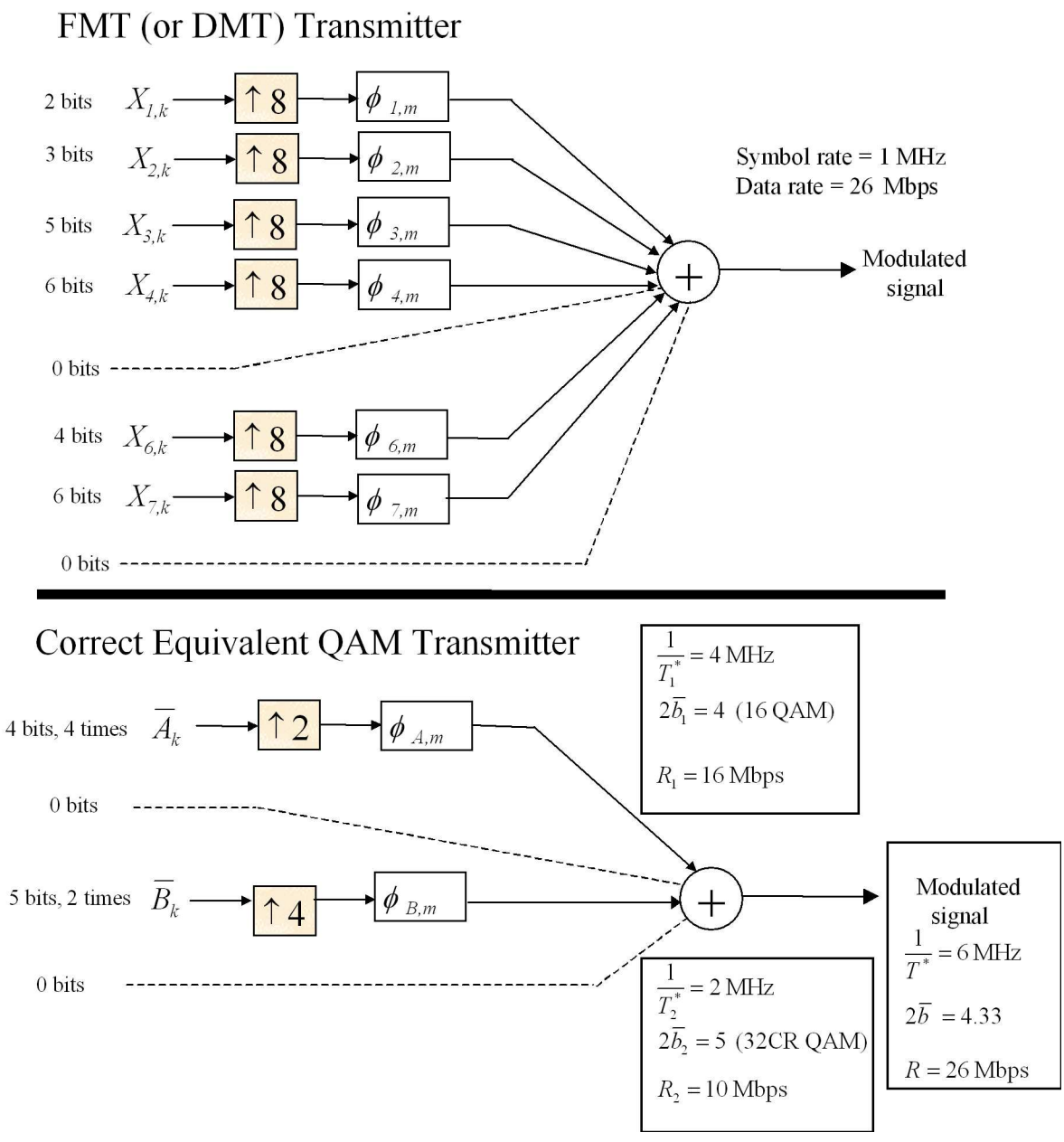


Figure 5.13: Simplified VDSL example multitone and equivalent 2-channel transmitters.

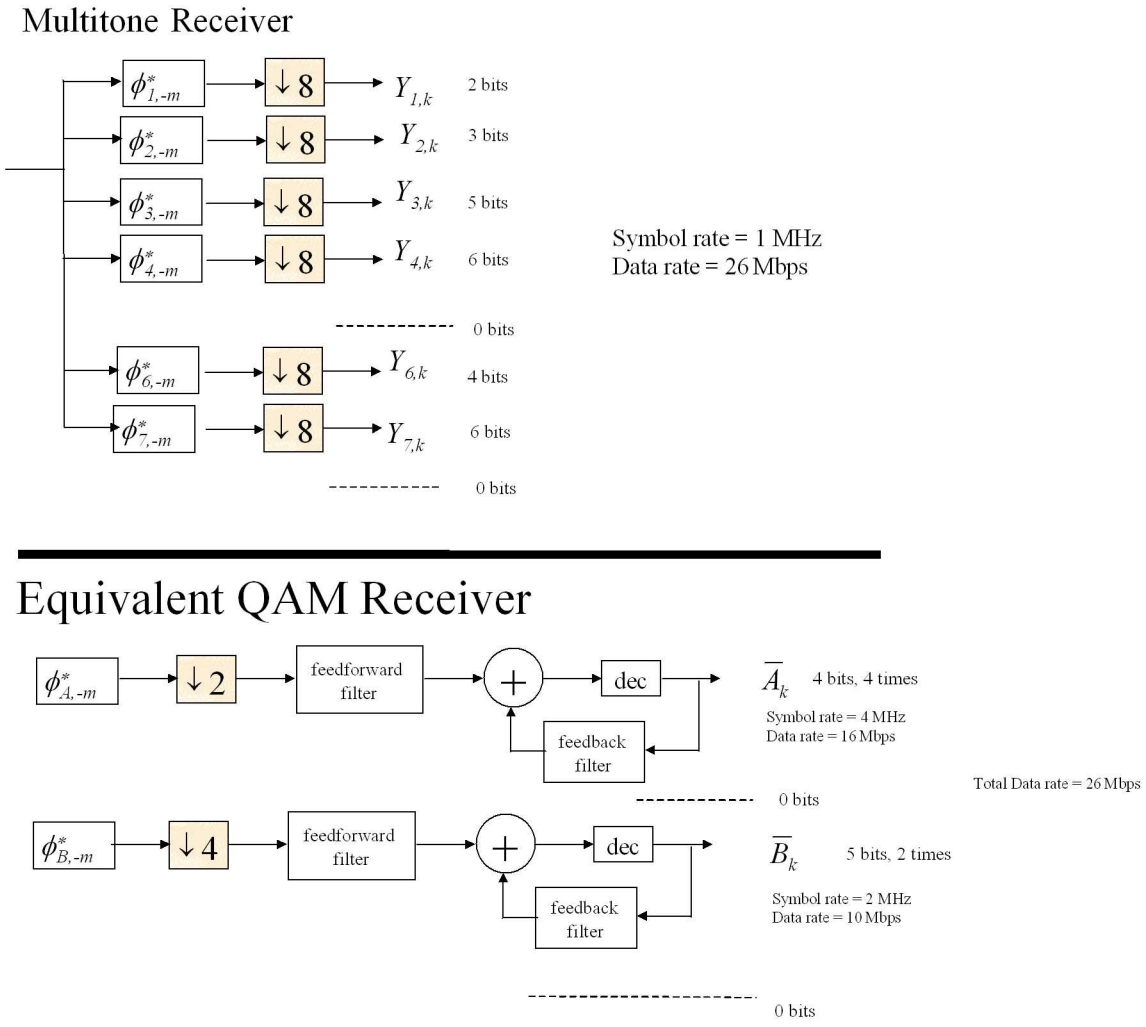


Figure 5.14: Simplified VDSL example multitone and equivalent 2-channel receivers.

QAM symbol (16 QAM) for a data rate of 16 Mbps. By the CDEF result, the MMSE-DFE receiver has an SNR of $\text{SNR}_{dfe,1}(1/T_1^* = 4\text{MHz}) = \Gamma \cdot (2^4 - 1)$=20.6 dB. The second system has symbol rate $1/T_2^* = 2$ MHz, and 5 bits per QAM symbol, which corresponds to 10 Mbps and $\text{SNR}_{dfe,2}(1/T_2^* = 2\text{MHz}) = \Gamma \cdot (2^5 - 1)$=23.7 dB. The total data rate of 26 Mbps is the same in both systems. The two systems carry the same data rate of 26 Mbps with a probability of symbol error of 10^{-6}.

Correct design thus finds a 2-band equivalent of the original 8-tone MT system, but absolutely depends on the use of the 2 separate QAM signals in the lower portions of Figures 5.13 and 5.14.

3rd Design - single-tone QAM designer:
Some designers might instead desire to see the loss of the single-band DFE system. The single-band (with $1/T = 8$ MHz) QAM system with MMSE-DFE then uses $2\bar{\mathcal{E}}_{\boldsymbol{x}} = 11/8 = 1.375$ at all frequencies and thus has

$$\text{SNR}_{dfe,flat} = \Gamma \cdot \left\{ \left[\prod_{n=1}^{8} \left(1 + \frac{1.375 \cdot g_n}{\Gamma} \right) \right]^{1/8} - 1 \right\} \tag{5.78}$$

$$= \quad 17\text{dB} \tag{5.79}$$

which is about 1.1 dB below the multitone result. If Tomlinson Precoding was used at the $\bar{b} = 1.625$, the loss is another .4 dB, so this is about 1.5 dB below the multitone result in practice. This $1/T = 8$ MHz single-tone QAM does then perform below MT because the two QAM tones are required to match MT performance.

As a continued example, this same transmission system is now applied to a longer line for which the upper frequencies are more greatly attenuated and $g_6 = g_7 \leq 2$, essentially zeroing the water-fill energy in band B. The multitone system might attempt 16 Mbps data rate. Such a system could reload the 1.875+1.97 units of energy on tones 6 and 7 to the lower-frequency tones, and the new SNR_{mt} will be 14.7 dB, which is a margin of 1.2 dB with respect to 16 Mbps ($2\bar{b} = 2$ requires 13.5 dB). A new computation of the single-tone ($1/T = 8$ MHz) SNR according to (5.78), which now has only 4 terms in the product provides $\text{SNR}_{DFE}(1/T = 8\text{MHz}) = 12.8$ dB that corresponds for the same data rate of 16 Mbps to a margin of -.7 dB ($\gamma = \frac{\text{SNR}_{DFE}}{\Gamma \cdot 2^{2\bar{b}}-1}$). The MT system (even without perfect loading here, because the margin-adaptive scheme was not used and the design instead approximated by boosting all the remaining tones equally) is 1.9 dB better. Furthermore, precoder loss would be 1.3 dB for a Tomlinson or Laroia precoder, leading to a 3.2 dB difference. Thus there is always a loss when the single-band system is used, and the amount of loss increases with the inaccuracy of the symbol-rate with respect to best symbol rate.

Since the margin of the QAM system is negative on the long line, the practice in industry is to "design for the worst case" long line, which means a better symbol rate choice would be 4 MHz. In this case, on the long line, both MT and QAM at 16 Mbps would have the SNR of the A band, 20.6 dB, increased by (11/(11-1.97-1.875) = 1.86 dB) to 22.5 dB and a margin of 1.86 dB at 10^{-6} probability of error with $\Gamma = 8.8$ dB. However, if the short line with bands A AND ALSO B is now again encountered, the 4 MHz-worst-case-designed QAM system will remain at 1.86 dB margin at 16 Mbps. The MT system margin at 16 Mbps now improves to (Equation 4.7 in Chapter 4)

$$\gamma_{mt} = \frac{2^{2\bar{b}_{max}} - 1}{2^{2\bar{b}} - 1} = \frac{2^3.25 - 1}{2^2 - 1} = 4.5\text{dB} \tag{5.80}$$

or a 2.6 dB improvement in margin over the 1.9 dB margin of the QAM system.

As the used-bandwidth error grows, the loss of a single-band DFE increases, but is hard to illustrate with small numbers of tones. In DSL, used bandwidth ratios can easily vary by a factor of 10 on different lines, increasing the deviation (but not easily depicted with a few tones) between MT systems and DFE systems that use a fixed bandwidth (or single tone). For instance, if only tone 4 were able to carry data that the best rate-adaptive solution is about 8.5 Mbps, while the full-band single-QAM DFE would attain only 420 kbps, and performs approximately 7 dB worse (be careful to avoid gap approximation margin calculations when number of bits per dimension is less than 1).

For real channels, independent test laboratories hosted a "VDSL" Olympics in 2003 for comparing DMT systems with variable-symbol-rate QAM systems. The test laboratories were neutral and wanted to ascertain whether a single-carrier system (so only one contiguous band that could be optimized and placed anywhere) really could "get the same performance as DMT" – an abuse of the CDEF result promulgated by single-carrier propopents). The results and channels appear in Figures 5.15 and 5.16 respectively. Clearly these fixed-margin (6 dB) tests for length of copper twisted pair at the data rate shown indicate that over a wider range of difficult channels as in DSL, the differences between DMT and single-carrier can be quite large. Generally speaking, the more difficult the channel, the greater the difference.

This section has shown that proper optimization of the MMSE-DFE may lead to several MMSE-DFEs on channels with severe ISI, but that with such a minimum-size set and properly optimized symbol rate and carrier frequency for the corresponding transmitter for each such MMSE-DFE determines a

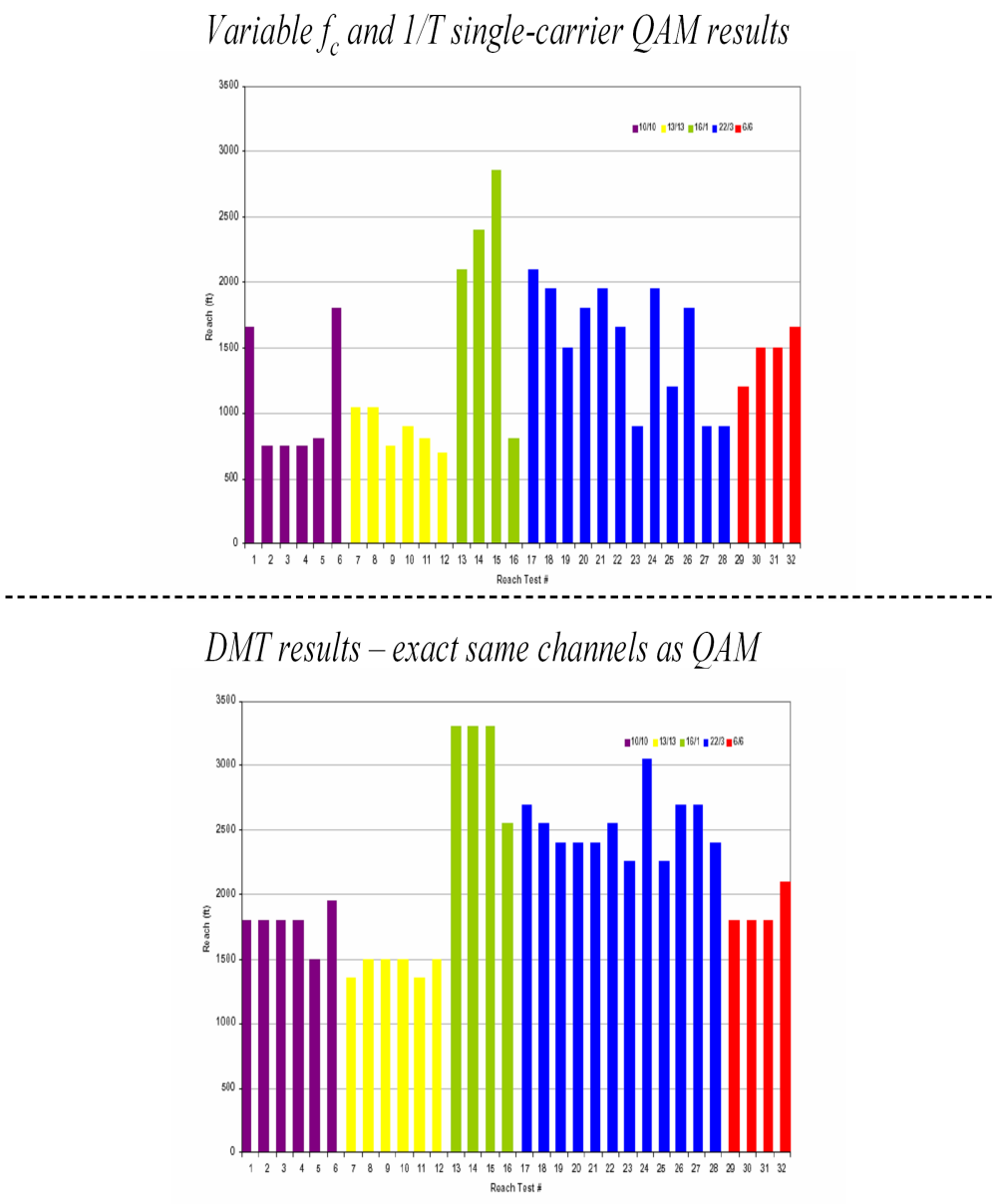


Figure 5.15: Illustration of 2003 VDSL Olympics results.

Config	Test	Service (Mbit/s) (Down/Up)	Reach (ft)
1	1	10/10	1800
2	2	10/10	1800
	3	10/10	1800
	4	10/10	1800
	5	10/10	1500
3	6	10/10	1950
4	7	13/13	1350
	8	13/13	1500
5	9	13/13	1500
6	10	13/13	1500
7	11	13/13	1350
	12	13/13	1500
8	13	16/1	3300
	14	16/1	3300
9	15	16/1	3300
	16	16/1	2550
10	17	22/3	2700
	18	22/3	2550
	19	22/3	2400
	20	22/3	2400
	21	22/3	2400
11	22	22/3	2550
	23	22/3	2250
12	24	22/3	3150
	25	22/3	2250
	26	22/3	2700
13	27	22/3	2700
	28	22/3	2400
14	29	6/6	1800
	30	6/6	1800
	31	6/6	1800
15	32	6/6	2100

Figure 5.16: Channel types and speeds for 2003 VDSL Olympics.

canonical transmission system, then matching the DMT/VC like systems of chapter 4. One then begins to question which might be best used if both are canonical. That question requires investigation of finite-length (finite N) designs and focuses on complexity. Sections 5.3 and beyond investigate this area as well as generalize the MMSE-DFE and DMT/VC into a very general type of interference channel. The reader will see that the DMT system is not only canonical, but also optimal, and indeed has the least complexity (using FFTs) of any of the systems if all properly designed at the same performance level.

5.2.6 Relations Zero-Forcing and MMSE DFE receivers

In some special situations, the ZF-DFE and the MMSE-DFE can induce the same equivalent channel, although the performance will not be the same. Some caution needs to be exercised here as they are not always the same, and some authors [8] have made some serious errors in interpretting results like CDEF.

Theorem 5.2.2 (CDEF Derivative) Over any of the continuous bands of water-filling, *if the water-fill spectra are used, then the ZF-DFE and MMSE-DFE result in the same channel shaping before bias removal.*

proof: This proof uses continuous bands, presuming the condition for choice of optimum sampling rate(s) and carrier frequencies have already been made so that all frequencies are used with non-zero energy (except possibly for countable set of infinitessimally narrow notches) – that is the PW criterion is satisfied. The power spectrum for an ISI channel for which the symbol/sampling rate has been altered for exact match to one (any) water-filling band $1/T^*$ that must satisfy

$$\bar{\mathcal{E}}_{\boldsymbol{x}}|\Phi(\omega)|^2 + \frac{\sigma^2}{|\tilde{H}(\omega)|^2} = K \tag{5.81}$$

for all frequencies ω in the band (zeros are only allowed at infinitesimally narrow notch points, so resampling is necessarily to avoid singularity). $\tilde{H}$ represents the equivalent-white-noise channel $\tilde{H} = \frac{H}{\sqrt{\bar{S}_n}}$. The MMSE-DFE fundamentally is determined by canonical factorization, which in the frequency domain has magnitudes related by

$$\bar{\mathcal{E}}_{\boldsymbol{x}}|\Phi(\omega)|^2 \cdot |H(\omega)|^2 + \sigma^2 = \frac{\gamma_0}{\|p\|^2} \cdot |G(\omega)|^2 \quad . \tag{5.82}$$

Insertion of (5.81) into (5.82) yields

$$K \cdot |H(\omega)|^2 = \frac{\gamma_0}{\|p\|^2} \cdot |G(\omega)|^2 \quad , \tag{5.83}$$

illustrating a very unusual event that the overall equivalent channel has the same shape as the channel itself. This happens when the input is water-filling. Thus, since the shape of H and G are the same, they differ only by an all-pass (phase-only) filter. Equivalently, the receiver's MS-WMF is essentially Φ^* times an all-pass filter. Such an all-pass can be implemented at the transmitter without increasing the transmit energy, leaving only the filter $\bar{S}_n^{-1/2} \cdot \Phi^*$ in the receiver. Such an all-pass phase shift never changes the performance of the DFE (or any infinite-length equalizer). The receiver's remaining noise whitening filter and matched filter to transmit water-fill filter constitute the entire MS-WMF, perhaps resulting in an opportunity to reduce receiver complexity. The MMSE-DFE has slightly non-unity gain through the feed-forward filter and the overall channel gain is $\frac{\text{SNR}_{\text{mmse-dfe,u}}}{\text{SNR}_{\text{mmse-dfe,u}^{+1}}}$, which is

[8] One somewhat incorrectly derived result is due to Price of MIT in early 70's where he tacitly assumed zero noise, and then water-filling of course always uses the entire band of transmission. Thus Price accidentally considered only one transmission band that was this entire band. Of course, trivially, ZF and MMSE are the same when there is no noise – this is hardly surprisingly and he erroneously arrived at a result resembling CDFE that is correct only when there is zero noise. This is sometimes referred to as "Price's result," but it is clearly Price did not understand many of the fundamental issues and concepts at the time of his conclusion.

removed by anti-bias scaling. However, ignoring the bias removal and consequent effect on signal and error sequence, water-filling creates an equivalent overall channel that has the same shaping of signal as the original channel would have had. That is the signal (ignoring bias in error) has the same minimum-phase equivalent ISI as the original channel would have, or equivalently as a ZF-DFE would have generated. **QED.**

The MMSE-DFE however does have a lower MMSE and better SNR, even after bias removal. The difference in SNR is solely a function of the error signal and scaling. The ZF-DFE has white noise and no bias. The MMSE-DFE with water-filling has the same minimum-phase shaping, that is achieved by filtering the signal, reducing noise power, but then ensuring that the components of the error (including bias) add in a way that the original channel reappears as a result of all the filtering.

The minimum-phase equivalent of the channel alone H is equal to G via 5.83. However $P_c(D)$, the canonical equivalent of the pulse response $\Phi \cdot H$ (which includes the water-filling shaping) is not the same unless water-filling shape is the special case of being flat. The precoder settings or equivalently unbiased feedback-section settings are not necessarily determined by G, but by G_u. G_u differs from the minimum-phase channel equivalent.

Thus, the MMSE-DFE results in same equivalent channel as ZF-DFE under water-filling, but still performs slightly better unless water-filling is flat (essentially meaning the channel is flat). Often water-filling is very close to flat in practical situations or there is very little loss. Thus, it may be convenient to implement directly a ZF-DFE once the exact water-filling band of frequencies is known.

Theorem 5.2.3 (Worst-Case Noise Equivalence) Over any of the continuous bands for which the PW Criterion holds, *if the noise power spectral density is the worst possible choice that minimizes mutual information for the channel, then ZF and MMSE are the same even if the noise is not zero.*

proof: The minimization of the mutual information

$$I = \frac{1}{2\pi} \int_{\Omega} \log_2 \left(1 + \frac{S_x(\omega) \cdot |H(\omega)|^2}{S_n(\omega)} \right) d\omega \tag{5.84}$$

with noise power constrained at

$$\sigma^2 = \frac{1}{2\pi} \int_{\Omega} S_n(\omega) d\omega \tag{5.85}$$

leads easily by differentiation with LaGrange constraint to the equation/solution

$$\frac{1}{S_x(\omega) \cdot |H(\omega)|^2 + S_n(\omega)} - \frac{1}{S_n(\omega)} = K \; \forall \; \omega \in \Omega \quad . \tag{5.86}$$

The magnitude of the MS-WMF with noise whitening included is

$$\begin{aligned} |FF|^2 &= \frac{|H(\omega)|^2 \cdot \frac{S_x(\omega)}{\mathcal{E}_{\boldsymbol{x}}}}{S_n^2(\omega) \cdot \gamma_0^2 \cdot \|p\|^2 |G(\omega)|^2} && (5.87) \\ &= \frac{|H(\omega)|^2 \cdot S_x(\omega)}{(S_x(\omega) \cdot |H(\omega)|^2 + S_n(\omega)) \cdot S_n(\omega)} && (5.88) \\ &= \frac{1}{S_n(\omega)} - \frac{1}{S_x(\omega) \cdot |H(\omega)|^2 + S_n(\omega)} = -K && (5.89) \end{aligned}$$

Thus, the MS-WMF section is exactly an all pass filter, meaning then ZF and MMSE-DFEs (the latter with bias-removal scaling) are the same. **QED.**

The above result holds for any input spectrum $S_x(\omega)$ (and clearly the worst-case noise may vary with this input spectrum choice). In particular, this result does hold for the water-filling spectrum also. The latter result has broad implications in multi-user optimization of systems as in Chapter 14. It suggests that if the noise is chosen to be the worst possible spectrum (for a given power), receiver processing is unnecessary (as the all-pass could be implemented at the transmitter without loss and a precoder could also be used). Worst-case noise (unlike water-filling transmit spectra) is not good – it means that receiver filtering is essentially useless.

5.3 Generalized Channel Partitioning: The GDFE

Sections 5.1 and 5.2 suggest that both a set of MMSE-DFE's and a MT transmission design can achieve the same data rate of $\bar{I}(X(D);Y(D))$, which is the maximum data rate for the given spectrum of $X(D)$ and for a gap of zero dB. The MT system partitions the channel into a infinite number of AWGN subchannels that each carry a different amount of information, while by contrast eaqch of the set of MMSE-DFE systems (with correct decisions) partitions the channel into an infinite set of AWGN subchannels that each carry the same amount of information. However, the sum of the data rate of the DMT subchannels is then equal to the sum of the data rates of each of the equal-information/dimesion DFE systems. The observation of two canonical-performance partitions suggests that more general partitioning of an ISI channel exists.

This section investigates partitioning for the finite-length packet channel characterized by the relationship

$$\boldsymbol{y} = P\boldsymbol{x} + \boldsymbol{n} \tag{5.90}$$

with nominal output dimensionality N and input dimensionality $N + \nu$ (as in Chapter 4), although all developments will hold for more general P of any dimensionality and any structure (so one could think of ν as possibly zero or negative in any of the ensuing developments (as well as the usual case of $\nu \geq 0$). In fact, the general P allows modelling of wireless systems where the inputs and outputs may correspond to different antennas, and not just successive time samples as in Chapters 3 and 4. Further the Toeplitz structure of P need not be maintained. One general choice of P models time-varying convolutions where the successive rows of P are indented as with convolution, but each row varies with respect to previous row proceeding down through the matrix. Such time-varying models may be useful in wireless mobile transmission systems. Other models of P could model crosstalk between different antennas in multiple-input-mulitple-output wireless transmission. Yet other models would be used for crosstalk between wires sharing a common physical space (like a binder of telephone wires or an ethernet cable). In short, the GDFE development allows any P and finds the corresponding canonical-performance partitions, opening new vistas to the concept of "equalization" extending to all kinds of linear interference and canonically providing designs for them. Vector coding of Chapter 4 also did not restrict P to have any structure (although DMT forced a cyclic structure), and thus could also be immediately generalized to any P with no change to its Chapter-4 development.

For vector coding on such a channel, the product of the subchannel SNR's can be computed (for $\Gamma = 0$dB) as

$$\text{SNR}_{vc,u} + 1 = \left[\prod_{n=1}^{N^*} \left(1 + \frac{\mathcal{E}_n \cdot \lambda_n}{\frac{\mathcal{N}_0}{2}}\right)\right]^{\frac{1}{N+\nu}} = \frac{|R_{\boldsymbol{yy}}|}{|R_{\boldsymbol{nn}}|} = 2^{2I(\bar{\boldsymbol{x}};\boldsymbol{y})} \quad . \tag{5.91}$$

Since VC has the optimum finite-length partitioning for any energy distribution on the subchannels, or equivalently attains the highest rate, then clearly $\bar{I}(\boldsymbol{x};\boldsymbol{y})$ is this highest rate. Any one to one transformation of either $\boldsymbol{x}$ or $\boldsymbol{y}$ to another random vector is reversible and cannot change the mutual information. This observation leads allows discovery of other structures and sets of parallel channels that perform at the same highest rate for the given input energy distribution (or given input autocorrelation function $R_{\boldsymbol{xx}}$).

Figure 5.17 overviews the concept of generalized partitioning. There are an infinite number of partitionings that all can achieve the same aggregate data rate $b = I(\boldsymbol{x};\boldsymbol{y})$ over a set of parallel independent AWGN channels. The partitionings can be described and summarized through the concept of Generalized Decision Feedback Equalization, or the GDFE for short. VC (and its special case DMT with cyclic prefixes) are the simplest and most natural of the partitionings, but many exist. A particularly interesting partitioning is Circulant Decision Feedback, or the CDFE. It is this latter structure, which under an assumption of stationarity of the channel over all sampling periods, will converge (with various care about the PW criterion) to a set of MMSE-DFEs. VC, under the same stationarity assumption, converges to MT. Some of the more general decompositions for the GDFE can be of interest in situations where the channel will never be stationary over all dimensions; for instance multi-dimensional communication systems that may have antenna arrays, two-dimensional scans of surfaces or "image-like" information, or channels that vary with time – these are sometimes called multiple-input, multiple-output (MIMO)

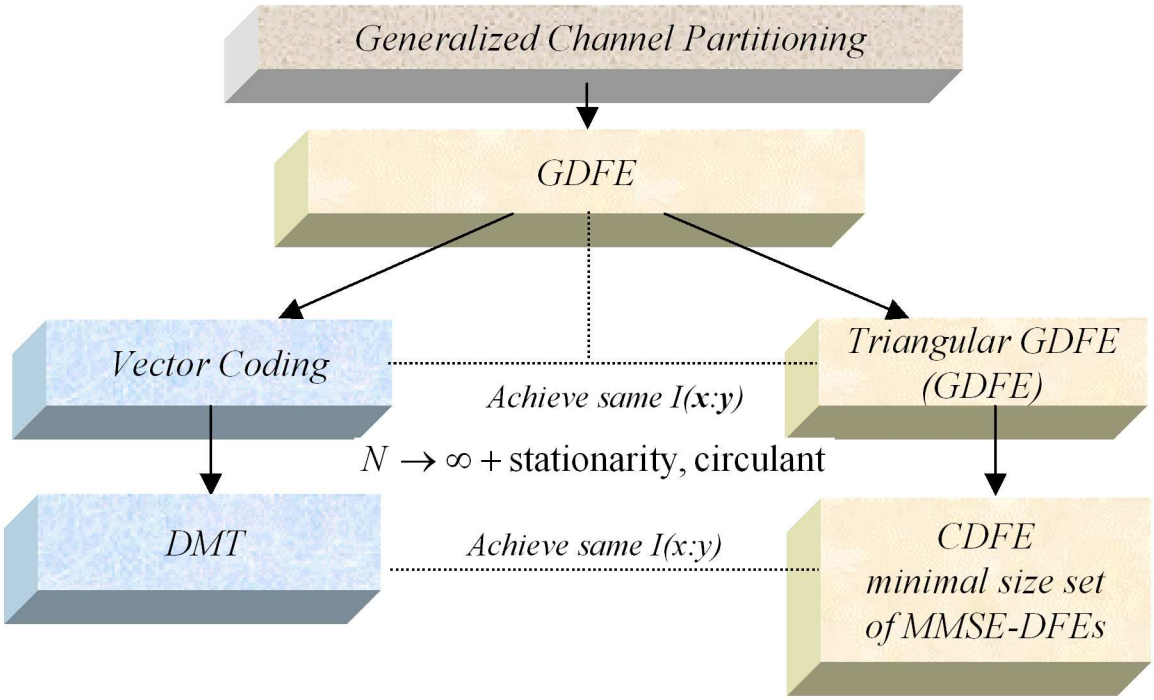


Figure 5.17: Generalized partitioning.

transmission systems, or even a set of crosstalking telephone lines bundled together. These types of matrix channels will possibly lead to P matrices that are not "Toeplitz" or convolution matrices and for which the size and rank are arbitrary.

Subsection 5.3.1 begins this section with decomposition of the channel input according to a combination of a channel-dependent deterministic vector space and a random (Hilbert) space corresponding to the transmitted signal's autocorrelation matrix. The decomposition extracts the essence of the input for which a finite-length canonical replacement channel can be found that is nonsingular and has the same mutual information as the original channel in Subsection 5.3.2. Having established an equivalent canonical channel, the GDFE then easily follows through canonical/Cholesky factorization of the input essence in Subsection 5.3.3. Subsection 5.3.4 investigates the proper input construction to get maximum GDFE performance, while Subsection 5.3.5 updates the precoders of Section 3.8 for the GDFE. Subsections 5.3.6 and 5.3.7 provide some interesting one-sided results for special channel structures that can often be imposed in practice.

5.3.1 Decomposition of the packet channel input

Figure 5.18 illustrates the division of the $(N+\nu)$-dimensional input space of vectors[9] into two mutually orthogonal subspaces: a **pass space** $\mathcal{P}$ containing input vectors that will pass through the channel so that $P\boldsymbol{x} \neq 0$ if $\boldsymbol{x} \in \mathcal{P}$ and a **null space** $\mathcal{N}$ containing input vectors that will be nulled by the channel so that $P\boldsymbol{x} = 0$ if $\boldsymbol{x} \in \mathcal{N}$. The component of any vector $\tilde{\boldsymbol{x}} \in \mathcal{P}$ in the null space is zero. 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 will appear at the channel output. Thus, only components in the pass space will carry information through the channel. A vector that has no component in the null space will be denoted by

[9]There are $2(N+\nu)$ dimensions when the input vector is complex and we thus mean $(N+\nu)$ complex vectors in the general case.

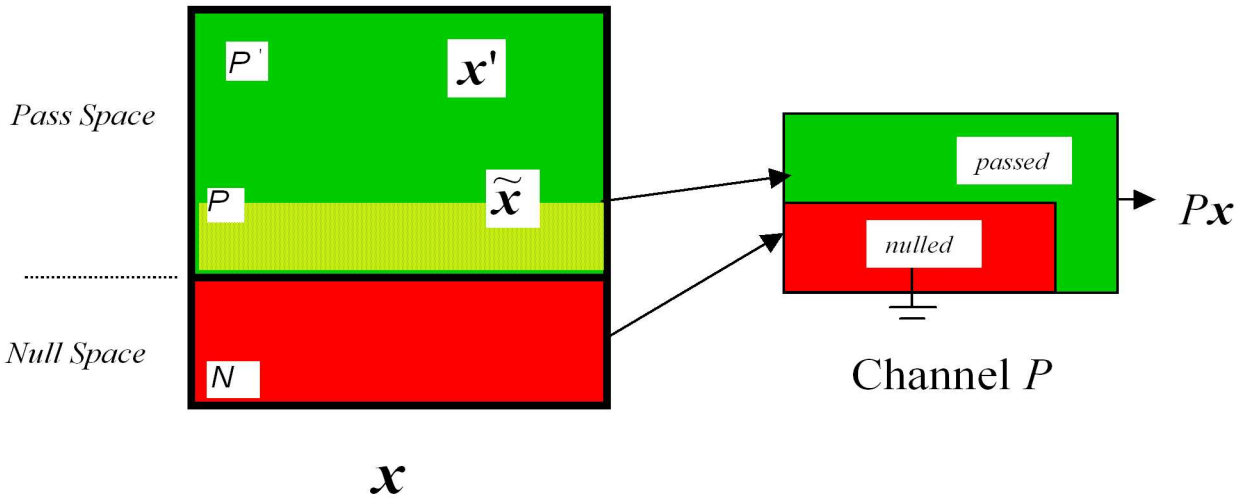


Figure 5.18: Decomposition of $\boldsymbol{x}$ into passed and nulled components.

$\tilde{\boldsymbol{x}}$.

A transmit vector can be written as

$$\boldsymbol{x} = \sum_{n=1}^{N+\nu} u_n \cdot \boldsymbol{c}_n = C\boldsymbol{u} \quad , \tag{5.92}$$

a generalization of modulation, where the vectors $\boldsymbol{c}_n$ $n = 1, ..., N+\nu$ need only be linearly independent (not necessarily an orthonormal set), and where the random-message components multiplying these vectors, u_n, need not be independent.[10] The set of vectors $\{\boldsymbol{c}_n\}_{n=1...N+\nu}$ may be projected into the rank-$\tilde{N}$ pass space of the channel to obtain a new set of $\tilde{N} \leq N$ linearly independent vectors $\{\tilde{\boldsymbol{c}}_n\}_{n=1...\tilde{N}}$, that characterize $\tilde{\boldsymbol{x}}$ by

$$\tilde{\boldsymbol{x}} = \sum_{n=1}^{\tilde{N}} \tilde{u}_n \tilde{\boldsymbol{c}}_n = \tilde{C} \cdot \tilde{\boldsymbol{u}} \quad . \tag{5.93}$$

The values for $\tilde{\boldsymbol{x}}$ span only an $\tilde{N}$ subspace of the $(N+\nu)$-dimensional complex vectors and are then, by construction, in one-to-one correspondence with values of the vector $\tilde{\boldsymbol{u}}$ (because of the linear independence of the vectors $\{\tilde{\boldsymbol{c}}_n\}_{N=1,...,\tilde{n}}$). A method for computing both $\tilde{\boldsymbol{u}}$ and $\tilde{C}$ from the channel matrix P and an arbitrary $\boldsymbol{u}$ and C appear in Example 5.3.1 below. The vector $\tilde{\boldsymbol{x}}$ is the finite-length discrete-time equivalent of an input signal $x(t)$ having no component in the dead-zones of the channel, i.e., those frequency ranges for which $H(f) = 0$.

In general, the input may have had components in the null space and so

$$R_{\boldsymbol{x}\boldsymbol{x}} \neq R_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}} \quad , \tag{5.94}$$

but any components of the message sequence not included in $\tilde{\boldsymbol{x}}$ are not of consequence to the transmission problem. The input has entropy

$$H_{\boldsymbol{x}} = H_{\boldsymbol{x}\in\mathcal{N}} + H_{\tilde{\boldsymbol{x}}} \geq H_{\tilde{\boldsymbol{x}}} \quad . \tag{5.95}$$

(The above equation is easily proved by thinking of entropy as the sum of scalar terms based on the eigenvalues of $R_{\boldsymbol{x}\boldsymbol{x}}$, which may be nonzero in the null-space of the channel.) However, basic MMSE

[10] A special case is when the input components $u_n = X_n$ are uncorrelated and the vectors $\boldsymbol{c}_n = \boldsymbol{m}_n$ are the channel-dependent, SVD-derived orthonormal basis of vector coding, but this need not be the case in general.

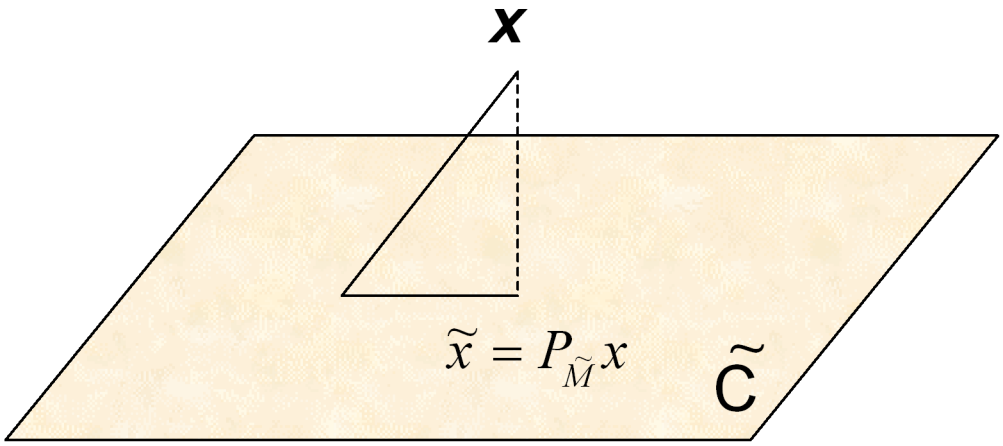


Figure 5.19: Projection illustration using singular vectors.

estimation relationships with entropy (Gaussian case) then also lead to[11]

$$H_{\boldsymbol{x}/\boldsymbol{y}} = H_{\boldsymbol{x}\in\mathcal{N}} + H_{\tilde{\boldsymbol{x}}/\boldsymbol{y}} \geq H_{\tilde{\boldsymbol{x}}/\boldsymbol{y}} \quad , \tag{5.96}$$

and thus

$$\begin{aligned} I(\boldsymbol{x};\boldsymbol{y}) &= H_{\boldsymbol{x}} - H_{\boldsymbol{x}/\boldsymbol{y}} && (5.97)\\ &= H_{\boldsymbol{x}\in\mathcal{N}} + H_{\tilde{\boldsymbol{x}}} - H_{\boldsymbol{x}\in\mathcal{N}} - H_{\tilde{\boldsymbol{x}}\boldsymbol{y}} && (5.98)\\ &= H_{\tilde{\boldsymbol{x}}} - H_{\tilde{\boldsymbol{x}}/\boldsymbol{y}} && (5.99)\\ &= I(\tilde{\boldsymbol{x}};\boldsymbol{y}) && (5.100) \end{aligned}$$

so that mutual information remains the same, an intuitively gratifying result – coding into the null space of the channel does not increase the achievable data rate. Example 5.3.1 illustrates one method for determining the null space and pass space of a channel and then finding the information-equivalent vector $\tilde{\boldsymbol{x}}$. Further, since $\tilde{\boldsymbol{u}}$ is in one-to-one correspondence with $\tilde{\boldsymbol{x}}$, then

$$I(\tilde{\boldsymbol{u}},\boldsymbol{y}) = I(\tilde{\boldsymbol{x}};\boldsymbol{y}) = I(\boldsymbol{x};\boldsymbol{y}) \quad . \tag{5.101}$$

Construction of $\tilde{C}$ via channel singular vectors

Figure 5.20 illustrates an algorithm developed in this section for elimination of components in $\boldsymbol{x}$ that are in the null space. The reader can refer to Figure 5.20 in reading the following.

The channel matrix P has a singular value decomposition $P = F\Lambda M^*$, which can be written

$$P = \sum_{n=1}^{\tilde{N}} \lambda_n \cdot \boldsymbol{f}_n \boldsymbol{m}_n^* \quad \text{(step 1 in Fig. 5.20)}, \tag{5.102}$$

where $\tilde{N} \leq N$ is the rank of the channel and corresponds to the number of nonzero singular values. The singular values are presumed here to be positive real and arranged in decreasing order $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_{\tilde{N}} \geq \lambda_{\tilde{N}+1} = 0$ without loss of generality. The pass space $\mathcal{P}$ is then spanned by the orthonormal set of vectors in the columns of the $(N+\nu) \times \tilde{N}$ matrix

$$\tilde{M} = [\boldsymbol{m}_1\ \boldsymbol{m}_2\ ...\ \boldsymbol{m}_{\tilde{N}}] \quad \text{(step 1 in Fig. 5.20)}. \tag{5.103}$$

[11] Recall the null space does not pass to $\boldsymbol{y}$ so $H_{\boldsymbol{x}\in\mathcal{N}/\boldsymbol{y}} = H_{\boldsymbol{x}\in\mathcal{N}}$.

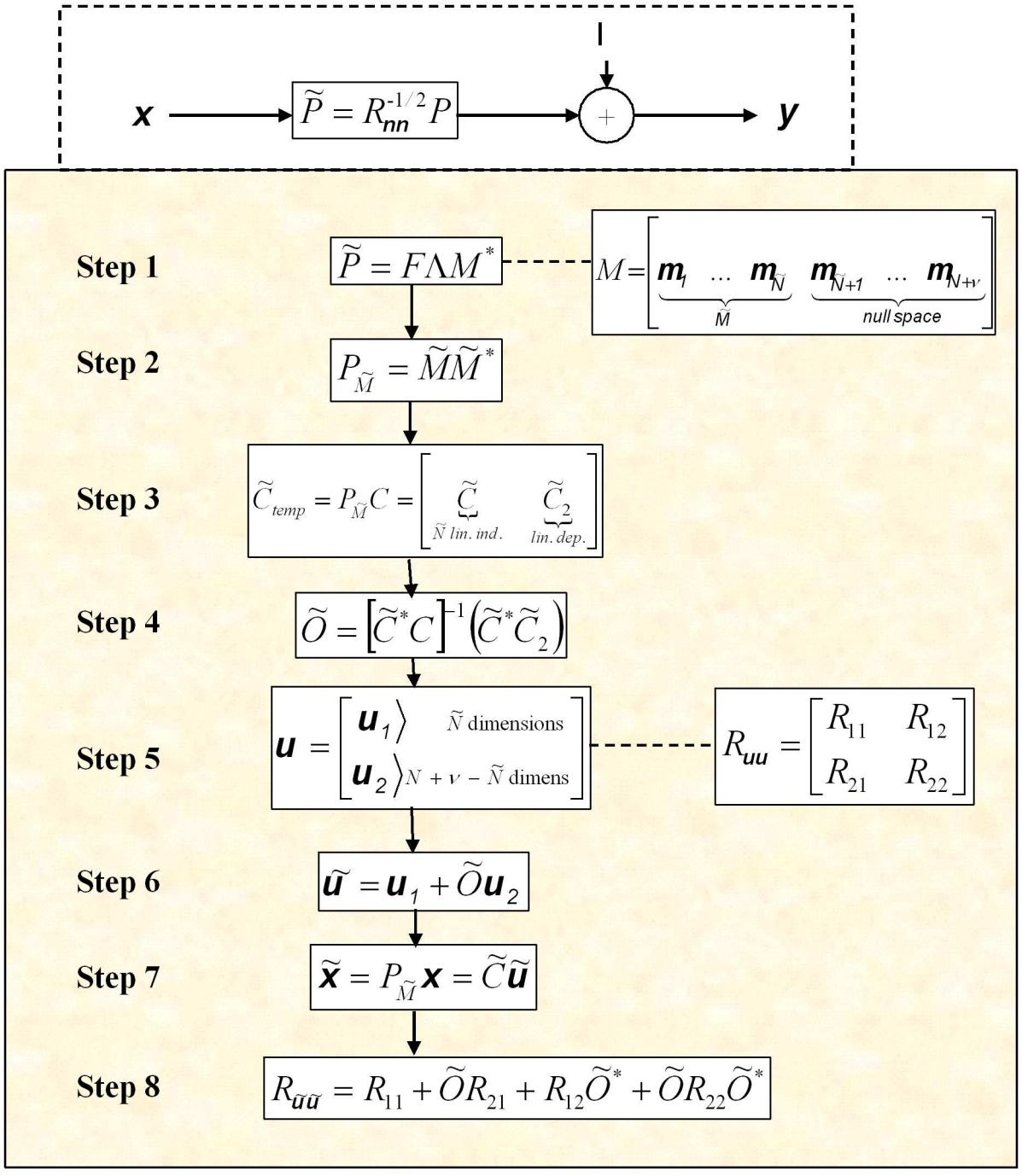


Figure 5.20: Algorithm for elimination of $\boldsymbol{x}$ components in channel null space.

Then,

$$\tilde{\boldsymbol{x}} = P_{\tilde{M}}\boldsymbol{x} \tag{5.104}$$

where

$$P_{\tilde{M}} = \tilde{M}\left(\tilde{M}^*\tilde{M}\right)^{-1}\tilde{M}^* = \tilde{M}\tilde{M}^* \quad \text{(step 2 in Fig. 5.20).} \tag{5.105}$$

The null space $\mathcal{N}$ is similarly spanned by the remaining columns of M. The projection in the pass space of each of the vectors $\tilde{\boldsymbol{c}}_n$ may be determined by the projection operation

$$\tilde{\boldsymbol{c}}_n = P_{\tilde{M}}\boldsymbol{c}_n \quad , \tag{5.106}$$

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

$$\tilde{C}_{temp} = P_{\tilde{M}}C \quad \text{(step 3 in Fig. 5.20),} \tag{5.107}$$

where the Hermitian matrix $P_{\tilde{M}}$ is known as a "projection matrix" in mathematics. Also, one may write

$$R_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}} = P_{\tilde{M}}R_{\boldsymbol{x}\boldsymbol{x}}P_{\tilde{M}} \quad . \tag{5.108}$$

The matrix C is a matrix of no more than $N+\nu$, $(N+\nu)$-dimensional column vectors that correspond to the original modulator for $\boldsymbol{x}$, and the matrix $\tilde{C}_{temp}$ is a rank-$\tilde{N}$ matrix of $(N+\nu)$-dimensional column vectors. This $\tilde{C}_{temp}$ matrix thus contains $\tilde{N}$ linearly independent columns, which can be reindexed (the designer must keep track of the indexing in practice) so that

$$\tilde{C}_{temp} = [\tilde{C} \mid \tilde{C}_2] \quad \text{(step 3 in Fig. 5.20),} \tag{5.109}$$

where the $(N+\nu)\times\tilde{N}$ matrix $\tilde{C}$ corresponds to the linearly independent vectors/columns of $\tilde{C}_{temp}$, and the remaining dependent vectors[12] are in $\tilde{C}_2$. Then, similarly reindexing the $\boldsymbol{u}$ components into the $\tilde{N}\times 1$ vector $\boldsymbol{u}_1$ and the remaining components in vector $\boldsymbol{u}_2$,

$$\begin{aligned}
\tilde{\boldsymbol{x}} &= P_{\tilde{M}}\boldsymbol{x} && (5.110)\\
&= P_{\tilde{M}}C\boldsymbol{u} && (5.111)\\
&= \tilde{C}_{temp}\boldsymbol{u} && (5.112)\\
&= [\tilde{C} \mid \tilde{C}_2]\begin{bmatrix}\boldsymbol{u}_1\\ \boldsymbol{u}_2\end{bmatrix} && (5.113)\\
&= \tilde{C}\boldsymbol{u}_1 + \tilde{C}_2\boldsymbol{u}_2 \quad . && (5.114)
\end{aligned}$$

The input autocorrelation matrix then can be partitioned according to the same labeling

$$R_{\boldsymbol{u}\boldsymbol{u}} = E\left\{\begin{bmatrix}\boldsymbol{u}_1\\ \boldsymbol{u}_2\end{bmatrix}[\boldsymbol{u}_1^* \;\; \boldsymbol{u}_2^*]\right\} = \begin{bmatrix}R_{11} & R_{12}\\ R_{21} & R_{22}\end{bmatrix} \quad , \tag{5.115}$$

The vectors/columns of the remaining matrix $\tilde{C}_2$ are linearly dependent on $\tilde{C}$ and can be represented in terms of their projections on $\tilde{C}$ by the matrix operation

$$\tilde{C}_2 = \tilde{C}\left[\tilde{C}^*\tilde{C}\right]^{-1}\tilde{C}^*\tilde{C}_2 \quad , \tag{5.116}$$

which defines a mtrix $\tilde{O}$ by

$$\tilde{C}_2 = \tilde{C}\left\{\left[\tilde{C}^*\tilde{C}\right]^{-1}\tilde{C}^*\tilde{C}_2\right\} = \tilde{C}\cdot\tilde{O} \quad \text{(step 4 in Fig. 5.20),} \tag{5.117}$$

[12]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($\tilde{P}_M$ * C)," then $\tilde{N}$ is obtained by the command "rank($\tilde{P}_M$* C)," and then $\tilde{C}$ is the first $\tilde{N}$ colums of the matrix formed by Q*R*J in matlab and the last $N-\tilde{N}$ columns are then $\tilde{C}_2$. The rearrangement of inputs by J needs to be noted and used by the designer also.

essentially a matrix post multiply by $\tilde{O}$ to get $\tilde{C}_2$ from $\tilde{C}$ in Equation 5.109. Then, Step 5 in Figure 5.20 shows explicitly the decomposition of $\boldsymbol{u}$ into the $\tilde{N}$ components $\boldsymbol{u}_1$ that multiply $\tilde{C}$ and the $N-\tilde{N}$ components $\boldsymbol{u}_2$ that multiply $\tilde{C}_2$. These two components can be combined into a single component $\tilde{\boldsymbol{u}}$ according to

$$\tilde{\boldsymbol{x}} = \tilde{C}\cdot\boldsymbol{u}_1 + \tilde{C}\cdot\tilde{O}\cdot\boldsymbol{u}_2 = \tilde{C}\underbrace{\left(\boldsymbol{u}_1 + \tilde{O}\cdot\boldsymbol{u}_2\right)}_{\text{Step 6}} = \underbrace{\tilde{C}\cdot\tilde{\boldsymbol{u}}}_{\text{Step 7}} \quad . \tag{5.118}$$

So, $\tilde{\boldsymbol{u}}$ is defined by

$$\tilde{\boldsymbol{u}} \triangleq \boldsymbol{u}_1 + \tilde{O}\boldsymbol{u}_2 \quad . \tag{5.119}$$

Then, since $\tilde{C}_2$ has been eliminated, the designer simply discards $\tilde{C}_{temp}$ and uses instead $\tilde{C}$. The new input autocorrelation matrix is

$$R_{\tilde{\boldsymbol{u}}\tilde{\boldsymbol{u}}} = R_{11} + \tilde{O}R_{21} + R_{12}\tilde{O}^* + \tilde{O}R_{22}\tilde{O}^* \quad \text{(Step 8 in Fig. 5.20).} \tag{5.120}$$

The command "orth" in matlab essentially performs the reduction of the set of $\tilde{\boldsymbol{c}}$ to a linearly independent set of minimal rank, while the more general command "qr" in matlab does QR-factorization, which also provides both the new set of vectors and the linear combinations necessary to compute $\tilde{\boldsymbol{u}}$ with less computation than Equation (5.117). This "qr" matlab command can also optionally reorder the indices of the vectors to ensure nonsingularity in the first $\tilde{N}$ positions.

Figure 5.21 illustrates the original channel (at the top) and the equivalent channel with singularity removed. 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 channel singularity.

EXAMPLE 5.3.1 ($1+D$ Channel) An example 2×3 matrix

$$P = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \tag{5.121}$$

corresponds to the $1+D$ channel with $N=2$ and $\nu=1$. Singular value decomposition of this channel produces

$$P = F\begin{bmatrix} \sqrt{3} & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}\begin{bmatrix} -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} \\ -\frac{\sqrt{2}}{\sqrt{3}} & 0 & -\frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} \end{bmatrix}^* \tag{5.122}$$

and

$$\tilde{M} = \begin{bmatrix} \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{2}} \\ \frac{\sqrt{2}}{\sqrt{3}} & 0 \\ \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{2}} \end{bmatrix} \quad . \tag{5.123}$$

Then

$$P_{\tilde{M}} = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} & -\frac{1}{3} \\ \frac{1}{3} & \frac{2}{3} & \frac{1}{3} \\ -\frac{1}{3} & \frac{1}{3} & \frac{2}{3} \end{bmatrix} \tag{5.124}$$

Then with a white PAM input then, $\boldsymbol{x}=\boldsymbol{u}$ or equivalently $C=I$ so that

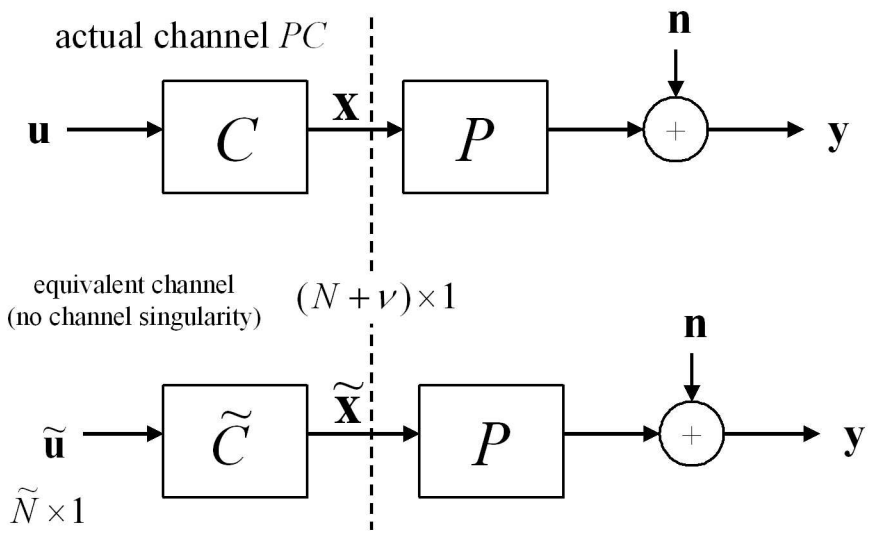
$$\tilde{\boldsymbol{x}} = P_{\tilde{M}}\boldsymbol{x} \tag{5.125}$$

and then

$$P_{\tilde{M}} = P_{\tilde{M}}C = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} & -\frac{1}{3} \\ \frac{1}{3} & \frac{2}{3} & \frac{1}{3} \\ -\frac{1}{3} & \frac{1}{3} & \frac{2}{3} \end{bmatrix} \tag{5.126}$$

where

$$\tilde{C} = \begin{bmatrix} \frac{2}{3} & \frac{1}{3} \\ \frac{1}{3} & \frac{2}{3} \\ -\frac{1}{3} & \frac{1}{3} \end{bmatrix} \quad \text{and} \quad \tilde{C}_2 = \begin{bmatrix} -\frac{1}{3} \\ \frac{1}{3} \\ \frac{2}{3} \end{bmatrix} \quad . \tag{5.127}$$



$$P = F\Lambda M^* \rightarrow P_{\widetilde{M}} = \widetilde{M}\widetilde{M}^* \rightarrow \widetilde{C}_{temp} = P_{\widetilde{M}} \cdot C = \left[\underbrace{\widetilde{C}}_{\substack{\text{linearly} \\ \text{independent}}} \quad \underbrace{\widetilde{C}_2}_{\substack{\text{dependent} \\ \text{on } \widetilde{C}}} \right] \rightarrow \mathbf{u} = \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix}$$

$$\widetilde{O} = \left(\widetilde{C}^* C\right)^{-1} \widetilde{C}^* \widetilde{C}_2 \quad \rightarrow \quad \widetilde{\mathbf{u}} = \mathbf{u}_1 + \widetilde{O} \cdot \mathbf{u}_2 \qquad \boxed{I(\mathbf{x};\mathbf{y}) = I\left(\widetilde{\mathbf{u}};\mathbf{y}\right)}$$

Figure 5.21: The original channel and its equivalent with singularity removed.

The matrix $\tilde{O}$ is then

$$\tilde{O} = \begin{bmatrix} -1 \\ 1 \end{bmatrix} \quad . \tag{5.128}$$

The new input $\tilde{\boldsymbol{u}}$ is then

$$\tilde{\boldsymbol{u}} = \boldsymbol{u}_1 + \tilde{O}\boldsymbol{u}_2 = \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} + \begin{bmatrix} -1 \\ 1 \end{bmatrix} u_3 = \begin{bmatrix} u_1 - u_3 \\ u_2 + u_3 \end{bmatrix} \quad . \tag{5.129}$$

The channel output

$$\begin{aligned} \boldsymbol{y} &= PP_{\tilde{M}}C\boldsymbol{u} + \boldsymbol{n} = P\tilde{\boldsymbol{x}} + \boldsymbol{n} && (5.130)\\ &= P\tilde{C}\tilde{\boldsymbol{u}} + \boldsymbol{n} && (5.131)\\ &= \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \tilde{u}_1 \\ \tilde{u}_2 \end{bmatrix} + \boldsymbol{n} && (5.132) \end{aligned}$$

and thus the final model is in terms of just those components of $\boldsymbol{x}$, $\tilde{u}_1 = u_1 - u_3 = x_1 - x_3$ and $\tilde{u}_2 = u_2 + u_3 = x_2 + x_3$, that "pass" through P. Furthermore, it is clear that any component of the input that lays on the vector

$$\boldsymbol{m}_3 = \begin{bmatrix} \frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \end{bmatrix} \tag{5.133}$$

for instance the PAM input

$$\begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} \tag{5.134}$$

will not pass. 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 are simply wasting energy as far as passage through the channel is concerned.

For this channel, better input design than PAM would let only $\mathcal{E}_{\tilde{u}_1}$ and $\mathcal{E}_{\tilde{u}_2}$ be nonzero while avoiding any oscillating components (i.e, +,-,+ or -,+,-).

The reduction of modulator vectors to the pass space so far has been based on the fixed deterministic vectors in C and on the deterministic channel, in particular through $\tilde{M}$. The pass space and the null space depend only upon the channel matrix P. However, the input vectors are random, constructed through modulation of a random component u_n on each of the fixed deterministic transmit modulator vectors $u_n \cdot \boldsymbol{c}_n$. Thus, the reduction so far to $\tilde{\boldsymbol{x}}$ depends only on the fixed part of the input, namely $\tilde{\boldsymbol{c}}$. The projected vectors $\tilde{\boldsymbol{x}}$ themselves span a random vector space that is determined (Gaussian case) by $R_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}}$. This random vector space rarely occurs by accident, or usually occurs by design when $R_{\boldsymbol{x}\boldsymbol{x}}$ (or $R_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}}$) is optimized, may have lesser rank or dimensionality than $\tilde{N}$. The discrete equivalent of the PW criterion for realization of channel inputs is that $R_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}}$ and also $R_{\boldsymbol{x}\boldsymbol{x}}$ be nonsingular, which can be written in the form $\ln|R_{\boldsymbol{x}\boldsymbol{x}}| < \infty$, remniscent of the integral form of the continuous PW criterion. With $\nu > 0$, the PW is never satisfied for white inputs like PAM – that is, PAM is never a good design on a channel with ISI and finite-length packets.

Elimination of Input Singularity

A second step in singularity elimination is that of any singularities in the input autocorrelation. These singularities can be unrelated to the channel null space, and may be introduced intentionally by design or by accident by the uniformed designer. Figure 5.22 illustrates the process that is subsequently described in this subsection.

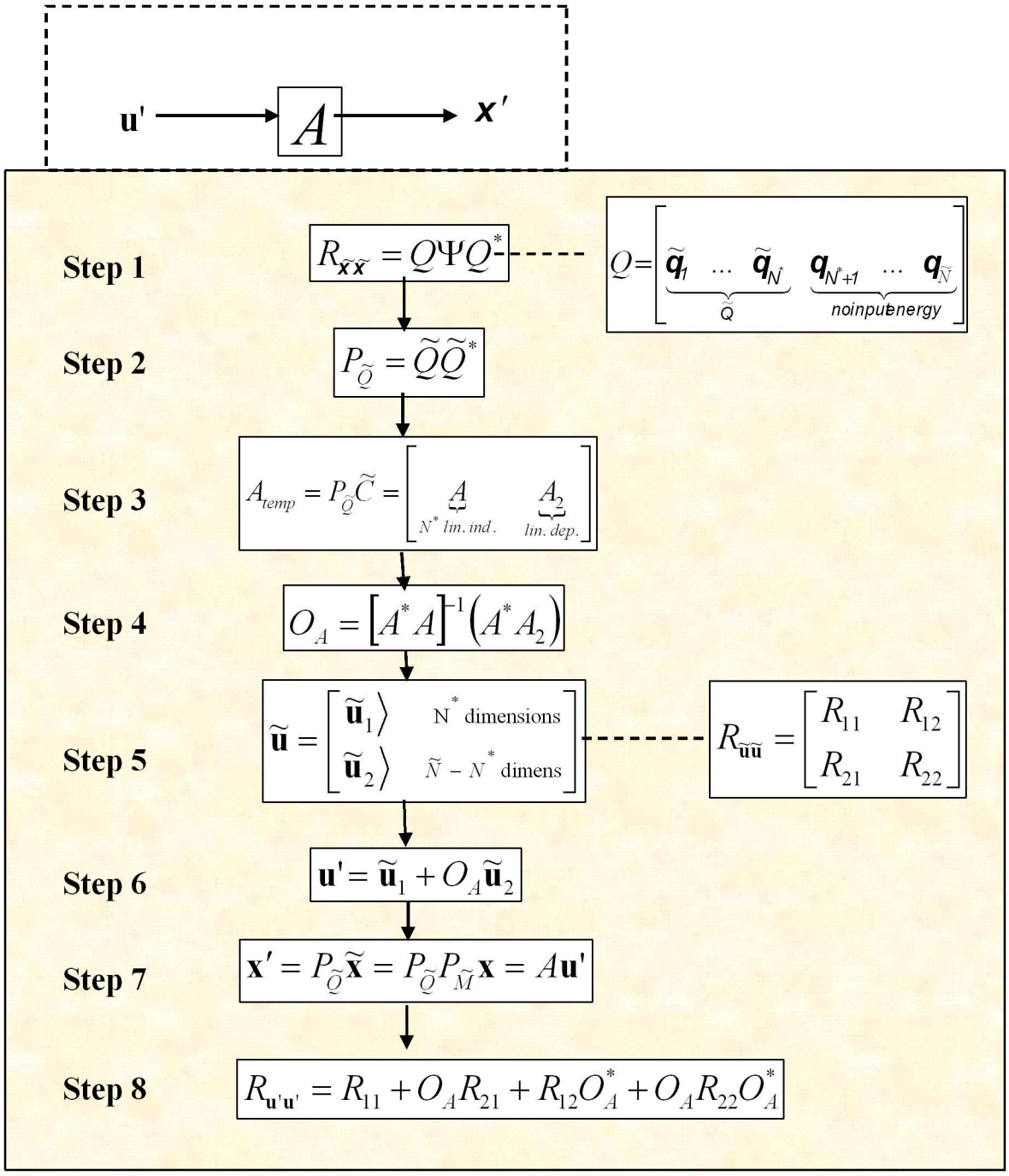


Figure 5.22: Algorithm for elimination of $\boldsymbol{x}$ components in the input autocorrelation $R_{\boldsymbol{xx}}$.

The input covariance matrix $R_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}}$ has a **eigendecomposition** (step 1 in Fig 5.22)

$$R_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}} = \sum_{n=1}^{N^*} \tilde{\mathcal{E}}_{x,n} \cdot \tilde{\boldsymbol{q}}_n \tilde{\boldsymbol{q}}_n^* = [\tilde{\boldsymbol{q}}_1\ \tilde{\boldsymbol{q}}_2\ ...\ \tilde{\boldsymbol{q}}_{\tilde{N}}] \begin{bmatrix} \tilde{\mathcal{E}}_{x,1} & ... & 0 & 0 & ... & 0 \\ 0 & \ddots & 0 & 0 & \vdots & 0 \\ 0 & ... & \tilde{\mathcal{E}}_{x,N^*} & 0 & ... & 0 \\ 0 & ... & 0 & \ddots & ... & 0 \\ 0 & ... & 0 & ... & ... & 0 \end{bmatrix} \begin{bmatrix} \tilde{\boldsymbol{q}}_1^* \\ \vdots \\ \tilde{\boldsymbol{q}}_{\tilde{N}} \end{bmatrix} \quad , \tag{5.135}$$

where $\tilde{\mathcal{E}}_{x,n}$ is the n^{th} eigenvalue, and $\tilde{\boldsymbol{q}}_n$ is the corresponding eigenvector. Those input vectors for which $\tilde{\mathcal{E}}_{x,n} = 0$ can be omitted from the construction of $\tilde{\boldsymbol{x}}$ (or $\boldsymbol{x}$) because $\tilde{\boldsymbol{x}}$ has no information-bearing component on these dimensions. Define $\tilde{Q} \triangleq [\tilde{\boldsymbol{q}}_1 ... \tilde{\boldsymbol{q}}_{N^*}]$ and $P_{\tilde{Q}} \triangleq \tilde{Q}\tilde{Q}^*$ (step 2 in Fig. 5.22).

Figure 5.18 further suggests a second projection, now of $\tilde{\boldsymbol{x}}$ to $\boldsymbol{x}'$, into a subset of the pass space $\mathcal{P}' \subseteq \mathcal{P}$ that depends only on the nonzero components/eigenmodes of the input autocorrelation matrix:

$$\begin{aligned} \boldsymbol{x}' &= P_{\tilde{Q}}\tilde{\boldsymbol{x}} && (5.136) \\ &= P_{\tilde{Q}}\tilde{C}\tilde{\boldsymbol{u}} && (5.137) \\ &= A\boldsymbol{u}' \quad , && (5.138) \end{aligned}$$

where A and $\boldsymbol{u}'$ remain to be determined:

Without loss of generality, one can assume that the indexing of the eigenvalues is such that $\mathcal{E}_{x,n} = 0$ for $n > N^*$. This projection is achieved by creating a set of $N^* \leq \tilde{N}$ new vectors as the columns of the $(N+\nu) \times N^*$ matrix A_{temp} by first computing (with A a set of N^* linearly independent columns of A_{temp})

$$A_{temp} = P_{\tilde{Q}}\tilde{C} = [A \mid A_2] \quad \text{(step 3 in Fig. 5.22)}, \tag{5.139}$$

and note[13]

$$A_2 = AO_A \tag{5.140}$$

where $O_A = (A^*A)^{-1}A^*A_2$ (step 4 in Fig. 5.22). Then

$$\tilde{\boldsymbol{u}} = \begin{bmatrix} \tilde{\boldsymbol{u}}_1 \\ \tilde{\boldsymbol{u}}_2 \end{bmatrix} \quad \text{(step 5 in Fig. 5.22)}, \tag{5.141}$$

and

$$\boldsymbol{u}' = \tilde{\boldsymbol{u}}_1 + O_A\tilde{\boldsymbol{u}}_2 \quad \text{(step 6 in Fig. 5.22)}. \tag{5.142}$$

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

The entropy of $\boldsymbol{x}'$ remains the same as $\tilde{\boldsymbol{x}}$, $H_{\boldsymbol{x}'} = H_{\tilde{\boldsymbol{x}}}$, because removing zeroed eigenvalues from the Gaussian autocorrelation matrix does not change the entropy nor the matrix itself. Thus, the mutual information remains identical to that of the original channel between the essence of the input $\boldsymbol{x}'$ and the channel output $\boldsymbol{y}$,

$$I(\boldsymbol{x}';\boldsymbol{y}) = I(\tilde{\boldsymbol{x}};\boldsymbol{y}) = I(\boldsymbol{x};\boldsymbol{y}) \quad . \tag{5.143}$$

The final desired decomposition of the input into its nonsingular essence is then

$$\boldsymbol{x}' = P_{\tilde{Q}}P_M\boldsymbol{x} = A\boldsymbol{u}' \quad \text{(step 7 in Fig. 5.22)}. \tag{5.144}$$

The new input autocorrelation matrix satisfies

$$R_{\boldsymbol{x}'\boldsymbol{x}'} = AR_{\boldsymbol{u}'\boldsymbol{u}'}A^* = A\left(R_{11} + O_AR_{21} + R_{12}O_A^* + O_AR_{22}O_A^*\right)A^* \quad \text{(step 8 in Fig. 5.22)}, \tag{5.145}$$

[13]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($\tilde{P}_{\tilde{Q}}$ * $\tilde{C}$)," then N^* is obtained by the command "rank($\tilde{P}_{\tilde{Q}}$ * $\tilde{C}$)," and then A is the first N^* colums of the matrix formed by Q*R*J in matlab and the last $\tilde{N} - N^*$ columns are then A_2. he rearrangement of inputs by J needs to be noted and used by the designer also.

where $R_{\boldsymbol{u}'\boldsymbol{u}'}$ is necessarily nonsingular and of rank N^* and of course the matrix entries R_{11}, R_{12}, R_{21} and R_{22} were obtained from the original $\tilde{\boldsymbol{u}}$, and hence then also from $\boldsymbol{u}$, from which the essence of the input was extracted (and then redefined after definition of these submatrices). For every value that the random vector $\boldsymbol{x}'$ is allowed to take with nonzero probability, there is a unique value of $\boldsymbol{u}'$ that corresponds to it, so these two are in one-to-one correspondence, and thus

$$I(\boldsymbol{u}';\boldsymbol{y}) = I(\boldsymbol{x}';\boldsymbol{y}) = I(\boldsymbol{x};\boldsymbol{y}) \quad . \tag{5.146}$$

Essentially, $\boldsymbol{u}'$ generates $\boldsymbol{x}'$ for the transmission problem of interest.

The decomposition of the input to its essence, $\boldsymbol{u}$, is the finite-length equivalent to finding only that transmit band over which there is nonzero energy (or which satisfies the PW criterion) and 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, a nonsingular equivalent system has been derived.

EXAMPLE 5.3.2 (reducing a singular input for the $1+D$ packet channel) This example returns to the $1+D$ packet channel, except with the input autocorrelation matrix

$$R_{\boldsymbol{x}\boldsymbol{x}} = \begin{bmatrix} 2/3 & 1/3 & 2/3 \\ 1/3 & 5/3 & 1/3 \\ 2/3 & 1/3 & 2/3 \end{bmatrix} \quad , \tag{5.147}$$

which is presumably generated from some $\boldsymbol{u}$ according to $\boldsymbol{x} = C\boldsymbol{u}$, where the input $\boldsymbol{u}$ has 3×3 autocorrelation matrix $R_{\boldsymbol{u}\boldsymbol{u}} = I$. The following sequence of matlab commands eliminates the energy on the channel null space

```
>> rxx
rxx = 0.6667    0.3333    0.6667
      0.3333    1.6667    0.3333
      0.6667    0.3333    0.6667
>> [F,L,M]=svd([1 1 0
0 1 1])
F =-0.7071   -0.7071    L= 1.7321       0      0    M = -0.4082   -0.7071    0.5774
   -0.7071    0.7071            0  1.0000      0        -0.8165    0.0000   -0.5774
                                                      -0.4082    0.7071    0.5774
>> Mt=M(1:3,1:2)
Mt = -0.4082   -0.7071
     -0.8165    0.0000
     -0.4082    0.7071
>> Pm=Mt*Mt'
Pm = 0.6667    0.3333   -0.3333
     0.3333    0.6667    0.3333
    -0.3333    0.3333    0.6667
>> rxxtt = Pm*rxx*Pm
rxxtt = 0.3333    0.6667    0.3333
        0.6667    1.3333    0.6667
        0.3333    0.6667    0.3333
>> Ctemp=Pm*eye(3)
Ctemp = 0.6667    0.3333   -0.3333
        0.3333    0.6667    0.3333
       -0.3333    0.3333    0.6667
>> Ct=Ctemp(1:3,1:2)
Ct  = 0.6667    0.3333
      0.3333    0.6667
     -0.3333    0.3333
```

```
>> Ot=inv(Ct'*Ct)*Ct'*Ctemp(1:3,3)
Ot = -1.0000
      1.0000
```

This means that that $\tilde{\boldsymbol{u}}$ is

$$\tilde{\boldsymbol{u}} = \begin{bmatrix} u_1 - u_3 \\ u_2 + u_3 \end{bmatrix} \tag{5.148}$$

which can be viewed as an input and renamed to $\boldsymbol{u}$ if desired. The input autocorrelation matrix $R_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}}$ is singular and $N^* = 1$ in this case, so additional reduction is necessary, as achieved by the additional matlab commands:

```
rxxtt = 0.3333    0.6667    0.3333
        0.6667    1.3333    0.6667
        0.3333    0.6667    0.3333
>> [v,d]=eig(rxxtt)
v = -0.0918   -0.9082    0.4082       d=  0         0         0
    -0.4082    0.4082    0.8165           0         0         0
     0.9082    0.0918    0.4082           0         0         2
>> q=v(:,3)
q = 0.4082
    0.8165
    0.4082
>> 2*q*q'   (check on autocorrelation matrix)
ans =
    0.3333    0.6667    0.3333
    0.6667    1.3333    0.6667
    0.3333    0.6667    0.3333
>> Atemp =.5*rxxtt*Ct1
Atemp = 0.1667    0.3333
        0.3333    0.6667
        0.1667    0.3333

>> A=Atemp(:,1)
A = 0.1667
     0.3333
     0.1667
>> Oa=inv(A'*A)*A'*Atemp(:,2)
Oa = 2.0000
```

Thus the final $\boldsymbol{u}'$ is a scalar and is equal to $u' = \tilde{u}_1 + 2\tilde{u}_2 = (u_1 - u_3) + 2(u_2 + u_3) = u_1 + 2u_2 + u_3)$ where u_1, u_2, and u_3 are independent and identically distributed unit-variance random processes. There is only one dimension of the input that makes it to the channel output

$$\boldsymbol{y} = PA\boldsymbol{u}' + \boldsymbol{n} = \begin{bmatrix} 1/6 \\ 1/3 \\ 1/6 \end{bmatrix} (u_1 + 2u_2 + u_3) + \boldsymbol{n} \quad . \tag{5.149}$$

The rest of the input components are lost and do not carry information to the channel output.

An alternative input that was white only over the pass space of this channel and zero elsewhere by design might be of interest. Such an input could have have autocorrelation matrix constructed by

$$R_{\boldsymbol{x}\boldsymbol{x}} = 1.5 \cdot \tilde{M} \cdot \tilde{M}^* = \begin{bmatrix} 2/3 & 1/3 & -1/3 \\ 1/3 & 2/3 & 1/3 \\ -1/3 & 1/3 & 2/3 \end{bmatrix} \quad , \tag{5.150}$$

which could correspond to an input constructed according to (the factor of 1.5 corresponds to increasing the total energy to 3 units for the 3 dimensions, which is the trace of the autocorrelation matrix)

$$\boldsymbol{x} = X_1\boldsymbol{m}_1 + X_2\boldsymbol{m}_2 = X_1 \cdot \begin{bmatrix} -\frac{1}{\sqrt{6}} \\ -\frac{\sqrt{2}}{\sqrt{3}} \\ -\frac{1}{\sqrt{6}} \end{bmatrix} + X_2 \cdot \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ 0 \\ -\frac{1}{\sqrt{2}} \end{bmatrix} \tag{5.151}$$

with X_1 and X_2 independent random variables each of variance 1.5. In this case, clearly $\tilde{\boldsymbol{x}} = \boldsymbol{x} = \boldsymbol{x}'$. The construction of an input autocorrelation matrix is not unique.

The construction of an input $\boldsymbol{x} = \sqrt{\frac{2}{3}}\,(X_1\boldsymbol{m}_1 + X_2\boldsymbol{m}_2 + X_3\boldsymbol{m}_3)$ would not be wise, but would have $R_{\boldsymbol{xx}} = I$, and would lose 1/3 the energy to the null space. Other constructions of such white inputs might not have independent components nor even orthogonal vectors. In all cases, the white input injects 1/3 of its power into the null space of the channel.

5.3.2 Finite-Length Canonical Channels

Given an input $\boldsymbol{u}'$ after the entire process of singularity elimination in Subsection 5.3.1, this subsection and all ensuing chapters and sections will rename this input simply to $\boldsymbol{u}$ to avoid the use of primes and tildes. The primes and tildes are necessary in singularity elimination for notational bookkeeping purposes, but need not be carried further. The input then to the non-singular channel thus necessarily constructed for the finite packet is heretofore $\boldsymbol{u}$.

The GDFE development generalizes to Gaussian noise vectors $\boldsymbol{n}$ that do not necessarily have a diagonal autocorrelation matrix

$$R_{\boldsymbol{nn}} = R_{\boldsymbol{nn}}^{1/2} \cdot R_{\boldsymbol{nn}}^{*/2} \quad , \tag{5.152}$$

where $R_{\boldsymbol{nn}}^{1/2}$ is any (non-unique) square root of $R_{\boldsymbol{nn}}$. The equivalent "white-noise" channel is formed by whitening the noise or

$$\tilde{\boldsymbol{y}} = R_{\boldsymbol{nn}}^{-1/2}\boldsymbol{y} = R_n^{-1/2}(P\tilde{A}\boldsymbol{u} + \boldsymbol{n}) = \tilde{P}\boldsymbol{u} + \tilde{\boldsymbol{n}} \quad , \tag{5.153}$$

where $\tilde{P} = R_{\boldsymbol{nn}}^{-1/2}PA$ and $\tilde{\boldsymbol{n}}$ is white noise with unit variance per complex dimension. This noise whitening was tacitly illustrated at the top of Figure 5.20 earlier, and can undergo singularity removal according to the same procedure. The filtered noise $\tilde{\boldsymbol{n}}$ remains independent of the input $\boldsymbol{u}$. By the sufficiency of matched filtering[14], the receiver may process the received signal by a set of matched filters (the rows of $\tilde{P}$) without loss of information to obtain the **canonical forward channel model**:

$$\boldsymbol{z} = \tilde{P}^*\tilde{P}\boldsymbol{u} + \tilde{P}^*\tilde{\boldsymbol{n}} = R_f\boldsymbol{u} + \boldsymbol{n}' \quad , \tag{5.154}$$

where R_f is the canonical forward-channel matrix $R_f = \tilde{P}^*\tilde{P}$ and $R_{\boldsymbol{n}'\boldsymbol{n}'} = R_f$. This forward channel is canonical because the channel shaping R_f is the same as the noise-autocorrelation or shaping, also R_f. Because the noise vector $\boldsymbol{n}$ is uncorrelated with the input vector $\boldsymbol{u}$, a right triangle can be drawn to illustrate Equation (5.154) as in Figure 5.23 (on the left). This triangle satisfies a "pythagorean theorem"

$$R_{\boldsymbol{zz}} = R_f R_{\boldsymbol{uu}} R_f + R_{\boldsymbol{n}'\boldsymbol{n}'} = R_f R_{\boldsymbol{uu}} R_f + R_f \quad , \tag{5.155}$$

observing that $R_f^* = R_f$. Also, because canonical channel noise $\boldsymbol{n}'$ is uncorrelated with the channel input $\boldsymbol{u}$, then

$$\hat{\boldsymbol{z}} = R_f\boldsymbol{u} \quad , \tag{5.156}$$

is the MMSE vector estimate of $\boldsymbol{z}$ given $\boldsymbol{u}$. Thus, another expression for R_f by the orthogonality principle is

$$R_f = R_{\boldsymbol{zu}}R_{\boldsymbol{uu}}^{-1} \quad . \tag{5.157}$$

[14]This matrix matched filter is a one-to-one mapping on the signal components of $\boldsymbol{x}'$ $\boldsymbol{u}$ and the noise after filtering is white and so any components on non-signal vectors are irrelevent.

The MMSE matrix for this forward-channel estimation problem is then clearly $R_{\boldsymbol{n}'\boldsymbol{n}'} = R_f$.

A corresponding MMSE estimate of $\boldsymbol{u}$ given $\boldsymbol{z}$ is found through the **canonical backward channel model:**

$$\boldsymbol{u} = R_b \boldsymbol{z} + \boldsymbol{e} \quad , \tag{5.158}$$

where R_b is the MMSE matrix estimator (equalizer)

$$R_b = R_{\boldsymbol{u}\boldsymbol{z}} R_{\boldsymbol{z}\boldsymbol{z}}^{-1} \quad . \tag{5.159}$$

The vector $\boldsymbol{e}$ is the MMSE error vector for the MMSE estimation of $\boldsymbol{u}$ given $\boldsymbol{z}$. This backward problem is a matrix linear equalizer. Cleary, the dual Pythagorean relationship shown on the right in Figure 5.23 is

$$R_{\boldsymbol{e}\boldsymbol{e}} = R_{\boldsymbol{u}\boldsymbol{u}} - R_{\boldsymbol{u}\boldsymbol{z}} R_{\boldsymbol{z}\boldsymbol{z}}^{-1} R_{\boldsymbol{z}\boldsymbol{u}} = R_{\boldsymbol{u}\boldsymbol{u}} - R_b R_{\boldsymbol{z}\boldsymbol{z}} R_b \quad . \tag{5.160}$$

A very useful alternatively expression for R_b is (recalling that $R_{\boldsymbol{u}\boldsymbol{z}}$ is square nonsingular because of the earlier singularity removal.

$$\begin{aligned}
R_b &= R_{\boldsymbol{u}\boldsymbol{z}} R_{\boldsymbol{z}\boldsymbol{z}}^{-1} && (5.161)\\
R_b^{-1} &= R_{\boldsymbol{z}\boldsymbol{z}} R_{\boldsymbol{z}\boldsymbol{u}}^{-*} && (5.162)\\
R_b^{-1} R_{\boldsymbol{u}\boldsymbol{u}} &= R_{\boldsymbol{z}\boldsymbol{z}} R_{\boldsymbol{z}\boldsymbol{u}}^{-*} R_{\boldsymbol{u}\boldsymbol{u}} && (5.163)\\
&= R_{\boldsymbol{z}\boldsymbol{z}} R_f^{-1} && (5.164)\\
&= \left(R_f R_{\boldsymbol{u}\boldsymbol{u}} R_f + R_f\right) R_f^{-1} && (5.165)\\
&= R_f R_{\boldsymbol{u}\boldsymbol{u}} + I && (5.166)\\
R_b^{-1} &= R_f + R_{\boldsymbol{u}\boldsymbol{u}}^{-1} && (5.167)
\end{aligned}$$

which allows computation of R_b from the easily computed $R_f = PP^*$ and $R_{\boldsymbol{u}\boldsymbol{u}}$ the given transmit autocorrelation matrix. Equivalently,

$$I = R_b R_f + R_b R_{\boldsymbol{u}\boldsymbol{u}}^{-1} = R_f R_b + R_{\boldsymbol{u}\boldsymbol{u}}^{-1} R_b \text{ since I is symmetric} \quad . \tag{5.168}$$

Using this convenient result (5.168) in Equation 5.160

$$\begin{aligned}
R_{\boldsymbol{e}\boldsymbol{e}} &= R_{\boldsymbol{u}\boldsymbol{u}} - R_{\boldsymbol{u}\boldsymbol{z}} R_{\boldsymbol{z}\boldsymbol{z}}^{1-} R_{\boldsymbol{z}\boldsymbol{u}} && (5.169)\\
&= R_{\boldsymbol{u}\boldsymbol{u}} \left(I - R_{\boldsymbol{u}\boldsymbol{u}}^{-1} R_{\boldsymbol{u}\boldsymbol{z}} R_{\boldsymbol{z}\boldsymbol{z}}^{-1} R_{\boldsymbol{z}\boldsymbol{u}}\right) && (5.170)\\
&= R_{\boldsymbol{u}\boldsymbol{u}} \left(I - R_b R_f\right) && (5.171)\\
&= R_{\boldsymbol{u}\boldsymbol{u}} \left(I - R_f R_b\right) && (5.172)\\
&= R_{\boldsymbol{u}\boldsymbol{u}} R_{\boldsymbol{u}\boldsymbol{u}}^{-1} R_b && (5.173)\\
&= R_b \quad , && (5.174)
\end{aligned}$$

so the backward channel is thus also canonical (the autocorrelation of the error vector equals the channel shaping $= R_b$). So just as $R_{\boldsymbol{n}\boldsymbol{n}} = R_f$, $R_{\boldsymbol{e}\boldsymbol{e}} = R_b$. 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 $R_{\boldsymbol{e}\boldsymbol{e}}$ by $R_{\boldsymbol{n}'\boldsymbol{n}'}$ (essentially then $\boldsymbol{z} \leftrightarrow \boldsymbol{u}$, $\boldsymbol{n}' \rightarrow \boldsymbol{e}$). The following relations use this duality of the forward and backward channels depicted in Figure 5.23 (see Homework 6, Prob 1.):

$$\begin{aligned}
R_b^{-1} &= R_f + R_{\boldsymbol{u}\boldsymbol{u}}^{-1} && (5.175)\\
R_f^{-1} &= R_b + R_{\boldsymbol{z}\boldsymbol{z}}^{-1} && (5.176)\\
R_{\boldsymbol{e}\boldsymbol{e}} &= R_b && (5.177)\\
R_{\boldsymbol{n}'\boldsymbol{n}'} &= R_f && (5.178)\\
R_{\boldsymbol{z}\boldsymbol{z}} &= R_f R_{\boldsymbol{u}\boldsymbol{u}} R_f + R_f = R_f R_{\boldsymbol{u}\boldsymbol{u}} R_b^{-1} && (5.179)\\
R_{\boldsymbol{u}\boldsymbol{u}} &= R_b R_{\boldsymbol{z}\boldsymbol{z}} R_b + R_b = R_b R_{\boldsymbol{z}\boldsymbol{z}} R_f^{-1} && (5.180)\\
R_{\boldsymbol{z}\boldsymbol{u}} &= R_f R_{\boldsymbol{u}\boldsymbol{u}} && (5.181)\\
R_{\boldsymbol{u}\boldsymbol{z}} &= R_b R_{\boldsymbol{z}\boldsymbol{z}} \quad . && (5.182)
\end{aligned}$$

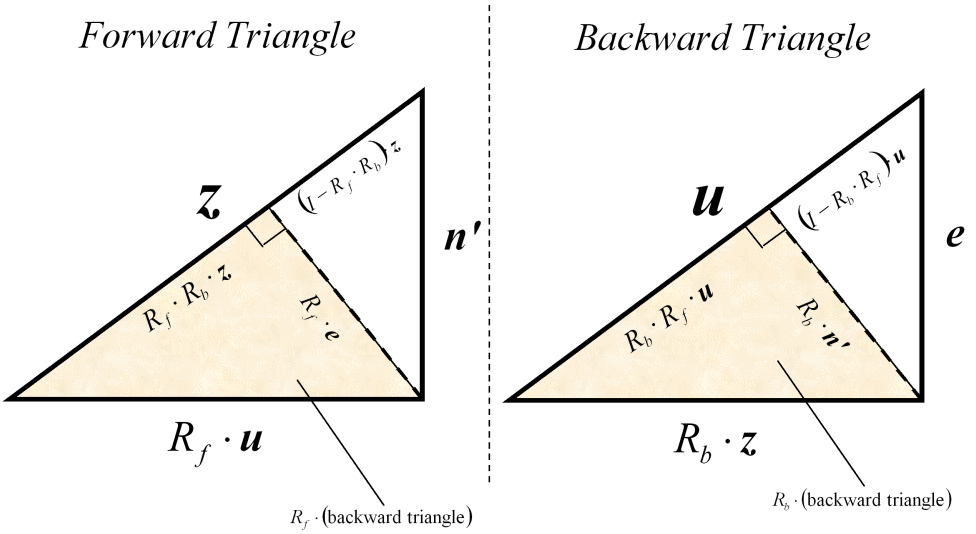


Figure 5.23: Canonical triangles.

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

$$I(\boldsymbol{u};\boldsymbol{z}) = I(\boldsymbol{u};\boldsymbol{y}) = I(\boldsymbol{x};\boldsymbol{y}) \quad , \tag{5.183}$$

so that the new channels between $\boldsymbol{u}$ and $\boldsymbol{z}$ carry all the pertinent information between channel input and output. The backward and forward canoncial channels both have the same mutual information. Transmission from $\boldsymbol{u}$ to $\boldsymbol{z}$ has the same performance as $\boldsymbol{z}$ to $\boldsymbol{u}$, and as from $\boldsymbol{x}$ to $\boldsymbol{y}$.

5.3.3 Generalized Decision Feedback Development

The GDFE exploits the structure in the canonical channel models in the same way that this structure was exploited for infinite-length in Section 5.1.3.

The backward canonical model is

$$\boldsymbol{u} = R_b \boldsymbol{z} + \boldsymbol{e} \quad , \tag{5.184}$$

where R_b is nonsingular because of the earlier elimination of singular dimensions to get $\boldsymbol{u} \rightarrow \boldsymbol{z}$. The inverse MMSE "spectrum" R_b^{-1} has an lower-diagonal-upper Cholesky factorization

$$R_b^{-1} = G^* \boldsymbol{S}_0 G \quad , \tag{5.185}$$

where G is upper triangular and monic (ones along the diagnonal), and $\boldsymbol{S}_0$ is a diagonal matrix of positive Cholesky factors. Then

$$R_b = G^{-1} \boldsymbol{S}_0^{-1} G^{-*} \quad . \tag{5.186}$$

Premultiplication of $\boldsymbol{u}$ in (5.184) by G produces

$$\boldsymbol{u}' = G\boldsymbol{u} = \boldsymbol{S}_0^{-1} G^{-*} \boldsymbol{z} + \boldsymbol{e}' = \boldsymbol{z}' + \boldsymbol{e}' \quad , \tag{5.187}$$

where $\boldsymbol{e}' = G\boldsymbol{e}$ has an autocorrelation matrix $\boldsymbol{S}_0$, and $\boldsymbol{z}'$ is the output of a matrix feedforward filter $\boldsymbol{S}_0^{-1} G^{-*}$ acting on the matched filter output $\boldsymbol{z}$. Again, mutual information is maintained because the triangular matrix G is one-to-one, so

$$I(\boldsymbol{u};\boldsymbol{z}') = I(\boldsymbol{x};\boldsymbol{y}) \quad . \tag{5.188}$$

The channel input vector $\boldsymbol{u}$ can be recovered from

$$\boldsymbol{z}' = G\boldsymbol{u} - \boldsymbol{e}' \tag{5.189}$$

by a process known as back substitution: A detailed matrix description of (5.189) is

$$\begin{bmatrix} z'_{N^*-1} \\ z'_{N^*-2} \\ \vdots \\ z'_0 \end{bmatrix} = \begin{bmatrix} 1 & g_{N^*-1,N^*-2} & \cdots & g_{N^*-1,0} \\ 0 & 1 & \cdots & g_{N^*-2,0} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \end{bmatrix} \begin{bmatrix} u_{N^*-1} \\ u_{N^*-2} \\ \vdots \\ u_0 \end{bmatrix} - \begin{bmatrix} e'_{N^*-1} \\ e'_{N^*-2} \\ \vdots \\ e'_0 \end{bmatrix} \quad , \tag{5.190}$$

where the convention of writing the first time sample in each block at the bottom of a vector has been maintained even though the process of elimination of the null space of P and $R_{\boldsymbol{xx}}$ earlier may have effectively "reordered the sample index" within the symbol. When $R_{\boldsymbol{uu}}$ is diagonal, by starting at the bottom of (5.190) and ascending, back-substitution solution of (5.190) obtains the decision-feedback equations:

$$\hat{u}_0 = \text{sbs}_0(z'_0) \tag{5.191}$$

$$\hat{u}_1 = \text{sbs}_1(z'_1 - g_{1,0}\hat{u}_0) \tag{5.192}$$

$$\hat{u}_2 = \text{sbs}_2(z'_2 - g_{2,1}\hat{u}_1 - g_{2,0}\hat{u}_0) \tag{5.193}$$

$$\vdots = \vdots \tag{5.194}$$

$$\hat{u}_{N^*-1} = \text{sbs}_{N^*-1}(z'_{N^*-1} - \sum_{i=0}^{N^*-2} g_{N^*-1,i}\hat{u}_i) \tag{5.195}$$

The "sbs_i" operation is just symbol-by-symbol detection for whatever constellation is used on the i^{th} transmitted subsymbol u_i, $i = 0, ..., N^* - 1$. Because of this symbol-by-symbol slicing, decision feedback is suboptimal (that is not a maximum likelihood detector for the whole sytmbol), but recursively and simply implemented. The feedback "filters" are a function of the sample-time index and represent a periodically time-varying[15] feedback section of the DFE. The period is the symbol period. Similarly, the feedforward section is also periodically time-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 5.3.3. The n^{th} tone's MSE is

$$E[|e'_n|^2] = S_{0,n}^{0-1} = \frac{1}{S_{0,n}} \quad . \tag{5.196}$$

When $\boldsymbol{u}$ is not white, then it can be related to a white signal that carries the information, leading to various interesting special cases as in the next subsection and also the next full section.

The GDFE is a set of parallel subchannels with (biased) SNR's equal to

$$\text{SNR}_{bias,n} = 1 + \text{SNR}_{u,n} = \frac{E[|u_n|^2]}{E[|e'_n|^2]} = \mathcal{E}_{u,n} \cdot S_{0,n} \tag{5.197}$$

where this expression is similar to the expression $\text{SNR}_{\text{mmse-dfe}} = \gamma_0 \cdot \frac{\bar{\mathcal{E}}_{\boldsymbol{x}} \cdot \|p\|^2}{\frac{N_0}{2}}$ with $S_{0,n}$ replacing $\gamma_0 \cdot \frac{\|p\|^2}{\frac{N_0}{2}}$ for the infinite-length DFE. Equation (5.197) uses a subscript u on $\mathcal{E}_{u,n}$ that stands for the input vector u - care should be exercised to avoid confusing this with the "u" for unbiased or for the "u" user index of Chapters 12-14 for readers referring back to GDFE material from those chapters. $S_{0,n}$ has a role similar to the (scaled) γ_0 of the infinite-length equivalent of Cholesky factorization, which is the canonical factorization in Chapter 3 when the channel, noise, and input processes are stationary. The product of these biased SNR's is then

$$\text{SNR}_{\text{gdfe}} = \left(\prod_{n=1}^{N^*} \text{SNR}_{bias,n} \right)^{\frac{1}{N+\nu}} \leq \frac{|R_{\boldsymbol{uu}}|^{1/(N+\nu)}}{|R_{\boldsymbol{e'e'}}|^{1/(N+\nu)}} = 2^{2\bar{I}(\boldsymbol{u},\boldsymbol{z'})} = 2^{2\bar{I}(\boldsymbol{x},\boldsymbol{y})} \quad , \tag{5.198}$$

[15] Or more generally, periodically "index-varying." The period is the symbol period.

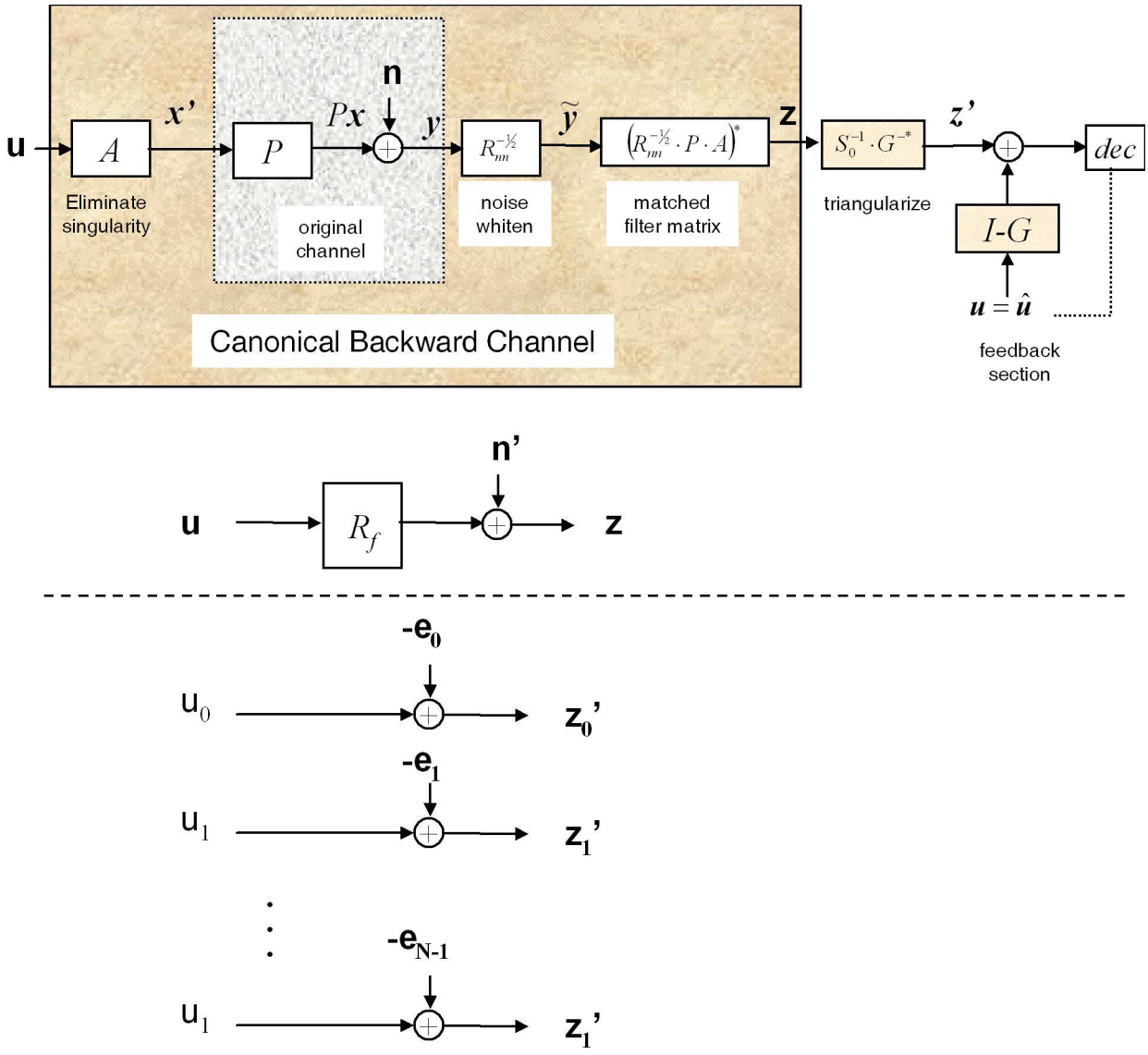


Figure 5.24: Generalized DFE.

with less than (and not just equal) because of the information-lossy action of slicing in the feedback operation. Equality occurs when $R_{\boldsymbol{uu}} = \text{diag}\{\mathcal{E}_{u,N^*-1},\ ...,\ \mathcal{E}_{u,0}\}$. With such a diagonal (white) input, this is a finite-length CDEF result. A suboptimal detector, namely the GDFE, is exhibiting an SNR on an equivalent set of independent AWGN channels that is equal to the best achievable for such a set. This surprising result occurs only for the backward canonical model of a channel. For this symbol-length, the same good codes that are used on the ISI-free channel can be applied to a succession of symbols in the GDFE system to acheive the same gap Γ with respect to the mutual information rate. If $R_{\boldsymbol{x}'\boldsymbol{x}'}$ is the same as that generated by water-filling on the $N+\nu$ discrete subchannels of P that vector coding would use, then transmission capacity can be approached by this GDFE structure, just as closely as with the VC system of Chapter 4, even though the slicing (symbol-by-symbol detector) is suboptimum. A white input (diagonal $R_{\boldsymbol{uu}}$ is required to attained this canonical level of performance.

Some interpretation is helpful: The GDFE replaces the linear filters of the usual DFE by matrix multiplies, indeed triangular matrix multiplies $(I - G)$ and $S_0 \cdot G^{-*}$ corresponding to a causal feedback section and a non-causal feed-forward section, respectively when P is Toeplitz (stationary channel). The matched filter is the multiply $(R_{\boldsymbol{nn}}^{-1/2}PA)^*$. These matrix multiplies can be interpreted as periodcially time-varying filters - a different set of coefficients for each index $n = 1, ..., N^*$ within the packet. The GDFE filters are thus then fundamentally different from the fixed finite-length filters that were derived in Chapter 3, and in some very special cases, converge to the same thing as $N \to \infty$. In the majority of cases, the GDFE filters do not converge to the same filters as in Chapter 3.

The matrix SNR for this channel is

$$\boldsymbol{SNR}_{\text{gdfe}} \triangleq R_{\boldsymbol{uu}} R_{\boldsymbol{ee}}^{-1} = \boldsymbol{SNR}_{\text{gdfe,u}} + I \quad , \tag{5.199}$$

and its determinant is $\text{SNR}_{\text{gdfe}} = |\boldsymbol{SNR}_{\text{gdfe}}|$. Thus

$$|\boldsymbol{SNR}_{\text{gdfe}}| = |\boldsymbol{SNR}_{\text{gdfe,u}} + I| = \prod_{i=0}^{N^*-1} (1 + \text{SNR}_{u,i}) \quad . \tag{5.200}$$

A task remaining is to make sure that $\boldsymbol{u}$ corresponds to a white input, for which there are several methods discussed in Sections 5.4 and 5.5. If $R_{\boldsymbol{uu}}$ is not diagonal, then $\text{SNR}_{\text{gdfe}} < 2^{2\bar{I}(\boldsymbol{x};\boldsymbol{y})}$ in (5.198) (that is, the inequality is strict).

Bias

The GDFE is biased like all MMSE detectors. Each subchannel is a MMSE estimate and thus has the relation $\text{SNR}_i = \text{SNR}_{u,i} + 1$. The matrix snr $\boldsymbol{SNR}_{\text{gdfe}}$ corresponds to a biased estimate. That is

$$E[\hat{u}_n / u_n] = (1 - \frac{1}{\text{SNR}_{bias,n}}) u_n \neq u_n \quad . \tag{5.201}$$

Bias can be removed individually after feedback on the subchannels or by scaling each output of the feedforward matrix by $\frac{\text{SNR}_{bias,n}}{\text{SNR}_{bias,n}-1}$.

5.3.4 The GDFE with diagonal input

For a general desired or given $R_{\boldsymbol{xx}}$, the "input-essence" vector $\boldsymbol{u}$ may not have diagonal $R_{\boldsymbol{uu}}$. However, the vector $\boldsymbol{u}$ can be related to a "white input" (diagonal autocorrelation matrix so that all inputs are uncorrelated) $\boldsymbol{v}$ in general via some $N^* \times N^*$ square matrix transformation:

$$\boldsymbol{u} = \Phi \boldsymbol{v} \quad , \tag{5.202}$$

where $R_{\boldsymbol{vv}}$ is diagonal. Then, the matrix $\tilde{A}$ is replaced by

$$\tilde{A} \leftarrow \tilde{A}\Phi \tag{5.203}$$

and all the development of this section for the GDFE follows with the simple replacements

$$\tilde{P} \leftarrow R_{\boldsymbol{nn}}^{-1/2} P \tilde{A} \Phi \tag{5.204}$$

$$\boldsymbol{u} \leftarrow \boldsymbol{v} \quad . \tag{5.205}$$

Figure 5.3.3 replaces Figure 5.25 with such substitution. After $\boldsymbol{v}$ has been correctly estimated, then $\boldsymbol{u}$ and in turn $\boldsymbol{x}'$ can be reconstructed. Any part of $\boldsymbol{x}$ not contained in $\boldsymbol{x}'$ is of course lost in transmission, but when $\boldsymbol{x} = \boldsymbol{x}'$, the arbitary input is determined (as long as the GDFE symbol-by-symbol detectors function without error in estimation). In practice of course $\boldsymbol{x}$ and consequently $\boldsymbol{v}$ is never quite Gaussian and instead is a discrete distribution corresponding to messages, even though with good codes that distribution may be close to Gaussian. In this case then, it is desirable to consider $\boldsymbol{v}$ as the message sequence itself and $\boldsymbol{x}'$ as derived from this message sequence in modulation. Thus $A\Phi$ is a matrix transmit filter. The GDFE then directly functions to estimate the message sequence. In this case, thet simple QAM/PAM modulation needs modification according to loading of information on to the set of parallel channels, to obtain maximum data rate. Thus, fixed-constellation QAM/PAM with an arbitary number of bits or even a number of bits per symbol equal to the packet mutual information $\bar{I}(\boldsymbol{x};\boldsymbol{y})$ are not canonical in general for the finite-length packet channel.

The next section investigates two specific structures for the matrix Φ and finds that a triangular factorization will lead to a finite-length equivalent of the MMSE-DFE. The other structure, an eigen-decomposition, leads to VC. Both are special cases of the GDFE.

There can be many such transformations and each will correspond to a different GDFE, but all will attain the same performance. The following example illustrates one such case.

EXAMPLE 5.3.3 (GDFE for $1 + .9D^{-1}$ Channel) This book has investigated numerous times the $1 + .9D^{-1}$ channel with input energy $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$ and $\sigma^2 = .181$. This current instance will investigate a form of the GDFE for this channel with $N = 2$ and $\nu = 1$ to keep computation to a minimum. The matrix $R_{\boldsymbol{nn}}^{-1/2} P$ describing this channel is

$$\frac{1}{\sigma} \cdot P = \frac{1}{\sqrt{.181}} \cdot \begin{bmatrix} .9 & 1 & 0 \\ 0 & .9 & 1 \end{bmatrix} = \begin{bmatrix} 2.1155 & 2.3505 & 0 \\ 0 & 2.1155 & 2.3505 \end{bmatrix} \quad , \tag{5.206}$$

with singular value decomposition:

$$\frac{1}{\sqrt{2}} \cdot \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 3.8694 & 0 & 0 \\ 0 & 2.2422 & 0 \end{bmatrix} \begin{bmatrix} .3866 & .6671 & .6368 \\ .8161 & .0741 & -.5731 \\ .4295 & -.7412 & .5158 \end{bmatrix}^* \quad . \tag{5.207}$$

The first two columns of M span the pass space of the channel. The original PAM modulator corresponds to $C = I$, so

$$\tilde{C} = P_{\tilde{M}} \cdot I = \begin{bmatrix} .5945 & .3649 & -.3285 \\ .3649 & .6715 & .2956 \\ -.3285 & .2956 & .7340 \end{bmatrix} \quad , \tag{5.208}$$

which in this special case of white PAM modulation corresponds to a rank-2

$$R_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}} = P_{\tilde{M}} \cdot I P_{\tilde{M}} = \tilde{C} \quad . \tag{5.209}$$

Since $R_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}}$ has rank 2, then $N^* = \tilde{N}$ and no further elimination of singularity is necessary. However, $R_{\boldsymbol{uu}}$ is not white. By using the steps in Figure 5.20,

$$R_{\boldsymbol{uu}} = \begin{bmatrix} 2.5242 & -1.3717 \\ -1.3717 & 2.2346 \end{bmatrix} \quad , \tag{5.210}$$

and

$$u_1 = x_1 - 1.235 x_3 \tag{5.211}$$

$$u_2 = x_2 + 1.111 x_3 \tag{5.212}$$

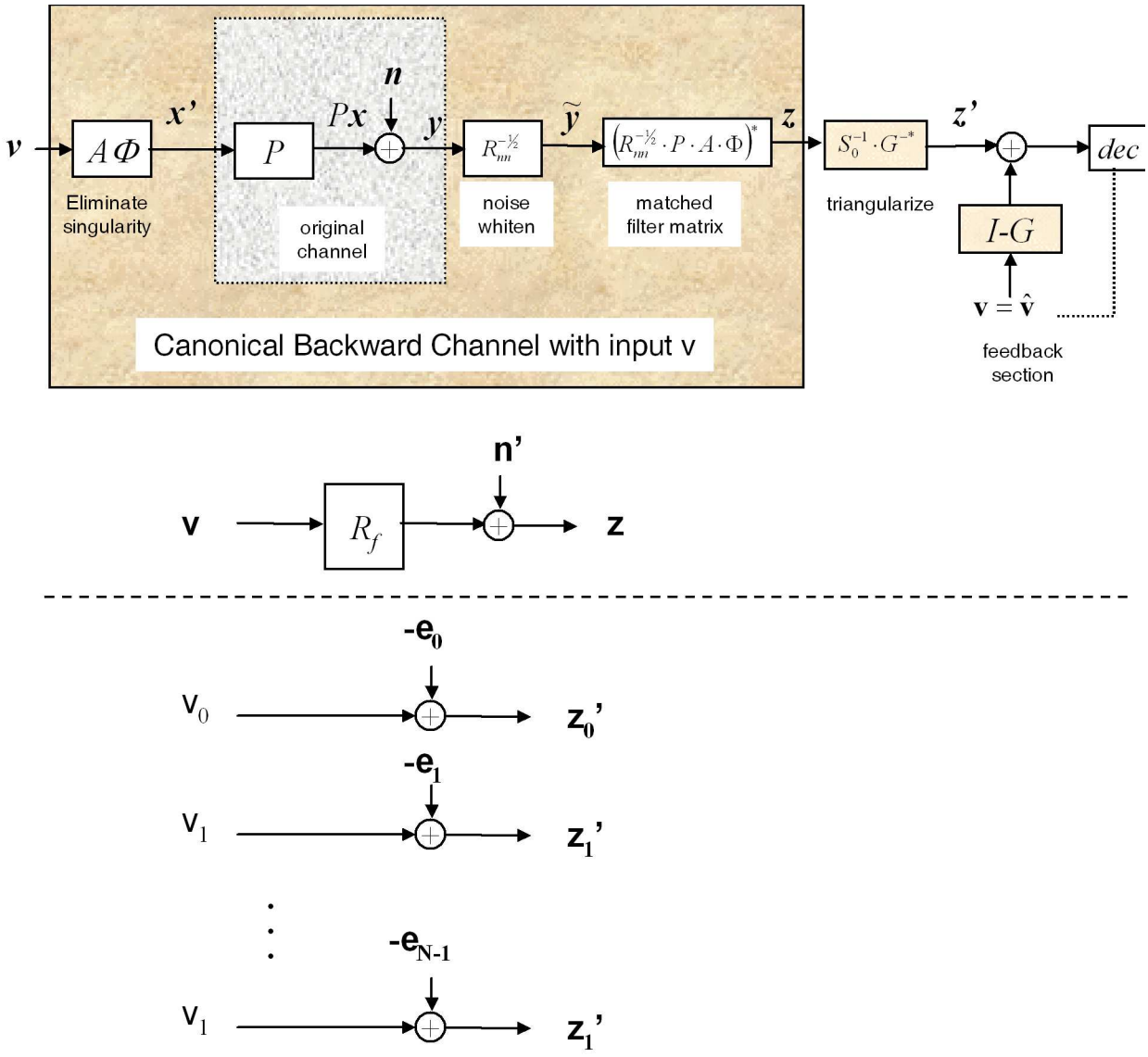


Figure 5.25: The Generalized DFE with white input $\boldsymbol{v}$

A GDFE based on estimating u_1 and u_2 will therefore not have canonical peformance.

Instead, the designer might at least attempt to find an equivalent white input $R_{\boldsymbol{vv}}$ that results in the same $R_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}}$ as $\boldsymbol{u}$ as per the following matlab commands:

```
>> rxxtt=Pm
rxxtt = 0.5945    0.3649   -0.3285
        0.3649    0.6715    0.2956
       -0.3285    0.2956    0.7340
>> [V,D]=eig(rxxtt)
V = 0.6368    0.3545   -0.6847            D =  0     0     0
   -0.5731    0.8117   -0.1128                 0     1     0
    0.5158    0.4642    0.7200                 0     0     1

>> Pq=V(1:3,2:3)*V(1:3,2:3)' =  0.5945    0.3649   -0.3285
                                0.3649    0.6715    0.2956
                               -0.3285    0.2956    0.7340

>> Phi=V(1:3,2:3)=  0.3545   -0.6847
                    0.8117   -0.1128
                    0.4642    0.7200

>> Phi*eye(2)*Phi' =  0.5945    0.3649   -0.3285   = rxtt (checks)
                      0.3649    0.6715    0.2956
                     -0.3285    0.2956    0.7340
>> P*Phi
    2.6578   -1.7136
    2.8083    1.4538

>> P*Pm*Phi
    2.6578   -1.7136
    2.8083    1.4538

>> Pt=P*Phi =  2.6578   -1.7136
               2.8083    1.4538
```

A GDFE based on the 2-dimensional white input $R_{\boldsymbol{vv}} = I$ achieves canoncial performance, which is found by the following sequence of matlab commands:

```
>> Rf=Pt'*Pt = 14.9500   -0.4715
               -0.4715    5.0500

>> Rbinv=Rf+inv(eye(2)) = 15.9500   -0.4715
                          -0.4715    6.0500

>> Gbar=chol(Rbinv) = 3.9937   -0.1181
                         0    2.4568

>> G=inv(diag(diag(Gbar)))*Gbar =  1.0000   -0.0296
                                      0    1.0000
>> S0=diag(diag(Gbar))*diag(diag(Gbar))

S0 =  15.9500         0
           0    6.0361
>> snr=10*log10((det(S0))^(1/3)-1) =
```

```
    5.5427 dB
```

The canonical performance is thus $\text{SNR}_{gdfe,u} = 5.5$ dB. The one unit of energy into the null space is wasted, pulling down performance if the white PAM-like autocorrelation matrix $R_{\boldsymbol{xx}} = I$ is used, even if the input is recoded so simple decisions on v_1 and v_2 can be made. The performance loss is caused by a poor choice in input autocorrelation matrix (which is nevertheless, that of the widely used PAM in this case). Assuming the matched filter matrix and the feedforward matrix are combined into a single 2×2 matrix, the complexity of this GDFE is 5 operations per 3 samples or 1.667/sample.

It is possible to attempt to recover some of the loss by reshaping the input so that no energy is placed into the null space and 1.5 units each are used on each of the components of the white $\boldsymbol{v}$. This provides the autocorrelation matrix

$$R_{\boldsymbol{xx}} = \begin{bmatrix} 0.8918 & 0.5474 & -0.4927 \\ 0.5474 & 1.0073 & 0.4434 \\ -0.4927 & 0.4434 & 1.1009 \end{bmatrix} \quad . \tag{5.213}$$

but

$$R_{\boldsymbol{vv}} = 1.5I \quad . \tag{5.214}$$

and

$$rxx = \Phi R_{\boldsymbol{vv}} \Phi^* \tag{5.215}$$

The corresponding GDFE now is computed according to these matlab commands:

```
>> Rbinv=Rf+inv(1.5*eye(2)) = 15.6166   -0.4715
                              -0.4715    5.7167
>> Gbar=chol(Rbinv)
Gbar = 3.9518   -0.1193
            0    2.3880

>> G=inv(diag(diag(Gbar)))*Gbar  =  1.0000   -0.0302
                                        0    1.0000

>> S0=diag(diag(Gbar))*diag(diag(Gbar)) =   15.6166         0
                                                0    5.7025

>> snr=10*log10((2.25*det(S0))^(1/3)-1) =
    6.8589 dB
>> W=inv(S0)*inv(G')=  0.0627         0
                       0.0049    0.1657
>> W*Pt' (MS-WMF combination)
    0.1666    0.1761
   -0.2709    0.2546
```

The SNR increases to 6.9 dB and the feedforward filter and the combination of feedforward-matched filter are illustrated. The transmit energy is 3 units per symbol. The complexity remains at 1.667/sample. 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 gets easier, because this $\mathcal{E}_1 = \mathcal{E}_2 = 1.5$ choice of inputs is close to water-filling. For $N = 3$ in the GDFE, one gets 8.0 dB and as N becomes large, the ultimate SNR is 8.4 dB for an i.i.d. input. Unfortunately, as N grows, the complexity of this choice of GDFE will grow as N^2 and unnecessarily be well above the complexity of DMT (which gets 8.8 dB when $N = 16$ and complexity is 3.8/sample). If a water-fill input is used and the water-filling input is used, then this ultimate SNR can be as high as 8.8 dB.

When $R_{\boldsymbol{uu}}$, or equivalently by design $R_{\boldsymbol{vv}}$, are diagonal, the GDFE creates a set of parallel channels that have the same maximum data rate possible as VC. This set of parallel channels is different than those of vector coding for the same $R_{\boldsymbol{xx}}$ and have a different set of SNR_n. However, the total data rate and overall SNR are the same. Loading needs to be excuted for this set of nonequal SNR_n in order to get the maximum performance level. The standard loading algorithms of Chapter 4 can be applied to the set of parallel independent subchannels created by the GDFE.

Chain rule relationships

The chain rule of mutual information

$$I(\boldsymbol{x}\ \boldsymbol{y}) = I(x_0;\boldsymbol{y}) + I(x_1\ \boldsymbol{y}/x_0) + ... + I(x_n\ \boldsymbol{y}/[x_1...x_{n-1}]) + ... + I(x_{N-1}\ \boldsymbol{y}/[x_1...x_{N-2}]) \tag{5.216}$$

has a very special relationship to all GDFE's. Essentially each term in equation 5.216 corresponds to a MMSE-DFE designed for x_n given the entire channel symbol output $\boldsymbol{y}$ and the previous values of x_1, ... x_n. A little reflection reveals this is exactly what the GDFE is – a series of independently designed MMSE-DFEs given the channel output and past decisions – when the input sequence has diagonal autocorrelation, i.e., is "white".

Any invertible transformation on the input from a possibly "non-white" $\boldsymbol{x}$ to a "white" $\boldsymbol{v}$ produces the same mutual information $I(\boldsymbol{v}\ \boldsymbol{y}) = I(\boldsymbol{x}\ \boldsymbol{y})$and a new chain rule for the components of $\boldsymbol{v}$. There are an infinite number of such possible transformations, leading to an infinite number of GDFEs, as previous sections have illustrated. This same chain-rule applied to the infinite-length MMSE-GDFE in Subsection 5.1.2. This same result applies also to the GDFE, as in the subsequent subsection.

Zero-Forcing GDFE's

The zero-forcing version of any MMSE design, like the GDFE, is obtained by assuming there is no noise in the MMSE solution. 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 noise.

For any given GDFE, the ZF-equivalent uses the forward channel model to derive a ZF-GDFE. The forward channel output $\boldsymbol{z}$ is again with lower-diagonal-upper Cholesky factorization of $R_f = P_f^* S_f P_f$

$$\boldsymbol{z} = P_f^* S_f P_f \boldsymbol{u} + \boldsymbol{n}' \quad . \tag{5.217}$$

The receiver can process $\boldsymbol{z}$ with $S_f^{-1} P_f^{-*}$ to obtain

$$\boldsymbol{z}' = S_f^{-1} P_f^{-*} \boldsymbol{z} = P_f \boldsymbol{u} + \boldsymbol{n}'' \quad . \tag{5.218}$$

Ignoring the noise $\boldsymbol{n}''$ as if it were zero produces a triangular set of equations (since P_f is upper triangular)

$$\begin{aligned}
\hat{u}_0 &= \text{sbs}_0(z'_0) && (5.219)\\
\hat{u}_1 &= \text{sbs}_1(z'_1 - p_{f,1,0}\hat{u}_0) && (5.220)\\
\hat{u}_2 &= \text{sbs}_2(z'_2 - p_{f,2,1}\hat{u}_1 - p_{f,2,0}\hat{u}_0) && (5.221)\\
\vdots &= \vdots && (5.222)\\
\hat{u}_{N^*-1} &= \text{sbs}_{N^*-1}(z'_{N^*-1} - \sum_{i=0}^{N^*-2} p_{f,N^*-1,i}\hat{u}_i) \quad . && (5.223)
\end{aligned}$$

The set of equations provides an estimate of the transmitted sequence, but the overall SNR (which is unbiased) is less than or equal to $2^{2\bar{I}(\boldsymbol{x};\boldsymbol{y})} - 1$. This SNR can be computed as (with 0 dB gap)

$$\text{SNR}_{zf-gdfe} = \left[\prod_{n=1}^{N} (1 + \mathcal{E}_{u,n} \cdot S_{f,n})\right]^{1/N} - 1 \quad . \tag{5.224}$$

Equality holds under various conditions like the matrix R_f is diagonal (that is $P_f = I$ and there is no feedback matrix) or certain types of noise (worst-case), as will be seen later in Section 5.5. Often, the ZF-GDFE, can have performance very close to the MMSE-GDFE cis anonical and highest SNR level.

5.3.5 Precoding for the GDFE

Precoding in the conventional DFE averts error propagation by essentially moving the feedback section to the transmitter. Section 3.8 introduced several types of precoders, with the Tomlinson and/or Laroia precoders (or more advanced lossless precoders of Chapter 14) of general utility. Precoders can increase the transmit power slightly. Some straightforward modification is necessary for the GDFE. However, the finite-length symbols in the GDFE prevent error bursts that exceed the length of one symbol in length. Thus, there may be less need for the precoder in the GDFE because any error-propagation will hault with probability one at the symbol boundary, unlike the conventional DFE that can have infinite error bursts (and very long error bursts with increasingly small probability). Thus, for reasons of performance and error propagation, the GDFE may be preferable in implementation.

Nonetheless, the transmission design may use precoding to eliminate even error propagation within a symbol. The rows of the feedback section now vary with subchannel index n and

$$\boldsymbol{g}_n \triangleq [1 \; g_{n,n-1} \; g_{n,n-2} \; ... \; g_{n,0}] \; \forall n = 0, ..., N^* - 1 \quad . \tag{5.225}$$

The Tomlinson precoder again requires an unbiased feedback section, which is obtained by

$$\boldsymbol{g}_{U,n} = \left[1 \; \frac{\text{SNR}_n + 1}{\text{SNR}_n} \cdot g_{n,n-1} \; ... \; \frac{\text{SNR}_n + 1}{\text{SNR}_n} \cdot g_{n,0}\right] \quad . \tag{5.226}$$

Each used dimension $n = 0, ..., N^* - 1$ has an associated $b_n = \frac{1}{2}\log_2\left(1 + \frac{\mathcal{E}_n \cdot S_n}{\Gamma}\right)$ and constellation, for which a modulato operator is defined as in Section 3.8, $\Gamma_{2^{2\bar{b}_n}}(\bullet)$. The Tomlinson precoder then becomes period in the symbol period with operation

$$u'_n = \Gamma_{2^{2\bar{b}_n}}\left(u_n - \sum_{i=0}^{n-1} g_{U,n,i} \cdot u'_i\right) \quad . \tag{5.227}$$

Each dimension has a loss in SNR_n of $\frac{2^{2\bar{b}_n}}{2^{2\bar{b}_n}-1}$, which can then be incorporated into the overall SNR for the GDFE through the same formula (5.73). Each of the N^* dimensions of the feedforward matrix output should have a modulo operator $\Gamma_{2^{2\bar{b}_n}}(\bullet)$ prior to the decision element.

The Laroia precoder then also easily follows with the periodic $\boldsymbol{g}_{U,n}$ vectors being used in the usual manner for each corresponding dimension – however, the Laroia precoder cannot be used if receiver coordination is necessary because it requires a second wave of corresponding decoder processing as in Section 3.8 after a first decision element (and this additional processing requires coordination among the dimensions in the receiver, see Chapter 14 on broadcast channels for more on the absence of receiver coordination).

5.3.6 Single-sided GDFE's for distributed channel inputs

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

- the cyclic matrix P corresponding to use of a cyclic prefix
- a matrix that corresponds to N dimensions independently exciting a common channel (for instance a wireless transmission system with N independent transmit antennae and N or more receive antennae)

The Vector OFDM systems of Section 5.5 also decompose into a number of smaller systems, each with such a nonsingular square P matrix.

In these situations only, the GDFE performance can be attained without coordination or co-generation of the input dimensions. The absence of an ability to transform the input dimensions with an A or C matrix might correspond to physically distinct locations in which the transmitted signals entering the

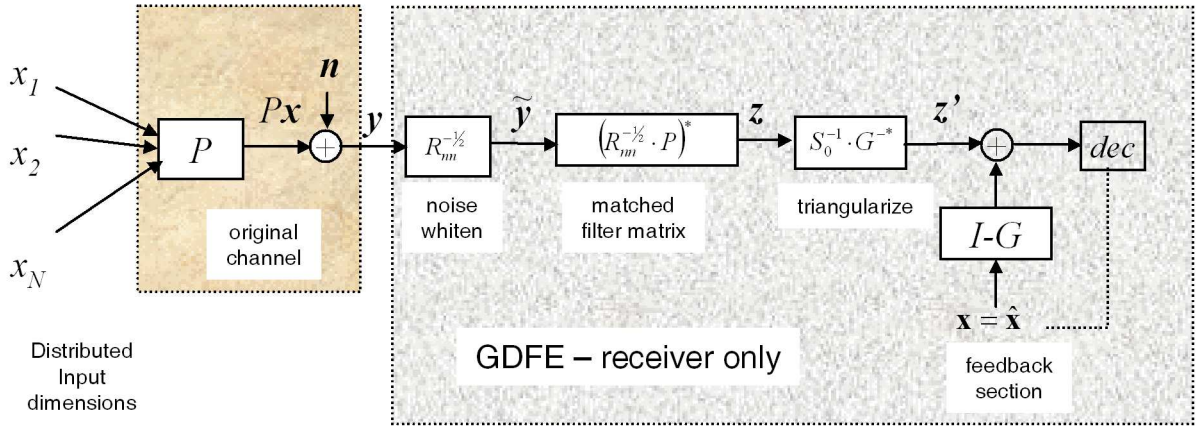


Figure 5.26: The distributed input GDFE.

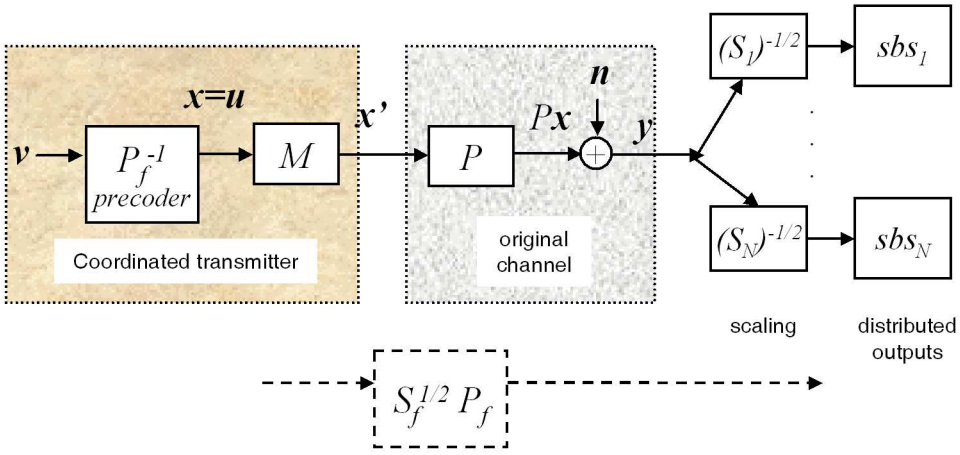


Figure 5.27: The distributed-output GDFE.

channel are generated, or **distributed inputs**. If the P matrix has rank greater than or equal to the number of inputs, $A = C = I$, and thus no dimensional reduction is necessary.

Figure 5.26 illustrates the GDFE in the case of distributed input dimensions, and it is clear that all processing of the GDFE occurs in the receiver (which must be able to co-process the various channel output dimensions). While seemingly trivial as written here – this particular situation of receiver-side-coordination-only can occur in a number of multiple-input-multiple-output channel situations (see Chapter 13).

5.3.7 Single-sided GDFE-Precoder for distributed channel outputs

The dual of the situation in Subsection 5.3.6 is one in which the channel output dimensions occur in physically separated locations, and thus matrix processing is not physically possible. However, the input dimensions to the channel are co-generated in a common place. The ZF-GDFE performance is possible only if the matrix P is square and nonsingular[16].

Figure 5.27 illustrates this situation. In this case, the matrix P is QR-factored into $P = S_f^{1/2} P_f M^*$

[16]More channel outputs than input dimensions is not of practical interest if the outputs cannot be coordinated in processing.

where S_f is the diagonal matrix from the ZF-GDFE, P_f is the monic upper triangular matrix and M is a Hermetian matrix. Then the transmit signal is $\boldsymbol{x} = M\boldsymbol{u}$ and the receive signal is processed by $\boldsymbol{z}' = S_f^{-1/2}\boldsymbol{z}$, leaving

$$\boldsymbol{z}' = P_f\boldsymbol{u} \tag{5.228}$$

A precoder is then used to create the set of parallel independent subchannels of the ZF-GDFE. This choice leads to the overall SNR being

$$\text{SNR}_{zf-gdfe} = \prod_{n=0}^{N-1} \mathcal{E}_{u,n} \cdot S_{f,n} \quad . \tag{5.229}$$

If the noise is not white, then P should be replaced by $PR_{\boldsymbol{nn}}^{-1/2}$ in all the above equations, and there may be additional performance loss with respect to GDFE levels.

Precoding in the MMSE sense can be done if P is singular and/or the input is singular in the usual sense with a GDFE. More general precoders are considered in Chapter 14. This means the matrix A is found and the input processing determined accordingly. The feedback section is implemented at the transmitter for G of the GDFE. However, the combination of the matched-filter feedforward filter $S_0^{-1}G^{-*}\tilde{P}$ would nominally require co-location of the recievers, which is not physically possible. However, there are many choices of the GDFE input processing that all provide the same performance, but different feedforward filters and feedback sections. A question is "Is there a choice of partitionings and white inputs $\boldsymbol{v}$ that produces a diagonal $W = S_0^{-1}G^{-*}\tilde{P}$?"

There are basically two "yes" answers when the system is square nonsingular:

1. If the noise is white and water-filling spectra (see Section 5.5.5) are also used, then the MS-WMF matrix can be diagonal with no performance loss.

2. If the noise is worst-case noise (see Subsection 5.5.5).

5.4 Special Cases of the GDFE: VC and The Packet DFE

A generalized DFE for a linear-ISI-with-AWGN channel $(P, R_{\boldsymbol{nn}})$ has the same performance for all Gaussian inputs with covariance $R_{\boldsymbol{uu}}$. The design of a realistic GDFE system should consider the message sequence to be the diagonal-autocorrelation input vector $\boldsymbol{v}$ of Figure 5.25 since only decisions made on v_n $n = 0, ..., N^* - 1$ achieve the cannonical performance maximum of $\boldsymbol{SNR} = 2^{2\bar{I}(\boldsymbol{x};\boldsymbol{y})}$. This input choice of diagonal $\boldsymbol{v}$ then generates any useful part $R_{\boldsymbol{x}'\boldsymbol{x}'}$ of the desired input autocorrelation/power-spectrum corresponding to $R_{\boldsymbol{xx}} = R_{\boldsymbol{x}'\boldsymbol{x}'}$.

The GDFE with white input has canonical performance, just as did the infinite-length GDFE (when inputs were properly constructed according to symbol-rate choice) and thus allows the following CDEF Lemma:

Lemma 5.4.1 (Finite-Length CDEF Equivalent) *The finite-length equivalent of the earlier CDEF result is that for a diagonal input $\boldsymbol{v}$ and any corresponding GDFE of Section 5.3:*

$$SNR_{GDFE,u} = 2^{2\bar{I}(\boldsymbol{x};\boldsymbol{y})} - 1 \quad . \tag{5.230}$$

That is a code with gap Γ on the AWGN can also be used (with appropriate loading) on all the subchannels of the GDFE for any input autocorrelation $R_{\boldsymbol{xx}}$ to obtain the same gap to best data rate for that $R_{\boldsymbol{xx}}$ on matrix channel P.

Proof: *the development in Section 5.3.* **QED.**

Some interpretation is necessary: In the GDFE, a set of parallel subchannels that do not all carry the same information is generated (unlike the CDEF result for the infinite-length case where all subchannels carry the same information or $\bar{b}$). The data rate on each of these subchannels will be determined by the given diagonal entries in $R_{\boldsymbol{vv}}$ and are

$$\bar{b}_n = \frac{1}{2}\log_2\left(1 + \frac{\mathcal{E}_{v,n} \cdot S_{0,n}}{\Gamma}\right) \quad . \tag{5.231}$$

When these subchannels have been loaded according to to a water-filling criterion for an $R_{\boldsymbol{xx}}$ that maximizes $I(\boldsymbol{x};\boldsymbol{y})$, then the highest possible data rate of capacity is obtained for $\Gamma = 0$ dB. This section will find ways of constructing such an input. Furthermore, codes applied to a GDFE system may span several successive symbols in order to ensure that the gap is correct – that is codes with gaps less than 8.8 dB (for $P_e = 10^{-6}$) may correspond to more than one symbol of the GDFE system, but are considered external to that GDFE system and simply achieve the gap – this prsumption of infinite codeword length is not an issue in the theoretical infinite-length systems, but clearly should be at least noted in finite-length implementation.

There may be more than one diagonal-autocorrelation $\boldsymbol{v}$ that achieves 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 Φ resulting from eigendecomposition of $R_{\boldsymbol{uu}}$ to obtain Hermetian Φ

2. **triangular GDFE** - which corresponds to Cholesky factorization of $R_{\boldsymbol{uu}}$ to obtain triangular (causal) Φ, and includes the important special case of cyclic GDFE or the CDFE when P is cyclic, only the latter of which corresponds to the original DFE's of Chapter 3.

As $N \to \infty$ for stationary (Toeplitz) $R_{\boldsymbol{uu}}$, the CDFE converges to the MMSE-DFE, while other forms of the GDFE do not. The CDFE will at each symbol length $N + \nu$ and cyclic prefix length ν exactly match the SNR of DMT, if the spectra of the CDFE input is the same as the spectra of the DMT input. Vector coding, again, converges to multitone as $N \to \infty$ for stationary (Toeplitz) $R_{\boldsymbol{uu}}$. Thus, the GDFE and all its white-input special cases of CDFE, DMT, and VC all asymptotically approach the same performance level if the $R_{\boldsymbol{xx}}$ is the same (and each method for uses codes with the same gap), even if that $R_{\boldsymbol{xx}}$ is not a water-fill solution. If the water-fill $R_{\boldsymbol{xx}}$ is used in all, then all attain the highest performance level for the given gap. Section 5.5 considers the construction of the optimum water-filling or any other desired $R_{\boldsymbol{xx}}$ for the CDFE and GDFE to avoid violation of the PW criterion (which will result in a minimum number of parallel CDFE's).

5.4.1 Vector-Coding as a special case of the GDFE

The partitioning of vector coding is a special case of the GDFE. In this case, the operation of noise-whitening is best considered first so that the designer replaces P by $R_{\boldsymbol{nn}}^{-1/2}P$ and then derives the singular value decomposition

$$R_{\boldsymbol{nn}}^{-1/2}P = F\Lambda M^* \quad . \tag{5.232}$$

The input vectors of vector coding are again chosen to be the right-singular-vectors of the channel spanned by M,

$$\boldsymbol{x} = \sum_{n=1}^{N+\nu} X_n \boldsymbol{m}_n = M\boldsymbol{X} \quad . \tag{5.233}$$

Only $R_{\boldsymbol{uu}}$ of the form $R_{\boldsymbol{uu}} = MR_{\boldsymbol{XX}}M^*$ constitute vector-coding systems – any other $R_{\boldsymbol{uu}}$ is not vector coding. The projection into the pass space corresponds trivially to removal of the vectors $\boldsymbol{m}_n$ $n > N$ from the sum so

$$\tilde{\boldsymbol{x}} = \sum_{n=1}^{N} X_n \boldsymbol{m}_n = \tilde{M}\tilde{\boldsymbol{X}} \quad . \tag{5.234}$$

In vector coding, when water-filling is used, some parts of the pass space are so poor that they are not used – a classic example of the situation where $R_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}}$ is singular. The projection, in any case where $R_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}}$ is singular for VC, is then

$$\boldsymbol{x}' = \sum_{n=1}^{N^*} X_n \boldsymbol{m}_n = M'\boldsymbol{X}' \quad . \tag{5.235}$$

and finally then

$$v_n = X_n \;\; n = 1, ..., N^* \quad . \tag{5.236}$$

In the receiver, the matrix R_f becomes the diagonal

$$R_f = \tilde{P}^*\tilde{P} = \text{diag}\left\{\lambda_1^2, \lambda_2^2, ..., \lambda_{N^*}^2\right\} \quad , \tag{5.237}$$

and thus $R_b^{-1} = R_f + R_{\boldsymbol{vv}}^{-1}$ is also diagonal, meaning that $G = I$. The feedback section thus degenerates to no feedback at all (and the concern with suboptimal detection is eliminated), and the feedforward section simply becomes $S_0^{-1} = R_b$ – an independent scaling of each subchannel.

Lemma 5.4.2 (Vector Coding as the Optimum and Canonical Transmission System) *Vector coding is both canonical and optimum as a transmission method.*

Proof: Since $\text{SNR}_{dgfe} = 2^{2\bar{I}(\boldsymbol{x};\boldsymbol{y})}$ is maintained by the GDFE with diagonal inputs $R_{\boldsymbol{vv}}$, VC is canonical. Since there is no feedback section in the GDFE, the concern for suboptimal detection is removed and an independent ML decoder on each subchannel (which is a simple slicer for uncoded transmission) is optimum. **QED.**

The joint optimality and canonical nature of the VC-based GDFE is unique to a structure based on modulating vectors that coincide with a set of singular vectors for the channel. No other structure produces a diagonal channel with the consequence of $G = I$. Any other $G \neq I$ may have canonical SNR ($2^{2\bar{I}(\boldsymbol{x};\boldsymbol{y})}$) but assumes correct decisions and thus is not ML nor optimum as a detector. Thus, VC is truly the best structure for transmission on the channel $(P, R_{\boldsymbol{nn}})$.

5.4.2 The non-VC triangular DFE as a special case of the GDFE

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

$$R_{\boldsymbol{uu}} = G_\phi R_{\boldsymbol{vv}} G_\phi^* \quad , \tag{5.238}$$

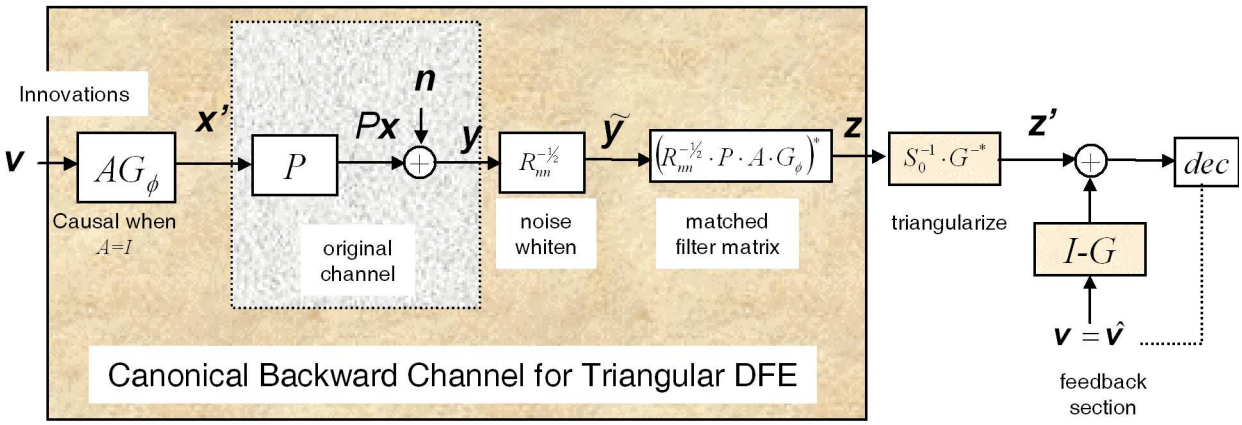


Figure 5.28: Triangular DFE with white input $\boldsymbol{v}$.

where G_ϕ is monic upper-triangular[17] and $R_{\boldsymbol{vv}}$ is a diagonal matrix. Furthermore,

$$\boldsymbol{u} = G_\phi \boldsymbol{v} \quad . \tag{5.239}$$

The data input for transmission will be considered to be the N^* components of $\boldsymbol{v}$, and

$$\boldsymbol{x}' = A\boldsymbol{u} = (AG_\phi)\boldsymbol{v} \quad . \tag{5.240}$$

The corresponding GDFE has

$$\tilde{P} = R_{\boldsymbol{nn}}^{-1/2} P A G_\phi \tag{5.241}$$

and otherwise follows exactly as in Section 5.3 with input v_n, $n = 0, ..., N^* - 1$ as data input and also as input to the feedback section in the GDFE. This structure appears in Figure 5.28.

The GDFE's feedforward filter and feedback filter remain triangular (and not necessarily diagonal) and nonzero in general, leading to a higher complexity of implementation and a loss of optimum detection performance – however, because the input is diagonal by construction, this GDFE is still canonical, the SNR_{gdfe} performance still approaches capacity just as closely as the VC system (with higher complexity). This GDFE's most interesting quality is the triangular transmission filter, which is suggestive of a causally implemented filter when the input $R_{\boldsymbol{xx}}$ can be made nonsingular as $N \to \infty$. Causal filters satisfy the PW criterion. Upper-triangular filters can be implemented essentially without delay in this special case. In general, however, because of the projection matrix A, AG_ϕ is not triangular and there is no special delay property exhibitted.

The sequence $\boldsymbol{v}$ is the vector innovations of $\boldsymbol{x}'$ or equivalently of $\boldsymbol{u}$. This relationship interestingly follows from the monic-triangular nature of G_ϕ so that

$$|R_{\boldsymbol{uu}}| = |G_\phi| \cdot |R_{\boldsymbol{vv}}| \cdot |G_\phi| = |R_{\boldsymbol{vv}}| \quad . \tag{5.242}$$

In this case, the product of the innovations input energies is thus equal to the product of the eigenvalues of $R_{\boldsymbol{uu}}$. Even though the innovations are not the same as the modal inputs of VC and the decision-based detector is not optimum, they carry the same information with the same SNR in the absence of any decision errors. The set of parallel subchannels is also different, even though the overall equivalent AWGN has the same SNR. The diagram in Figure 5.29 illustrates the equivalent of the linear prediction relationships for the MMSE-DFE. Figure 5.29 is intended to be similar to Figure 5.1. G_ϕ replaces $G_x(D)$ in the finite-length symbol case. Only a GDFE with diagonal-autocorrelation input has the canonical SNR performance.

[17] Note the upper triangular matrix correct appears first here, unlike the factorization of R_b^{-1} where the upper triangular matrix appeared last. In Matlab, use chol($JR_{\boldsymbol{uu}}J$) command where J has all ones down the antidiagonal and zeros elsewhere, and **pre- and post-**multiply the resultant matrix by J, and then take congugate 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. $R_{\boldsymbol{vv}}$ is diag(diag(Gxbar))*diag(diag(Gxbar).

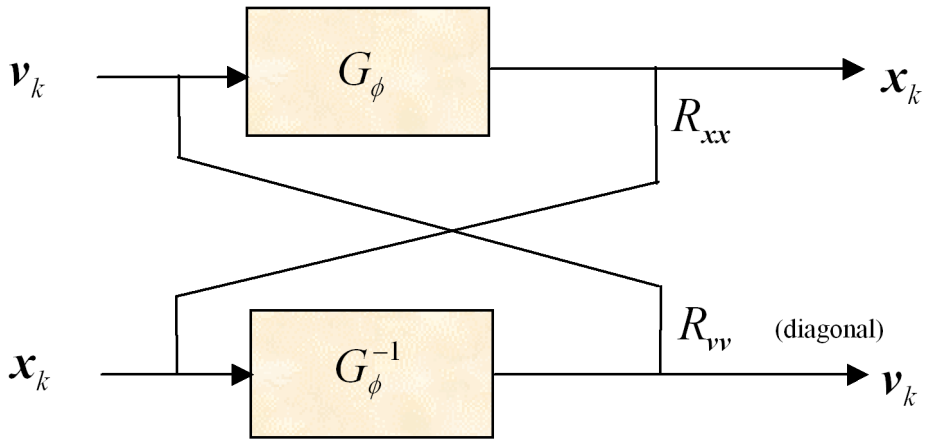


Figure 5.29: Innovations and input.

5.4.3 The Circulant DFE (CDFE)

The $N \times (N + \nu)$ convolution matrix P can never have rank greater than N. Also, the square $N \times N$ matrix P that arises from the use of a cyclic prefix (the same cyclic prefix used for DMT) always has rank $\tilde{N} \leq N$. The case $N^* = \tilde{N} = N$ would occur when the input autocorrelation matrix $R_{\boldsymbol{xx}}$ had been selected to correspond to a sampling rate where all subchannels are used in loading. The situation of $\tilde{N} < N$ occurs when the channel has notches (zero gain) at the exact frequencies that would be considered "carrier" frequencies in DMT. Instead of eliminating these singularities, the wise input design would always zero energy on any such notch frequencies, which will lead to multiple discontiguous bands as is further discussed in Section 5.5.

In the simpler case of a non-singular input,

$$R_{\boldsymbol{uu}} = R_{\boldsymbol{xx}} \quad . \tag{5.243}$$

The input factorization then trivially becomes Cholesky factorization and

$$\boldsymbol{x} = G_x \boldsymbol{v} \tag{5.244}$$

with G_x a upper-triangular $N \times N$ matrix. This situation corresponds to the **circulant DFE** or **CDFE**. In this case, for a stationary channel, namely a circulant Toeplitz P and finite fixed ν as $N \to \infty$, the CDFE will converge to a MMSE-DFE as in Section 5.7. For the more complex case of input singularity corresponding to input design zeroing certain frequencies for any reason (channel notches or choice of loading algorithm, or both), then the situation essentially leads to multiple CDFEs each converging to its own corresponding MMSE-DFE.[18]

The transmit filter will correspond to any of the middle rows of G_x while the feedback section is any of the middle rows of G.[19] The matched filter matrix clearly converges to a matched filter, and any of the middle rows of this feedforward matrix converges to the anti-causal feedback section. The CDFE as described here exists only when $R_{\boldsymbol{xx}}$ is nonsingular, which may not occur especially if water-filling is used. The next section (5.5) deals with an adjustment that needs to be made when there are 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, obtains the same SNR, but is only canonical because it does not have an ML detector.

[18] One can think of applying everything in this section to each CDFE individually in this case of multiple discontiguous band use.

[19] All these middle rows will have the nonzero coefficient values tending to the same fixed filter when the block symbol length is long enough.

EXAMPLE 5.4.1 ($1 + .9D^{-1}$ CDFE with PAM input) Returing the familiar $1+.9D^{-1}$ example often used in this text, the AWGN noise variance is $\frac{\mathcal{N}_0}{2} = .181$ and $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$. This example will use flat energy transmission with cyclic prefix of length $\nu = 1$ on a block size of $N = 8$. Thus, $R_{\boldsymbol{vv}} = R_{\boldsymbol{uu}} = I$. For this case, $C = I$ for a circulant matrix and $A = I$ also because the input is white. Rather than repeat the analysis of previous developments mathematically here, we simply insert some descriptive matlab text with results:

```
>> P=toeplitz([.9 zeros(1,7)]',[.9 1 zeros(1,6)]);
>> P(8,1)=1
P = 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
>> Rf=(1/.181)*P'*P
Rf = 10.0000    4.9724         0         0         0         0         0    4.9724
      4.9724   10.0000    4.9724         0         0         0         0         0
           0    4.9724   10.0000    4.9724         0         0         0         0
           0         0    4.9724   10.0000    4.9724         0         0         0
           0         0         0    4.9724   10.0000    4.9724         0         0
           0         0         0         0    4.9724   10.0000    4.9724         0
           0         0         0         0         0    4.9724   10.0000    4.9724
      4.9724         0         0         0         0         0    4.9724   10.0000
>> Rbinv= Rf + eye(8)
Rbinv = 11.0000    4.9724         0         0         0         0         0    4.9724
         4.9724   11.0000    4.9724         0         0         0         0         0
              0    4.9724   11.0000    4.9724         0         0         0         0
              0         0    4.9724   11.0000    4.9724         0         0         0
              0         0         0    4.9724   11.0000    4.9724         0         0
              0         0         0         0    4.9724   11.0000    4.9724         0
              0         0         0         0         0    4.9724   11.0000    4.9724
         4.9724         0         0         0         0         0    4.9724   11.0000
>> Gbar = chol(Rbinv)
Gbar = 3.3166    1.4992         0         0         0         0         0    1.4992
              0    2.9584    1.6807         0         0         0         0   -0.7598
              0         0    2.8592    1.7391         0         0         0    0.4466
              0         0         0    2.8241    1.7607         0         0   -0.2750
              0         0         0         0    2.8107    1.7691         0    0.1723
              0         0         0         0         0    2.8054    1.7724   -0.1086
              0         0         0         0         0         0    2.8033    1.8424
              0         0         0         0         0         0         0    2.1128
>> G=inv(diag(diag(Gbar)))*Gbar
G = 1.0000    0.4520         0         0         0         0         0    0.4520
         0    1.0000    0.5681         0         0         0         0   -0.2568
         0         0    1.0000    0.6082         0         0         0    0.1562
         0         0         0    1.0000    0.6234         0         0   -0.0974
         0         0         0         0    1.0000    0.6294         0    0.0613
         0         0         0         0         0    1.0000    0.6318   -0.0387
         0         0         0         0         0         0    1.0000    0.6572
         0         0         0         0         0         0         0    1.0000
>> S0=diag(diag(Gbar))*diag(diag(Gbar))
```

```
S0 = 11.0000         0         0         0         0         0         0         0
          0    8.7523         0         0         0         0         0         0
          0         0    8.1751         0         0         0         0         0
          0         0         0    7.9756         0         0         0         0
          0         0         0         0    7.9000         0         0         0
          0         0         0         0         0    7.8703         0         0
          0         0         0         0         0         0    7.8585         0
          0         0         0         0         0         0         0    4.4639

>> 10*log10(det(S0)^(1/9)-1) = 7.1666 dB
>>W=(1/.181)*inv(S0)*inv(G')*P'
W = 0.4520         0         0         0         0         0         0    0.5023
    0.3744    0.5681         0         0         0         0         0   -0.2853
   -0.2277    0.3303    0.6082         0         0         0         0    0.1736
    0.1420   -0.2059    0.3135    0.6234         0         0         0   -0.1082
   -0.0894    0.1296   -0.1973    0.3069    0.6294         0         0    0.0681
    0.0565   -0.0819    0.1247   -0.1939    0.3043    0.6318         0   -0.0430
   -0.0357    0.0518   -0.0789    0.1227   -0.1926    0.3033    0.6327    0.0272
   -0.1702    0.0762    0.0017   -0.0800    0.1753   -0.3078    0.5056    0.3005
```

The important G matrix clearly is converging to what is known (from Chapter 3) to be the MMSE-DFE feedback section of $1 + .633D$ in the middle rows of G. 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 blocking is necessary if the transmitted symbol sequence is just considered 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 the more realistic situation illustrated in this example is that transmission begins at one point in time and continues. In that latter more realistic case, the first few samples are better handled by the CDFE than by the usual MMSE-DFE design. Indeed the 7.16 dB SNR here increases to 7.75 dB at $N = 16$, 8.26 dB at $N = 100$ and 8.3 dB at $N = 200$ illustrated the fastest or best convergence to the eventual 8.4 dB. Thus, one notes that use of a stationary-designed MMSE-DFE over finite block lengths (as often occurs in wireless transmission, for instance GSM (Group Special Mobile or widely used cell-phone transmission method) where block lengths are only about 150 symbols long with essentially QPSK and a MMSE-DFE equalizer) will not achieve the performance computed for the asymmptotic convergence. Some GSM designers are aware of the effect and actually use time-varying DFE's within the packet of samples.

The feedforward filter is converging also, but clearly has significant terms for all samples at least for $N = 8$. Note the "end" sample here is related to $\nu = 1$ and eventually just implies an implementation with $\Delta > 0$ to handle the maximum-phase channel.

It is interesting to compare the CDFE with a conventional GDFE as in the design of Example 5.3.3 execept that N is increased from 2 there to 8 here to produce the results with now energy 9/8 fed into each of the dimensions recognizing the channel singularity on one dimension and not inserting energy there:

```
>> P=(1/sqrt(.181))*toeplitz([.9 zeros(1,7)]',[.9 1 zeros(1,7)])
P = 2.1155    2.3505         0         0         0         0         0         0         0
         0    2.1155    2.3505         0         0         0         0         0         0
         0         0    2.1155    2.3505         0         0         0         0         0
         0         0         0    2.1155    2.3505         0         0         0         0
         0         0         0         0    2.1155    2.3505         0         0         0
```

```
         0         0         0         0         0    2.1155    2.3505         0         0
         0         0         0         0         0         0    2.1155    2.3505         0
         0         0         0         0         0         0         0    2.1155    2.3505
>> M
M =
    0.0775   -0.1527    0.2232    0.2868   -0.3414    0.3852   -0.4153   -0.4214    0.4728
    0.2319   -0.4037    0.4712    0.4182   -0.2608    0.0428    0.1748    0.3238   -0.4255
    0.3583   -0.4657    0.2480   -0.1415    0.4320   -0.4280    0.1475   -0.1871    0.3830
    0.4415   -0.3099   -0.2232   -0.4674    0.1108    0.3852   -0.4008    0.0278   -0.3447
    0.4714   -0.0090   -0.4712   -0.0208   -0.4705    0.0428    0.4666    0.1348    0.3102
    0.4445    0.2960   -0.2480    0.4602    0.0526   -0.4280   -0.3140   -0.2812   -0.2792
    0.3639    0.4626    0.2232    0.1806    0.4522    0.3852    0.0146    0.3936    0.2513
    0.2395    0.4127    0.4712   -0.3975   -0.2097    0.0428    0.2917   -0.4586   -0.2261
    0.0862    0.1697    0.2480   -0.3187   -0.3794   -0.4280   -0.4615    0.4683    0.2035
>> Mt
Mt =
    0.0775   -0.1527    0.2232    0.2868   -0.3414    0.3852   -0.4153   -0.4214
    0.2319   -0.4037    0.4712    0.4182   -0.2608    0.0428    0.1748    0.3238
    0.3583   -0.4657    0.2480   -0.1415    0.4320   -0.4280    0.1475   -0.1871
    0.4415   -0.3099   -0.2232   -0.4674    0.1108    0.3852   -0.4008    0.0278
    0.4714   -0.0090   -0.4712   -0.0208   -0.4705    0.0428    0.4666    0.1348
    0.4445    0.2960   -0.2480    0.4602    0.0526   -0.4280   -0.3140   -0.2812
    0.3639    0.4626    0.2232    0.1806    0.4522    0.3852    0.0146    0.3936
    0.2395    0.4127    0.4712   -0.3975   -0.2097    0.0428    0.2917   -0.4586
    0.0862    0.1697    0.2480   -0.3187   -0.3794   -0.4280   -0.4615    0.4683
>> Pm=Mt*Mt'
Pm =
    0.7764    0.2012   -0.1811    0.1630   -0.1467    0.1320   -0.1188    0.1069   -0.0962
    0.2012    0.8189    0.1630   -0.1467    0.1320   -0.1188    0.1069   -0.0962    0.0866
   -0.1811    0.1630    0.8533    0.1320   -0.1188    0.1069   -0.0962    0.0866   -0.0779
    0.1630   -0.1467    0.1320    0.8812    0.1069   -0.0962    0.0866   -0.0779    0.0702
   -0.1467    0.1320   -0.1188    0.1069    0.9038    0.0866   -0.0779    0.0702   -0.0631
    0.1320   -0.1188    0.1069   -0.0962    0.0866    0.9221    0.0702   -0.0631    0.0568
   -0.1188    0.1069   -0.0962    0.0866   -0.0779    0.0702    0.9369    0.0568   -0.0511
    0.1069   -0.0962    0.0866   -0.0779    0.0702   -0.0631    0.0568    0.9489    0.0460
   -0.0962    0.0866   -0.0779    0.0702   -0.0631    0.0568   -0.0511    0.0460    0.9586

>> rxx=(9/8)*Pm
rxx =
    0.8735    0.2263   -0.2037    0.1833   -0.1650    0.1485   -0.1337    0.1203   -0.1083
    0.2263    0.9213    0.1833   -0.1650    0.1485   -0.1337    0.1203   -0.1083    0.0974
   -0.2037    0.1833    0.9600    0.1485   -0.1337    0.1203   -0.1083    0.0974   -0.0877
    0.1833   -0.1650    0.1485    0.9913    0.1203   -0.1083    0.0974   -0.0877    0.0789
   -0.1650    0.1485   -0.1337    0.1203    1.0167    0.0974   -0.0877    0.0789   -0.0710
    0.1485   -0.1337    0.1203   -0.1083    0.0974    1.0373    0.0789   -0.0710    0.0639
   -0.1337    0.1203   -0.1083    0.0974   -0.0877    0.0789    1.0540    0.0639   -0.0575
    0.1203   -0.1083    0.0974   -0.0877    0.0789   -0.0710    0.0639    1.0675    0.0518
   -0.1083    0.0974   -0.0877    0.0789   -0.0710    0.0639   -0.0575    0.0518    1.0784
>> A*ruu*A'  (checks)
ans =
    0.8736    0.2264   -0.2037    0.1833   -0.1650    0.1485   -0.1336    0.1203   -0.1083
    0.2264    0.9212    0.1833   -0.1650    0.1485   -0.1336    0.1203   -0.1082    0.0974
   -0.2037    0.1833    0.9601    0.1485   -0.1337    0.1203   -0.1083    0.0975   -0.0877
    0.1833   -0.1650    0.1485    0.9913    0.1203   -0.1083    0.0974   -0.0876    0.0790
```

```
   -0.1650    0.1485   -0.1337    0.1203    1.0167    0.0974   -0.0877    0.0790   -0.0710
    0.1485   -0.1336    0.1203   -0.1083    0.0974    1.0373    0.0790   -0.0711    0.0639
   -0.1336    0.1203   -0.1083    0.0974   -0.0877    0.0790    1.0540    0.0639   -0.0574
    0.1203   -0.1082    0.0975   -0.0876    0.0790   -0.0711    0.0639    1.0674    0.0517
   -0.1083    0.0974   -0.0877    0.0790   -0.0710    0.0639   -0.0574    0.0517    1.0783
>> A  (caution: different Matlab versions produce different eigenvectors)
A =
   -0.3794    0.0519   -0.5466    0.2553   -0.3330   -0.0143   -0.3291   -0.2157
    0.2803   -0.0590   -0.2591    0.7194    0.1414   -0.0333    0.1254   -0.3396
   -0.0047   -0.0481   -0.0369    0.0147   -0.1217   -0.3389    0.8394   -0.1235
   -0.0285   -0.0068   -0.4208   -0.5844   -0.0748    0.2734    0.1743   -0.5010
    0.1331    0.0490    0.5621    0.0263    0.0256    0.1161   -0.1697   -0.7239
   -0.7324    0.0778    0.1464    0.0011    0.3665   -0.4337   -0.0155   -0.1883
   -0.1253    0.4857   -0.1207    0.1086    0.5678    0.5325    0.2147    0.0829
   -0.4401   -0.1120    0.3135    0.2511   -0.4834    0.5255    0.2567    0.0741
   -0.1141   -0.8570   -0.0738         0    0.3972    0.2187         0         0
>> ruu
ruu =
    1.1250         0         0         0         0         0         0         0
         0    1.1250         0         0         0         0         0         0
         0         0    1.1250         0         0         0         0         0
         0         0         0    1.1250         0         0         0         0
         0         0         0         0    1.1250         0         0         0
         0         0         0         0         0    1.1250         0         0
         0         0         0         0         0         0    1.1250         0
         0         0         0         0         0         0         0    1.1250
>> Pt=P*A
Pt =
   -0.1438   -0.0289   -1.7653    2.2310   -0.3721   -0.1085   -0.4014   -1.2545
    0.5819   -0.2379   -0.6348    1.5564    0.0131   -0.8670    2.2383   -1.0087
   -0.0769   -0.1177   -1.0672   -1.3425   -0.4333   -0.0743    2.1854   -1.4389
    0.2526    0.1008    0.4310   -1.1745   -0.0981    0.8513   -0.0302   -2.7614
   -1.4399    0.2865    1.5332    0.0582    0.9156   -0.7738   -0.3954   -1.9740
   -1.8439    1.3062    0.0260    0.2576    2.1099    0.3342    0.4719   -0.2035
   -1.2995    0.7642    0.4815    0.8199    0.0649    2.3617    1.0576    0.3495
   -1.1992   -2.2513    0.4897    0.5312   -0.0890    1.6257    0.5430    0.1568
>> Rf=Pt'*Pt
Rf =
    9.0291   -1.2142   -3.3934   -1.8697   -5.1169   -4.7888   -1.1417    1.5820
   -1.2142    7.5222    0.1098   -0.6111    3.3171   -1.3365   -0.6924   -0.7499
   -3.3934    0.1098    7.6671   -3.2492    2.5150    1.9439   -2.8762    0.4135
   -1.8697   -0.6111   -3.2492   11.6059    0.4898    0.3495    0.9436    1.0085
   -5.1169    3.3171    2.5150    0.4898    5.6382   -0.0170   -0.1114   -0.8802
   -4.7888   -1.3365    1.9439    0.3495   -0.0170   10.4246    1.7590    1.3068
   -1.1417   -0.6924   -2.8762    0.9436   -0.1114    1.7590   11.7403   -3.6760
    1.5820   -0.7499    0.4135    1.0085   -0.8802    1.3068   -3.6760   16.3715
>> Rbinv=Rf+inv(ruu)
Rbinv =
    9.9180   -1.2142   -3.3934   -1.8697   -5.1169   -4.7888   -1.1417    1.5820
   -1.2142    8.4111    0.1098   -0.6111    3.3171   -1.3365   -0.6924   -0.7499
   -3.3934    0.1098    8.5560   -3.2492    2.5150    1.9439   -2.8762    0.4135
   -1.8697   -0.6111   -3.2492   12.4947    0.4898    0.3495    0.9436    1.0085
   -5.1169    3.3171    2.5150    0.4898    6.5271   -0.0170   -0.1114   -0.8802
   -4.7888   -1.3365    1.9439    0.3495   -0.0170   11.3135    1.7590    1.3068
```

```
   -1.1417   -0.6924   -2.8762    0.9436   -0.1114    1.7590   12.6292   -3.6760
    1.5820   -0.7499    0.4135    1.0085   -0.8802    1.3068   -3.6760   17.2604
>> Gbar=chol(Rbinv)
Gbar =
    3.1493   -0.3855   -1.0775   -0.5937   -1.6248   -1.5206   -0.3625    0.5023
         0    2.8744   -0.1063   -0.2922    0.9361   -0.6689   -0.2895   -0.1935
         0         0    2.7173   -1.4426    0.3179    0.0862   -1.2136    0.3438
         0         0         0    3.1584    0.0815   -0.1977   -0.3505    0.5529
         0         0         0         0    1.7039   -1.0991   -0.0089   -0.0218
         0         0         0         0         0    2.7017    0.3848    0.7391
         0         0         0         0         0         0    3.2665   -0.9868
         0         0         0         0         0         0         0    3.8764
>> G=inv(diag(diag(Gbar)))*Gbar
G =
    1.0000   -0.1224   -0.3421   -0.1885   -0.5159   -0.4828   -0.1151    0.1595
         0    1.0000   -0.0370   -0.1017    0.3257   -0.2327   -0.1007   -0.0673
         0         0    1.0000   -0.5309    0.1170    0.0317   -0.4466    0.1265
         0         0         0    1.0000    0.0258   -0.0626   -0.1110    0.1750
         0         0         0         0    1.0000   -0.6450   -0.0052   -0.0128
         0         0         0         0         0    1.0000    0.1424    0.2736
         0         0         0         0         0         0    1.0000   -0.3021
         0         0         0         0         0         0         0    1.0000
>> S0=diag(diag(Gbar))*diag(diag(Gbar))
S0 =
    9.9180         0         0         0         0         0         0         0
         0    8.2624         0         0         0         0         0         0
         0         0    7.3837         0         0         0         0         0
         0         0         0    9.9758         0         0         0         0
         0         0         0         0    2.9033         0         0         0
         0         0         0         0         0    7.2993         0         0
         0         0         0         0         0         0   10.6702         0
         0         0         0         0         0         0         0   15.0262
>> snr=10*log10(det(ruu*S0)^(1/9)-1)
snr = 7.9306
```

The filters are non-stationary in this case and will not converge, but the performance is better (because no energy is wasted in a cyclic prefix or in a null space to the channel) and the SNR is 8 dB. Vector Coding with flat-energy transmission would achieve this same performance level. This nonstationary triangular GDFE will more rapidly converge 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 the 8.8 dB maximum that occurs with water-filling.

$$8.4 \text{ dB} < \lim_{N \to \infty} \text{SNR}_{N+\nu/N,GDFE,u} < 8.8 \text{ dB} \quad . \tag{5.245}$$

5.4.4 Vector-Side DMT (VS-DMT) as a special case of the GDFE

Vector-Side DMT is an interesting and increasingly practical transmission method often used when several transmission channels can be coordinated at a receiving side, but the transmitters corresponding to the different symbol dimensions are physically separated. Such a situation can arise in both wireless (when several physically separate portable devices transmit to a base-station antenna or array of such base-station antennae) and wireline (when several phone lines at different remote sites all converge on a common "central office" point and all transmit up to that point through a binder of wires that may have considerable crosstalk between the lines).

The channel output can be written as

$$\begin{aligned} \boldsymbol{y}_{small} &= P_1\boldsymbol{x}_1 + P_2\boldsymbol{x}_2 + ... + P_L\boldsymbol{x}_L + \boldsymbol{n}_{small} && (5.246)\\ &= [P_1\ P_2\ ...\ P_L]\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\\ \vdots\\ \boldsymbol{x}_L\end{bmatrix} + \boldsymbol{n}_{small} && (5.247)\\ &= P\boldsymbol{x} + \boldsymbol{n}_{small} \quad . && (5.248)\end{aligned}$$

Each of the matrices P_i, $i = 1, ..., L$ is cyclic because a DMT system with a cyclic prefix and the same clock is used in all systems (for instance with Zippering of Section 4.6). Thus a single DFT in the receiver then diagonalizes each of the individual contributions. A GDFE can be directly applied in this case, but an ML detector will need to be substituted for the "sbs" device and a joint decision on the L components added together from the different input locations would need to be made. Of more practical interest is the case where the receiver also has L independent observations (i.e., L lines or L receiver antennae, each with channel matrix $P(i)$) and thus

$$\boldsymbol{y} = \begin{bmatrix}\boldsymbol{y}_1\\ \boldsymbol{y}_2\\ \vdots\\ \boldsymbol{y}_L\end{bmatrix} = \begin{bmatrix}P_{1,1} & P_{1,2} & ... & P_{1,L}\\ P_{2,1} & P_{2,2} & ... & P_{2,L}\\ \vdots & \vdots & \ddots & \vdots\\ P_{L,1} & P_{L,2} & ... & P_{L,L}\end{bmatrix}\begin{bmatrix}\boldsymbol{x}_1\\ \boldsymbol{x}_2\\ \vdots\\ \boldsymbol{x}_L\end{bmatrix} + \boldsymbol{n} \quad . \tag{5.249}$$

After a receiver DFT on each output $\boldsymbol{y}_i$ to generate $\boldsymbol{Y}_i$, there will be a sum of contributions of all the inputs $\boldsymbol{X}_i$ into each and every $\boldsymbol{Y}_i$. These contributions, because of the DMT structure however will be independent for each tone, meaning

$$Y_{n,i} = \sum_{j=1}^{L} P_{n,i,j}X_{n,i} + N_{n,i} \quad , \tag{5.250}$$

where $P_{n,i,j}$ is the n^{th}-tone DFT value for the channel response from input j to output i (that is the DFT of the cyclic matrix $P_{i,j}$'s first row. By creating a new matrix channel for each tone n,

$$\boldsymbol{Y}_n = \begin{bmatrix}Y_{n,1}\\ Y_{n,2}\\ \vdots\\ Y_{n,L}\end{bmatrix} = \begin{bmatrix}P_{n,1,1} & P_{n,2} & ... & P_{n,L}\\ Pn,2,1 & P_{n,2,2} & ... & P_{n,2,L}\\ \vdots & \vdots & \ddots & \vdots\\ P_{n,L,1} & P_{n,L,2} & ... & P_{n,L,L}\end{bmatrix}\begin{bmatrix}X_{n,1}\\ X_{n,2}\\ \vdots\\ X_{n,L}\end{bmatrix} + \boldsymbol{N}_n \quad . \tag{5.251}$$

or the $L \times L$ matrix channel

$$\boldsymbol{Y}_n = \mathcal{P}_n\boldsymbol{X}_n + \boldsymbol{N}_n \quad , \tag{5.252}$$

where $\mathcal{P}_n$ is used instead of the usual P to prevent confusion with the original time-domain cyclic matrices P_i. $\mathcal{P}_n$ is not in general cyclic, nor nonsingular, but the general GDFE theory can be applied to it, independently for each n. Singularity can often be avoided by design (don't include input antennae or lines in output model that don't get to that output). Thus, Vector-Side DMT is a set of up to N independent GDFE's, one for each tone of the common-clock DMT transmission systems. The complexity of each tone's GDFE will be L^2 generally, while the FFT's used are $Nlog_2(N)$ for each user. Thus total complexity per line or user is the order of $\left(N \cdot L^2 + L \cdot N\log_2(N)\right)/L = N \cdot L + N\log_2(N)$, which remains low for up to say $log_2(N)$ users or less.

Of course, for each tone, the designer has the choice of using either $G = I$ (vector coding) or $G \neq I$ triangular GDFE. The latter is preferred because vector coding requires coordination (i.e., the vector matrix M_n) on the transmit side, which may not be physically possible. In the GDFE with the square $\mathcal{P}_n$, a similar problem that might be encountered with the matrix A_n. This problem can be avoided by judiciously exciting all N input dimensions on at least one of the L transmitters, insuring effectively that $A_n = I$ and can be ingored in implementation (eliminating then transmitter side coordination).

The orginal Vector-Side DMT with vector-coding was first introduced in a 1994 version of this class and caught some attention of students who later called it Vector DMT or when inputs are equally loaded the more common "Vector OFDM" in wireless, and pushed it into the IEEE802.16 standard for wireless transmission. However, Vector-Side DMT with a triangular GDFE is more practical and preferred because it performs the same, has same complexity, and does not require any physical co-location of the transmitters, and places all complexity of the GDFE at the base-station side where more power is usually available. This latter approach is preferred in both future wireless efforts (perhaps 802.11wng project) and vectored-DSLAM-DSL efforts.

Ginis Precoder

The dual situation to the physically separated transmitters is the opposite direction of transmission in any of the above examples. Then, receiver coordination is not possible. The ZF-GDFE as in Section 5.3.7 with precoding is of interest. In this case, the Vector-Side-DMT analysis above is used and the channel model has identical structure and can be reduced to a set of independent tone-indexed subchannels with Vector-Side DMT. In this case, the ZF-GDFE with transmit precoder can be used. George Ginis, a former student and TA of this class, noted that for wireline DSL applications there is essentially zero performance loss between the ZF-GDFE and the GDFE generally (because of certain transmission characteristics of crosstalk and telephone lines). Thus the precoder of Section 5.3.7 can be applied directly to each and every tone-indexed $L \times L$ subchannel in Vector-Side DMT without performance loss (other than precoding loss, which Ginis and Yu also found a way to make much smaller by exploiting the L-dimensional space over which precoding could be implemented). This is known as a **Ginis Precoder**.

5.5 Transmit Optimization for the GDFE

The GDFE depends upon an input covariance $R_{\boldsymbol{xx}}$, which can be any autocorrelation matrix. There is one $R_{\boldsymbol{xx}}$ that maximizes the SNR_{gdfe} and is thus considered to be optimum for packet transmission. This section revisits determination of this $R_{\boldsymbol{xx}}$ through finite-length symbol partitioning and subsequent water-fill loading. Subsection 5.5.1 returns to vector coding to compute the best $R_{\boldsymbol{xx}}$ while also revisiting briefly the set of parallel channels thus derived. Subsection 5.5.2 then visits the construction of this best $R_{\boldsymbol{xx}}$ using the triangular white input of the GDFE, requiring a generalization of Cholesky factorization. Subsection 5.5.3 then specificially investigates loading for the GDFE and the inherent interpolation in the transmit signal construction.

The Circulant DFE of Subsection 5.4.3 has the same performance as DMT when both have the same P, $R_{\boldsymbol{nn}}$, and $R_{\boldsymbol{xx}}$. The CDFE can also converge to a set of the MMSE-DFEs in Chapter 3. Section 5.5.4 studies the CDFE and uses a DMT system to optimize the performance of a CDFE by two methods – digital interpolation of sampling rate and analog interpolation of sampling rate, 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 ν. The optimization of symbol rate and carrier frequency is equivalent to the interpolation in the generalized-Cholesky and CDFE of Subsection 5.5.3.

5.5.1 The channel-dependent optimized input

Section 5.4.1 showed that VC is a special case of the GDFE 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. The forward canonical channel in the VC case is characterized by $R_f = \tilde{P}^*\tilde{P} = M^*M\Lambda^*FF^*\Lambda M^*M = |\Lambda|^2$. The forward canonical channel model becomes

$$\boldsymbol{z} = R_f\boldsymbol{u} + \boldsymbol{n}' = |\Lambda|^2\boldsymbol{u} + \boldsymbol{n}' = \text{diag}(|\lambda_n|^2 \cdot u_n + n'_n) \quad , \tag{5.253}$$

where $\boldsymbol{n}'$ has autocorrelation matrix $R_{\boldsymbol{n}'\boldsymbol{n}'} = |\Lambda|^2$. The SNR of each subchannel is $\text{SNR}_n = \mathcal{E}_n|\lambda_n|^2$.[20] The backward channel has

$$R_b^{-1} = S_0 = R_f + S_{uu}^{-1} = \text{diag}(|\lambda_n|^2 + \mathcal{E}_n^{-1}) \quad , \tag{5.254}$$

or

$$R_b = R_{\boldsymbol{ee}} = \text{diag}(\frac{\mathcal{E}_n}{1+\text{SNR}_n}) \quad . \tag{5.255}$$

Equation (5.255) validates the relation concerning the (biased) SNR on each channel of the GDFE

$$\text{SNR}_{gdfe,n} = 1 + \text{SNR}_n \quad . \tag{5.256}$$

Bias removal on each subchannel then provides an unbiased SNR for the GDFE and backward-canonical channel (MMSE case) of

$$\text{SNR}_{gdfe,u,n} = \text{SNR}_{gdfe,n} - 1 = \text{SNR}_n \quad . \tag{5.257}$$

The unbiased SNR on each of the forward subchannels in (5.253) is also SNR_n meaning that the forward and backward channels, in the special case of Vector Coding ($A = M'$, N^* columns of M) are essentially the same. The best structure of VC thus exhibits a unique property within the set of all GDFE-like systems: 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 there is no ISI on any of the independent subchannels in this special case.

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 $\tilde{P}^*$, feedforward filtering by $\boldsymbol{S}_0^{-1}M^*$, and

[20] Recalling that λ_n includes the effect of the noise normalization in (5.232), the g_n of Chapter 4's loading discussion is $g_n = |\lambda_n|^2$ here.

removing bias on each subchannel:

$$\text{diag}\left(\frac{\text{SNR}_n + 1}{\text{SNR}_n}\right) S_0^{-1} M^* \tilde{P}^* \tag{5.258}$$

$$= \text{diag}\left(\frac{|\lambda_n|^2 + \mathcal{E}_n^{-1}}{|\lambda_n|^2}\right) \cdot \text{diag}\left(\frac{1}{|\lambda_n|^2 + \mathcal{E}_n^{-1}}\right) \cdot \Lambda^* F^* \tag{5.259}$$

$$= \text{diag}\left(\frac{1}{|\lambda_n|^2}\right) \cdot \text{diag}(\lambda_n^*) F^* \tag{5.260}$$

$$= \text{diag}\left(\frac{1}{\lambda_n}\right) F^* \tag{5.261}$$

$$= \Lambda^{-1} F^* \quad . \tag{5.262}$$

This unbiased feedforward section then simply completes the parallelization into subchannels and scales each subchannel by $|\lambda_n|^2$. The subchannels created can and usually do carry different amounts of information: For any $\{\mathcal{E}_n\} \rightarrow \{\text{SNR}_n\}$ distribution,

$$\bar{b}_n = \frac{1}{2} \log_2 \left(1 + \frac{\text{SNR}_n}{\Gamma}\right) \tag{5.263}$$

and $\bar{b}_n = \bar{I}_n$ when $\Gamma = 0$ dB. Exceptionally important is that in the DMT/VC case, the (unbiased) error on each subchannel contains no component of the signal in this special GDFE, as the Subsection 5.5.1 elaborates:

Non-zero Gap Inputs

The optimum input for any GDFE will have $\Gamma = 0$ dB. However, VC/DMT systems have an optimum decoder for any code of any gap $\Gamma > 0$ dB because each subchannel is still exactly an AWGN. When $G \neq I$, as in any other GDFE, then the error of the GDFE contains a (non-Gaussian) error component. In this case, the choice of GDFE will lead to different SNR's even though all are MMSE-based for the same input autocorrelation matrix $R_{\boldsymbol{xx}}$ and the same channel and noise. The SNR for any gap is

$$\text{SNR}_{gdfe} = \Gamma \cdot \left\{ \left[\prod_{i=1}^{N^*} \left(1 + \frac{\text{SNR}_n}{\Gamma}\right) \right]^{\frac{1}{N+\nu}} - 1 \right\} \quad . \tag{5.264}$$

Since different choices of the input realization lead to different SNR_n sets, a non-zero gap causes SNR_{gdfe} to vary with the particular set of SNR_n's. All GDFE's will have the same $\text{SNR}_{gdfe,u} = 2^{2\bar{I}(\boldsymbol{x}\,\boldsymbol{y})} - 1$ only when $\Gamma = 0$ dB.

> **Theorem 5.5.1 (VC/DMT Optimality at non-zero Gap)** *Vector coding (which has a special case DMT when cyclic prefixes are used) has the highest SNR amoung all GDFE's for the same channel and input when the gap is non-zero, $\Gamma > 0$ dB.*
>
> **proof:** Because there is no component of the error due to the signal when VC is used, division by the gap on each subchannel is a true measure of performance of that subchannel. The overall gap works just as well as does the gap formula on each of the subchannels for VC/DMT. However, with an SNR that has a MMSE component depending on the signal, then multiplication of this "distortion" or MSE by $\Gamma > 0$ increases the signal component of the MSE also (but that signal component did not increase since the transmit power was not increased. Thus, the overall SNR computed according to (5.264 will be too low on each subchannel, and indeed then the product of such terms is also lower than the exact $\text{SNR}_{gdfe,u} = \Gamma \cdot (2^{2\bar{I}(\boldsymbol{x}\,\boldsymbol{y})} - 1)$ of the VC/DMT system. **QED.**

The computed SNR of other GDFE's may thus actually under-estimate the performance of the GDFE system when $\Gamma > 0$ dB, and after all, the gap was an approximation anyway. However, because

the VC/DMT detector is maximum likelihood, and since the SNR accurately represents exactly what is happening on each subchannel, then no other detector could have better performance. Since the SNR_{gdfe} is directly related (via the gap) to a constant probability of symbol error associated with the gap for the number of bits transmitted on each subchannel, the overall probability of symbol error for the VC/DMT system is that same value. However, for other GDFE's, the error is not Gaussian for non-zero 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 in error and the SNR-based approximation underestimates performance for non-VC/DMT GDFE's, that performance cannot be as good as the VC/DMT anyway). Further, a larger gap means a larger deviation in computed SNR.

As an example, the $1 + .9D^{-1}$ channel DMT example of Section 4.6 provided for $N = 8$ produced for $\Gamma = 0$ dB an $\text{SNR}_{DMT} = 7.6$ dB exactly equal to the CDFE SNR for that same channel and input in Section 5.4. However, if those examples are reworked (which the student can do as an exercise) with $\Gamma = 8.8$ dB, the two SNR's are then

$$\text{SNR}_{dmt}(\Gamma = 8.8) = 9.5\text{db} > \text{SNR}_{cdfe,u}(\Gamma = 8.8) = 8.6\text{db} \quad , \tag{5.265}$$

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 at capacity-achieving codes (so performance of the non-zero DMT system would be 9.5-8.8 or 0.7 dB, compared with 7.6 dB when the gap is 0 dB).

5.5.2 Maximization over $R_{\boldsymbol{xx}}$ of GDFE SNR

This subsection continues study of the special case of Vector Coding $A = M'$. Optimization of the GDFE is equivalent to maximization of $\text{SNR}_{\text{gdfe}} = \text{SNR}_{\text{gdfe,u}}$. As always with the GDFE

$$\text{SNR}_{\text{gdfe,u}} = 2^{2\overline{I(\boldsymbol{x};\boldsymbol{y})}} - 1 \quad . \tag{5.266}$$

The problem of maximization of a set of parallel channels with mutual information $I(\boldsymbol{x};\boldsymbol{y})$, which is what the GDFE has created, is solved by water-filling. The solution is then

$$\mathcal{E}_n + \frac{1}{|\lambda_i|^2} = K \tag{5.267}$$

where K is a constant determined either the data rate (MA problem) or energy constraint (RA) problem. In the RA case,

$$\text{SNR}_{\text{gdfe,u}} = 2^{2\bar{C}} - 1 \quad , \tag{5.268}$$

where $\bar{C}$ is the capacity or maximum value for $\overline{I(\boldsymbol{x};\boldsymbol{y})}$ The number of bits on each subchannel is

$$\bar{b}_n = \bar{c}_n = \frac{1}{2}\log_2(1 + \text{SNR}_n) \quad . \tag{5.269}$$

The water-filling procedure determines N^*, the number of used input dimensions. Those dimensions with zero energy are eliminated from the GDFE in the input-singularity-elimination step of generation of the canonical forward and backward channels.

The optimum covariance is then

$$R_{\boldsymbol{xx}}(opt) = R_{\boldsymbol{x}'\boldsymbol{x}'}(opt) = \sum_{n=1}^{N^*} \mathcal{E}_n \cdot \boldsymbol{m}_n \boldsymbol{m}_n^* \quad , \tag{5.270}$$

assuming a reordering of inputs so that used dimensions are $n = 1, ..., N^*$.

5.5.3 Using the optimized $R_{\boldsymbol{xx}}$ in the Triangular GDFE

To optimize the triangular GDFE, the designer first constructs M by SVD of the channel matrix $R_{\boldsymbol{nn}}^{-1/2}P$, and then computes the water-filling subchannel energies, before finally computing the optimum $R_{\boldsymbol{xx}}$ as in (5.270). Other input energy distributions can also be used, and the procedure in this Section applies to any $R_{\boldsymbol{xx}}$ that may be singular by whatever construction used to design that $R_{\boldsymbol{xx}}$. The implementation of a non-VC triangular GDFE can admit recursive causal implementation of the transmit packet, i.e. from $\boldsymbol{v}$ to best $R_{\boldsymbol{xx}}$. Such an implementation may be of interest for several reasons, some of which are related to the desire for a single-sided GDFE's implementation as in Sections 5.3.6 and 5.3.7.

The objective is a relationship

$$\boldsymbol{x} = A\boldsymbol{v} \tag{5.271}$$

where $R_{\boldsymbol{vv}}$ is diagonal and A is generalized triangular as defined next:

Definition 5.5.1 (Generalized Triangular) *An $(N+\nu)\times N^*$ matrix A is generalized triangular if the lower N^* rows are upper triangular. The remaining rows can be arbitrary.*

A generalized triangular matrix essentially interpolates in a step-by-step causal-like manner from a set of N^* white-input samples v_n to a larger set of $N+\nu$ modulator-output samples.

Reordering samples for a singular covariance

The input samples can be reordered so that the first N^* indices correspond to the nonsingular information-bearing inputs $\boldsymbol{v}$. The permutation matrix $J_{i,j}$ differs from an indentity 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 $J_{i,j}\boldsymbol{x}$ switches the position of the i^{th} and j^{th} elements of the vector $\boldsymbol{x}$. For example, with a 4-dimensional input

$$J_{2,3} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{5.272}$$

and $J_{2,3}[1234]^* = [1324]^*$. $J_{i,j}$ is symmetric, unitary, and $J_{i,j}J_{i,j}^* = I$. Therefore, or trivially, $|J_{i,j}| = 1$. If $\boldsymbol{x}' = J_{i,j}\boldsymbol{x}$, then $R_{\boldsymbol{x}'\boldsymbol{x}'} = J_{i,j}R_{\boldsymbol{xx}}J_{i,j}$.

The submatrix $R_{\boldsymbol{xx}}(i)$, $i = 1, ..., N+\nu-1$ is found in the $(i+1)\times(i+1)$ lower-right-hand-corner of $R_{\boldsymbol{xx}}$. The following algorithm reorders the input sample index of $\boldsymbol{x}$ so that its first N^* dimensions hare non-singular, that is $|R_{\boldsymbol{xx}}(i)| > 0$ for $i = 0, ..., N^*-1$.

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

Upon completion, the input indices will have been reordered so that the first N^* correspond to the informative part of the input and the rest simply are interpolated samples used to obtain the desired covariance for the channel. That set of last $N+\nu-N^*$ samples is linearly dependent on the first N^* samples. The first N^* components are denoted $\boldsymbol{x}_v$ and the rest are $\boldsymbol{x}_{\bar{v}}$. The transmitter will need to redo the order of the samples in $\boldsymbol{x} = A\boldsymbol{v}$ after any processing to generate the correct $R_{\boldsymbol{xx}}$ for input to the channel, that is – the resulting order matrix J_A is absorbed into A for all further GDFE calculations, that is

$$A \leftarrow J_A^*A \quad . \tag{5.273}$$

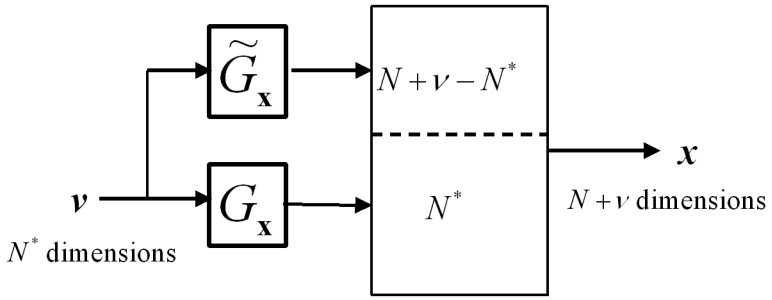


Figure 5.30: Illustration of inherent interpolation of Generalized Cholesky transmitter realization.

Generalized Cholesky Algorithm

The Generalized Cholesky Algorithm is now simple. Traditional Cholesky acts upon computed on the nonsingular reordered autocorrelation matrix

$$R_{\boldsymbol{x}\boldsymbol{x}}(N^*-1) = G_x R_{\boldsymbol{v}\boldsymbol{v}} G_x^* \quad , \tag{5.274}$$

which can be computed by conventional means.[21] The rightmost N^* positions in the upper $N+\nu-N^*$ rows of $R_{\boldsymbol{x}\boldsymbol{x}}$ are called $R_{\boldsymbol{x}_{\bar{v}}\boldsymbol{x}_v}$, which is the $(N+\nu-N^*)\times N^*$ cross-correlation matrix between $\boldsymbol{x}_v$ and $\boldsymbol{x}_{\bar{v}}$. then

$$A = \begin{bmatrix} R_{\boldsymbol{x}_{\bar{v}}\boldsymbol{x}_v} G_x^{-*} R_{\boldsymbol{v}\boldsymbol{v}}^{-1} \\ G_x \end{bmatrix} = \begin{bmatrix} \tilde{G}_x \\ G_x \end{bmatrix} \tag{5.275}$$

is generalized triangular, and $\boldsymbol{x} = A\boldsymbol{v}$ and also

$$R_{\boldsymbol{x}\boldsymbol{x}} = A R_{\boldsymbol{v}\boldsymbol{v}} A^* \quad . \tag{5.276}$$

The input can then be generated in a causal manner with the last $N+\nu-N^*$ samples being generated from only previous inputs (so strictly causal) as shown in Figure 5.5.1. Equivalently, one could extend $\boldsymbol{v}$ by an upper $N+\nu-N^*$ zeros and then extend A by zeros everywhere, except ones on the diagonal, to a square upper triangular monic $(N+\nu)\times(N+\nu)$ matrix A.

The following example illustrates the basic procedure.

EXAMPLE 5.5.1 (Generalized Cholesky and GDFE for $1+.9D^{-1}$ Channel.) The singular $R_{\boldsymbol{x}'\boldsymbol{x}'}$ for the pass space of the channel with white input was previously found to be (Example 5.4.1)

$$R_{\boldsymbol{x}'\boldsymbol{x}'} = \begin{bmatrix} .5945 & .3649 & -.3285 \\ .3649 & .6715 & .2956 \\ -.3285 & .2956 & .7340 \end{bmatrix} \quad . \tag{5.277}$$

This matrix will be considered now to have been provided as a desired input autocorrelation (perhaps the designer wanted transmit energy 2/3 per dimension, or a total of 2 for the symbol). Then Generalized Cholesky proceeds as follows with the Matlab instructions simply inserted for the reader convenience and understanding:

[21] The correct upper triangular matrix is not directly computed in Matlab. The Matlab command to compute this desired upper form is chol($JR_{\boldsymbol{u}\boldsymbol{u}}J$), 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 congugate transpose produces intermediate upper triangular matrix Gxbar. Finally, removal of diagonal terms by post multiplication generates Gx=Gxbar*inv(diag(diag(Gxbar))) or thus obtains the G_x used in this text. $R_{\boldsymbol{v}\boldsymbol{v}}$ is diag(diag(Gxbar))*diag(diag(Gxbar). For matlab command examples, see Examples 5.5.1, 5.5.2, and 5.5.3.

```
>> P=(1/sqrt(.181))*[.9 1 0
0 .9 1] =   2.1155    2.3505         0
                0    2.1155    2.3505
>> [F,L,M]=svd(P)

F=   -0.7071   -0.7071     L= 3.8694         0         0   M= -0.3866   -0.6671    0.6368
     -0.7071    0.7071             0    2.2422         0      -0.8161   -0.0741   -0.5731
                                                             -0.4295    0.7412    0.5158
>> Mt=M(1:3,1:2) = -0.3866   -0.6671
                   -0.8161   -0.0741
                   -0.4295    0.7412
>> rxx=Mt*Mt' =
    0.5945    0.3649   -0.3285
    0.3649    0.6715    0.2956
   -0.3285    0.2956    0.7340
>> Gxbar=(J*chol(J*rxx(2:3,2:3)*J)*J)' = 0.7433    0.3451
                                             0    0.8567
>> Gx=Gxbar*inv(diag(diag(Gxbar))) =     1.0000    0.4028
                                             0    1.0000
>> rvv=(diag(diag(Gxbar)))*(diag(diag(Gxbar))) =
    0.5525         0
         0    0.7340
>> Gx*rvv*Gx'  (check) = 0.6715    0.2956
                         0.2956    0.7340
>> A=[rxx(1,2:3)*inv(Gx')*inv(rvv)              % tildeGx
Gx] =
    0.9000   -0.4475
    1.0000    0.4028
         0    1.0000
>> A*rvv*A'  (another check)   0.5945    0.3649   -0.3285
                               0.3649    0.6715    0.2956
                              -0.3285    0.2956    0.7340
>> Pt=P*A = 4.2544    0.0000
            2.1155    3.2025
>> Rf=Pt'*Pt =
   22.5751    6.7748
    6.7748   10.2562
>> Rbinv=Rf+inv(rvv) =
   24.3851    6.7748
    6.7748   11.6187
>> Gbar=chol(Rbinv) = 4.9381    1.3719
                         0    3.1203
>> G=inv(diag(diag(Gbar)))*Gbar =    1.0000    0.2778
                                        0    1.0000
>> S0=diag(diag(Gbar))*diag(diag(Gbar))=
   24.3851         0
         0    9.7365
>> 10*log10(det(S0*rvv)^(1/3)-1) = 5.5427 dB
```

This is exactly the same value as obtained previously. The input matrix A is generalized triangular. By chance, the earlier Example 5.3.3 using this channel without generalized Cholesky happened to have a zero in the M matrix in just the right place that the resultant A then was also generalized triangular. This does not occur in general, and is just a result of the particular values in that case. Nonetheless, this example illustrates the Generalized

Cholesky procedure always ensures a generalized triangular "causal" input by construction for the GDFE.

5.5.4 CDFE with Causal Triangular input

The optimum spectrum with a CDFE can be constructed from water-filling in a DMT system with transmit IDFT matrix $\mathcal{Q}^*$ and energy distribution $\mathcal{E}_n$ on each of the tones. When the sampling rate has been adjusted so that all tones are used, then this input will be nonsingular. The designer may then proceed with direct application of the triangular GDFE of Section 5.5.4 with such a nonsingular input. Example 5.5.3 later in this section takes this approach. The resampling would of course have to be executed independently for each discontiguous band of frequency – a design and implementation nightmare. Another approach is to use a digital-interpolation technique that separates the A-matrix "modulation" into a carrier-frequency-and-interpolation part (which is never non-causal) and a "causal" part. This method finds the independent bands of the CDEF by essentially extracting those same bands from a DMT design of the same block-length. That second digital-interpolation approach appears first here in this subsection.

A good $N \times N$ $R_{\boldsymbol{xx}}$ is constructed for a cyclic channel P from a DMT system,[22]

$$R_{\boldsymbol{xx}} = \mathcal{Q}_{\bar{N}}^* R_{\boldsymbol{XX}} \mathcal{Q}_{\bar{N}} \quad , \tag{5.278}$$

where $R_{\boldsymbol{XX}}$ is an $N \times N$ diagonal matrix containing the DMT or OFDM input energies on the diagonal. The matrix $\mathcal{Q}_{\bar{N}}$ is an $\bar{N} \times \bar{N}$ FFT matrix. In the case of real-baseband DMT, then (5.278) has $\bar{N}$ replaced by $N = 2\bar{N}$ but the diagonal matrix $R_{\boldsymbol{XX}}$ is such that $R_{\boldsymbol{XX}}(i,i) = R_{\boldsymbol{XX}}(N-i, N-i)$ to ensure real time-domain outputs. These vectors $\boldsymbol{X}$ and $\boldsymbol{x}$ will be respectively ordered in frequency or time from smallest index 0 at the bottom to highest index $\bar{N} = \tilde{\bar{N}}$ at the top. Because of the DMT system used in design and ordering of the frequency index, it will be clear there are $M \geq 1$ bands where there are ALL nonzero entries within each band for $R_{\boldsymbol{XX}}$. In fact, each of these bands has $\bar{N}_i$ $i = 1, ..., M$ nonzero input energies and[23]

$$\bar{N}^* = \sum_{i=1}^{M} \bar{N}_i \quad . \tag{5.279}$$

The information-bearing part of the input contains non-zero tone inputs $\tilde{\boldsymbol{X}}$, which are interpolated from N^* used tones to $\bar{N}$ total tones according to

$$\boldsymbol{X} = J_g \tilde{\boldsymbol{X}} \quad . \tag{5.280}$$

For complex baseband $\boldsymbol{x}$, J_g is an $\bar{N} \times \bar{N}^*$ matrix 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

$$J_g = \begin{bmatrix} 0_{\bar{N}_{z,M+1} \times \bar{N}_M} & 0_{\bar{N}_{z,M+1} \times \bar{N}_{M-1}} & \cdots & 0_{\bar{N}_{z,M+1} \times \bar{N}_1} \\ I_{\bar{N}_M \times \bar{N}_M} & 0_{\bar{N}_M \times \bar{N}_{M-1}} & \cdots & 0_{\bar{N}_M \times \bar{N}_1} \\ 0_{\bar{N}_{z,M} \times \bar{N}_M} & 0_{\bar{N}_{z,M} \times \bar{N}_{M-1}} & \cdots & 0_{\bar{N}_{z,M} \times \bar{N}_1} \\ 0_{\bar{N}_{z,M-1} \times \bar{N}_M} & I_{\bar{N}_{M-1} \times \bar{N}_{M-1}} & \cdots & 0_{\bar{N}_{z,M-1} \times \bar{N}_1} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{\bar{N}_{z,1} \times \bar{N}_M} & 0_{\bar{N}_{z,1} \times \bar{N}_{M-1}} & \cdots & I_{\bar{N}_1 \times \bar{N}_1} \\ 0_{\bar{N}_{z,1} \times \bar{N}_M} & 0_{\bar{N}_{z,1} \times \bar{N}_{M-1}} & \cdots & 0_{\bar{N}_{z,1} \times \bar{N}_1} \end{bmatrix} \quad , \tag{5.281}$$

[22] This procedure works for any DMT energy distribution, not just limited to, but of course including, water-filling.

[23] the total number of real dimensions is, as always consistently, $N_i = 2\bar{N}_i$, twice the number of complex tones.

For the real baseband case, J_g is an $N \times N^*$ matrix given by[24]

$$J_g = \left[\begin{array}{cccc} 0_{\bar{N}_{z,1}/2\times\bar{N}_M/2} & 0_{\bar{N}_{z,1}/2\times\bar{N}_{M-1}/2} & \cdots & 0_{\bar{N}_{z,1}/2\times\bar{N}_1/2} \\ 0_{\bar{N}_{z,1}/2\times\bar{N}_M/2} & 0_{\bar{N}_{z,1}/2\times\bar{N}_{M-1}/2} & \cdots & I_{\bar{N}_1/2\times\bar{N}_1/2} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{\bar{N}_{z,M-1}/2\times\bar{N}_M/2} & I_{\bar{N}_{M-1}/2\times\bar{N}_{M-1}/2} & \cdots & 0_{\bar{N}_{z,M-1}/2\times\bar{N}_1/2} \\ 0_{\bar{N}_{z,M}/2\times\bar{N}_M}/2 & 0_{\bar{N}_{z,M}/2\times\bar{N}_{M-1}/2} & \cdots & 0_{\bar{N}_{z,M}/2\times\bar{N}_1/2} \\ I_{\bar{N}_M/2\times\bar{N}_M/2} & 0_{\bar{N}_M/2\times\bar{N}_{M-1}/2} & \cdots & 0_{\bar{N}_M/2\times\bar{N}_1/2} \\ 0_{\bar{N}_{z,M+1}/2\times\bar{N}_M/2} & 0_{\bar{N}_{z,M+1}/2\times\bar{N}_{M-1}/2} & \cdots & 0_{\bar{N}_{z,M+1}/2\times\bar{N}_1/2} \\ ---- & ----- & ----- & ----- \\ 0_{\bar{N}_{z,M+1}/2\times\bar{N}_M/2} & 0_{\bar{N}_{z,M+1}/2\times\bar{N}_{M-1}/2} & \cdots & 0_{\bar{N}_{z,M+1}/2\times\bar{N}_1/2} \\ I_{\bar{N}_M/2\times\bar{N}_M/2} & 0_{\bar{N}_M/2\times\bar{N}_{M-1}/2} & \cdots & 0_{\bar{N}_M/2\times\bar{N}_1/2} \\ 0_{\bar{N}_{z,M}/2\times\bar{N}_M}/2 & 0_{\bar{N}_{z,M}/2\times\bar{N}_{M-1}/2} & \cdots & 0_{\bar{N}_{z,M}/2\times\bar{N}_1/2} \\ 0_{\bar{N}_{z,M-1}/2\times\bar{N}_M/2} & I_{\bar{N}_{M-1}/2\times\bar{N}_{M-1}/2} & \cdots & 0_{\bar{N}_{z,M-1}/2\times\bar{N}_1/2} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{\bar{N}_{z,1}/2\times\bar{N}_M/2} & 0_{\bar{N}_{z,1}/2\times\bar{N}_{M-1}/2} & \cdots & I_{\bar{N}_1/2\times\bar{N}_1/2} \\ 0_{\bar{N}_{z,1}/2\times\bar{N}_M/2} & 0_{\bar{N}_{z,1}/2\times\bar{N}_{M-1}/2} & \cdots & 0_{\bar{N}_{z,1}/2\times\bar{N}_1/2} \end{array}\right] \quad . \tag{5.282}$$

Since the zero-energy tones between the bands carry no information, a nonsingular input $\boldsymbol{u}$ can be associated with only the information-carrying (non-zero input) tones as:

- **complex case** M size-$\bar{N}_i$ IFFT's $\mathcal{Q}^*_{\bar{N}_i}$ or
- **real case** M size-N_i conjugate-symmetric IFFT's.

A block FFT matrix operates on each of the non-singular input components individually: **For the complex case**, the entire set of non-singular-part $\boldsymbol{X}$ components can be represented by

$$\tilde{\boldsymbol{X}} = \left[\begin{array}{c} \boldsymbol{X}_M \\ \boldsymbol{X}_{M-1} \\ \vdots \\ \boldsymbol{X}_1 \end{array}\right] = \left[\begin{array}{cccc} \mathcal{Q}_{\bar{N}_M} & 0 & ... & 0 \\ 0 & \mathcal{Q}_{\bar{N}_{M-1}} & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & \mathcal{Q}_{\bar{N}_1,n_1} \end{array}\right] \left[\begin{array}{c} \boldsymbol{u}_M \\ \boldsymbol{u}_{M-1} \\ \vdots \\ \boldsymbol{u}_1 \end{array}\right] = \tilde{\mathcal{Q}}\boldsymbol{u} \quad . \tag{5.283}$$

For the real baseband-DMT case, the additional subdivision of an FFT matrix is necessary as

$$\mathcal{Q}_{N_i} = \left[\begin{array}{c} \mathcal{Q}^-_{N_i} \\ \mathcal{Q}^+_{N_i} \end{array}\right] \tag{5.284}$$

to form

$$\tilde{\boldsymbol{X}} = \left[\begin{array}{c} \boldsymbol{X}^-_1 \\ \boldsymbol{X}^-_2 \\ \vdots \\ \boldsymbol{X}^-_M \\ \boldsymbol{X}^+_M \\ \vdots \\ \boldsymbol{X}^+_2 \\ \boldsymbol{X}^+_1 \end{array}\right] = \left[\begin{array}{cccc} 0 & 0 & ... & \mathcal{Q}^-_{N_1} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \mathcal{Q}^-_{N_{M-1}} & ... & 0 \\ \mathcal{Q}^-_{N_M} & 0 & ... & 0 \\ \mathcal{Q}^+_{N_M} & 0 & ... & 0 \\ 0 & \mathcal{Q}^+_{N_{M-1}} & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & \mathcal{Q}^+_{N_1} \end{array}\right] \left[\begin{array}{c} \boldsymbol{u}_M \\ \boldsymbol{u}_{M-1} \\ \vdots \\ \boldsymbol{u}_1 \end{array}\right] = \tilde{\mathcal{Q}}\boldsymbol{u} \quad . \tag{5.285}$$

Also, then for both real and complex cases, understanding that $\tilde{\mathcal{Q}}$ is defined differently in these cases,

$$\boldsymbol{X} = J_g\tilde{\boldsymbol{X}} = J_g\tilde{Q}\boldsymbol{u} \quad . \tag{5.286}$$

[24] When $\bar{N}_{M/2}$ is an odd number, then the identity matrices for positive and negative frequencies will be slightly different in the equation below – see later examples in this Section.

which leads a computation of a time-domain $\boldsymbol{u}$ autocorrelation matrix of

$$R_{\boldsymbol{uu}} = \tilde{\mathcal{Q}}^* R_{\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}} \tilde{\mathcal{Q}} \tag{5.287}$$

for the complex case. In the real case, there is some regrouping of positive and negative frequencies together for each of the bands in forming $R_{\boldsymbol{uu}}$ terms. In either the real or complex cases, if $\boldsymbol{u}$ is considered to be a stacked vector of N_i-dimensional baseband-equivalent and decimated time-domain contributions of the different bands. Those contributions are $\boldsymbol{u}_i$ $i = 1, ..., M$, and

$$R_{\boldsymbol{uu}} = \begin{bmatrix} R_{\boldsymbol{uu}}(M) & 0 & ... & 0 \\ 0 & R_{\boldsymbol{uu}}(M-1) & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & ... & 0 & R_{\boldsymbol{uu}}(1) \end{bmatrix} \quad . \tag{5.288}$$

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

$$R_{\boldsymbol{uu}} = \tilde{\mathcal{Q}}^*_{\bar{N}_i} \begin{bmatrix} R^-_{\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}}(i) & 0 \\ 0 & R^+_{\tilde{\boldsymbol{X}}\tilde{\boldsymbol{X}}} \end{bmatrix} \tilde{\mathcal{Q}}_{\bar{N}_i} \tag{5.289}$$

The relationship between the modulated channel input $\boldsymbol{x}$ and $\boldsymbol{u}$ is

$$\boldsymbol{x} = \mathcal{Q}^* \boldsymbol{X} = \mathcal{Q}^* J_g \tilde{\mathcal{Q}} \boldsymbol{u} \quad . \tag{5.290}$$

The M corresponding summed components of $\boldsymbol{x}$ could be called $\boldsymbol{x}_i$, each corresponding to $\boldsymbol{u}_i$. Then

$$\boldsymbol{x} = \sum_{i=1}^{M} \boldsymbol{x}_i \quad , \tag{5.291}$$

where $\boldsymbol{x}_i$ is formed with all other components of $\boldsymbol{u}_{j \neq i} = 0$. Each of these bands could appear stationary if considered by itself. The multiplication by the matrix $\mathcal{Q}^* J_g \tilde{\mathcal{Q}}$ essentially performs carrier modulation and interpolation. 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 non-white inputs $\boldsymbol{u}_i$ can be related to white inputs $\boldsymbol{v}_i$ by the usual Cholesky factorization

$$R_{\boldsymbol{uu}} = G_u R_{\boldsymbol{vv}} G_u^* \quad , \tag{5.292}$$

or

$$R_{\boldsymbol{uu}}(i) = G_u(i) R_{\boldsymbol{vv}}(i) G_u^*(i) \tag{5.293}$$

breaking the Cholesky factorization into M such factorizations. G_u 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

$$A = \underbrace{\mathcal{Q}^* J_g \tilde{\mathcal{Q}}}_{f_{c,i} \text{ mods}} \cdot \underbrace{G_u}_{\text{causal filter}} \quad . \tag{5.294}$$

The G_u will converge to stationary filters in the middle rows of the Cholesky factors of each block 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 subchannel of the MMSE-DFE system.

In fact, any nonsingular input with transmit matrix $\mathcal{Q}$ and $\mathcal{E}_n \neq 0 \ \forall \ n = 1, ..., N$ can be used to construct an acceptable (PW satsifying) input for a CDFE. Such an input necessarily will have a circulant $R_{\boldsymbol{xx}} = \mathcal{Q}^* \text{diag}\{\mathcal{E}_n\} \mathcal{Q}$ (the IDFT in this form always produces a circulant matrix). Thus, the CDFE channel input $R_{\boldsymbol{xx}}$ is circulant, and the resultant $\boldsymbol{x}$ represents a sum of contributions from the different bands. As $N \to \infty$, the diagonal terms of $\mathcal{E}_n$ will become constant, and the rows of $G_u \to G(D)$ for each DFE as in Section 5.6.

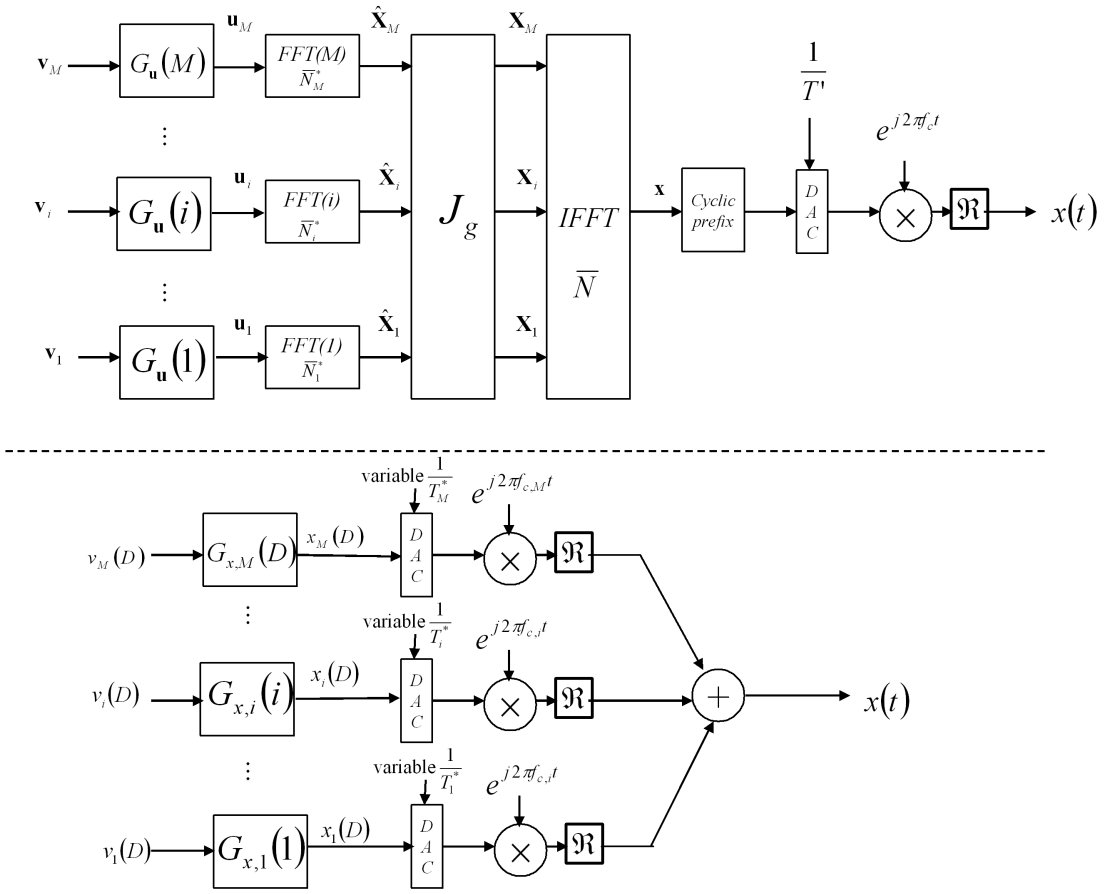


Figure 5.31: Comparison of interpolated CDFE transmitter structure for multiple bands with the infinite-length multi-MMSE DFE transmitter.

Then for complex systems, the actual carrier frequencies of each band are (where Δ_i is the first nonzero tone index of the i^{th} band)

$$f_{c,i} = f_c + \frac{2\pi}{NT'}\left(\frac{\bar{N}_i}{2} + \Delta_i\right) \quad , \tag{5.295}$$

The symbol rate for each band is

$$\frac{1}{T_i^*} = \frac{\bar{N}_i}{\bar{N}T'} \quad . \tag{5.296}$$

Figures 5.31 and 5.32 compare the CDFE transmitter and receiver respectively with the corresponding limiting structures of the multi-band MMSE-DFE.

EXAMPLE 5.5.2 (Real baseband optimized CDFE for 3dB Gap) The $1 + .9D^{-1}$ example 4.6.1 is revisted here with a gap of 0 dB and the CDFE. The following matlab commands illustrate the sequence of computations

```
>> [gn,en_bar,bn_bar,Nstar,b_bar]=DMTra([.9 1],10,1,8,0)
gn = 2.9680   10.0000   17.0320   19.9448   17.0320   10.0000    2.9680    0.0552
en_bar = 1.2415    1.2329    1.1916    0.9547         0
```

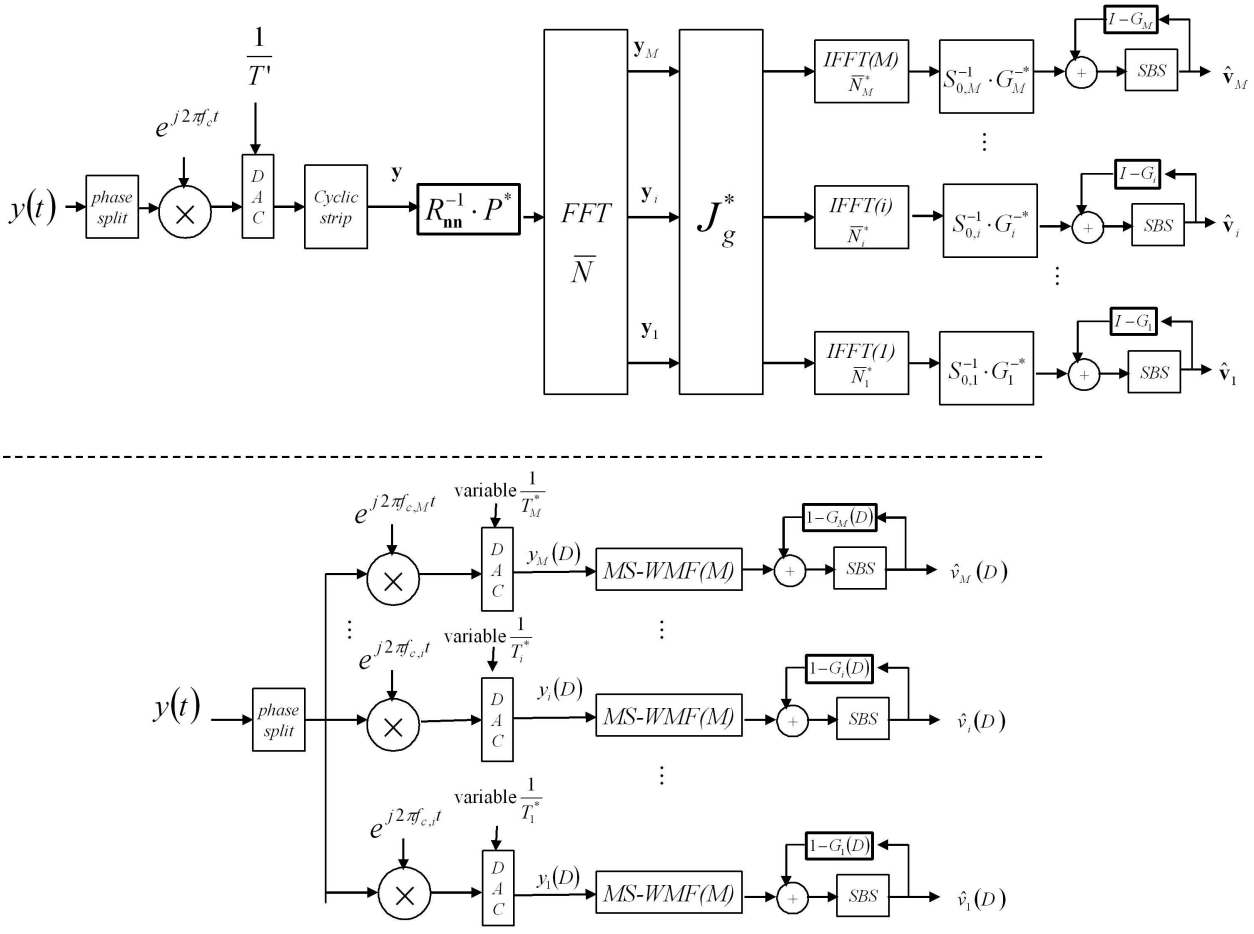


Figure 5.32: Comparison of interpolated CDFE receiver structure for multiple bands with infinite-length multi-MMSE DFE receiver.

```
bn_bar = 2.3436    2.2297    1.8456    0.9693         0
Nstar = 7
b_bar = 1.3814
>> 10*log10(2^(2*b_bar) -1) = 7.6247 dB
>> rXX=diag([en_bar(2) en_bar(3) en_bar(4) 0 en_bar(4) en_bar(3) en_bar(2) en_bar(1)]) =

    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         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    1.2415

>> rXXbar=diag([en_bar(2) en_bar(3) en_bar(4) en_bar(4) en_bar(3) en_bar(2) en_bar(1)]) =

    1.2329         0         0         0         0         0         0
         0    1.1916         0         0         0         0         0
         0         0    0.9547         0         0         0         0
         0         0         0    0.9547         0         0         0
         0         0         0         0    1.1916         0         0
         0         0         0         0         0    1.2329         0
         0         0         0         0         0         0    1.2415

>> 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

>> Gubar=(J7*chol(J7*ruu*J7)*J7)' =

    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

>> Gu=Gubar*inv(diag(diag(Gubar))) =

    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

>> rvv=(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 ];

>> A=real(Q'*Jg*Qtilde*Gu) =

    0.9120   -0.0404    0.0248   -0.0375    0.0603   -0.0981    0.1912
    0.2861    0.8644   -0.1288    0.0736   -0.0702    0.0878   -0.1335
   -0.1853    0.4327    0.7741   -0.1786    0.1025   -0.0873    0.0992
    0.1483   -0.2037    0.5807    0.6501   -0.1925    0.1108   -0.0878
   -0.1237    0.1322   -0.1994    0.7168    0.5033   -0.1753    0.0992
    0.1001   -0.0879    0.1017   -0.1660    0.8286    0.3460   -0.1335
   -0.0686    0.0483   -0.0459    0.0569   -0.1002    0.9054    0.1912
   -0.0000   -0.0000    0.0000   -0.0000   -0.0000   -0.0000    0.9354

>> C=[.9
zeros(6,1)
1];
>> R=[.9 1 0 0 0 0 0 0];
>> P=toeplitz(C,R) =

    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

>> Pt=(1/sqrt(.181))*P*A =
```

```
    2.6018    1.9463   -0.2502    0.0936   -0.0375   -0.0011    0.0906
    0.1697    2.8457    1.5471   -0.2641    0.0925   -0.0195   -0.0493
   -0.0433    0.4365    3.0026    1.1503   -0.2356    0.0757    0.0035
    0.0231   -0.1202    0.7597    3.0600    0.7758   -0.1776    0.0474
   -0.0263    0.0730   -0.1830    1.1260    3.0125    0.4426   -0.1040
    0.0506   -0.0724    0.1072   -0.2176    1.5174    2.8602    0.1669
   -0.1451    0.1022   -0.0971    0.1203   -0.2120    1.9154    2.6031
    2.1437   -0.0950    0.0584   -0.0881    0.1417   -0.2306    2.4282

>> Rf=Pt'*Pt =

   11.4207    5.3013   -0.3515   -0.0274    0.2784   -0.6529    5.0672
    5.3013   12.1213    5.0985   -0.3165    0.0690    0.0395    0.0474
   -0.3515    5.0985   12.1067    5.1003   -0.3252    0.0889   -0.1265
   -0.0274   -0.3165    5.1003   12.1029    5.0991   -0.3244    0.1165
    0.2784    0.0690   -0.3252    5.0991   12.1100    5.0770   -0.2400
   -0.6529    0.0395    0.0889   -0.3244    5.0770   12.1361    4.8500
    5.0672    0.0474   -0.1265    0.1165   -0.2400    4.8500   12.7237

>> Rbinv=Rf+inv(rvv) =

   12.3065    5.3013   -0.3515   -0.0274    0.2784   -0.6529    5.0672
    5.3013   13.0026    5.0985   -0.3165    0.0690    0.0395    0.0474
   -0.3515    5.0985   12.9870    5.1003   -0.3252    0.0889   -0.1265
   -0.0274   -0.3165    5.1003   12.9831    5.0991   -0.3244    0.1165
    0.2784    0.0690   -0.3252    5.0991   12.9901    5.0770   -0.2400
   -0.6529    0.0395    0.0889   -0.3244    5.0770   13.0150    4.8500
    5.0672    0.0474   -0.1265    0.1165   -0.2400    4.8500   13.5987

>> Gbar=chol(Rbinv) =

    3.5081    1.5112   -0.1002   -0.0078    0.0794   -0.1861    1.4444
         0    3.2740    1.6035   -0.0931   -0.0156    0.0980   -0.6522
         0         0    3.2258    1.6271   -0.0906   -0.0269    0.3299
         0         0         0    3.2135    1.6324   -0.0849   -0.1462
         0         0         0         0    3.2110    1.6286   -0.0300
         0         0         0         0         0    3.2110    1.6282
         0         0         0         0         0         0    2.8818

>> G=inv(diag(diag(Gbar)))*Gbar =

    1.0000    0.4308   -0.0286   -0.0022    0.0226   -0.0531    0.4117
         0    1.0000    0.4898   -0.0284   -0.0048    0.0299   -0.1992
         0         0    1.0000    0.5044   -0.0281   -0.0083    0.1023
         0         0         0    1.0000    0.5080   -0.0264   -0.0455
         0         0         0         0    1.0000    0.5072   -0.0093
         0         0         0         0         0    1.0000    0.5071
         0         0         0         0         0         0    1.0000

>> S0=diag(diag(Gbar))*diag(diag(Gbar)) =

   12.3065         0         0         0         0         0         0
         0   10.7190         0         0         0         0         0
         0         0   10.4057         0         0         0         0
```

```
         0         0         0   10.3269         0         0         0
         0         0         0         0   10.3107         0         0
         0         0         0         0         0   10.3104         0
         0         0         0         0         0         0    8.3049

>> 10*log10(det(rvv*S0)^(1/9) -1) =

    7.6247 dB

>> W=inv(S0)*inv(G') =

    0.0813         0         0         0         0         0         0
   -0.0402    0.0933         0         0         0         0         0
    0.0230   -0.0471    0.0961         0         0         0         0
   -0.0127    0.0267   -0.0488    0.0968         0         0         0
    0.0047   -0.0144    0.0276   -0.0493    0.0970         0         0
    0.0039    0.0047   -0.0145    0.0276   -0.0492    0.0970         0
   -0.0660    0.0284   -0.0056   -0.0124    0.0321   -0.0611    0.1204
```

The limiting transmit filter appears to be $G_x \to 1 + .146D - .07D^2 + .02D^3$, $G \to 1 + .50D - .03D^2$, and $W \to .97 - .049D^{-1} + .028D^{-2} - .0145D^{-3}$ on interior rows, but that A does not approach any nice stationary form. As $N \to \infty$, the energy of the components of $\boldsymbol{v}$ become constant. The modulation and interpolation process is not causal, and will require N^2 operations in the transmitter. Similarly P^* 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 5.5.3 will avoid the non-causality and A altogether through analog resampling.

The following example provides an illustration an implementation that 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 5.5.3 (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

n	0	1	2	3	4	5	6	7
$\mathcal{E}_n$	1.24	1.23	1.19	.96	.96	1.19	1.23	0

Or, equivalently from matlab

```
>> [gn,en_bar,bn_bar,Nstar,b_bar]=waterfill([.9 1],10,1,8,0)
gn =
 2.9680   10.0000   17.0320   19.9448   17.0320   10.0000    2.9680    0.0552
en_bar = 1.2415    1.2329    1.1916    0.9547         0
bn_bar = 2.3436    2.2297    1.8456    0.9693
Nstar = 7
b_bar =  1.5541
```

Figure 5.33 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 samples at spacing $T^* = 8/7$. The DFT size will be $N = 7$ in this case, allowing for one sample of cylic prefix. A TEQ might be necessary to restore intersymbol interference to one sample, 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, $\mathcal{E}_{\boldsymbol{x}} = 64/7$, but because 1/8 of this energy is lost in the cyclic prefix, the energy available for DMT loading is 8 units.

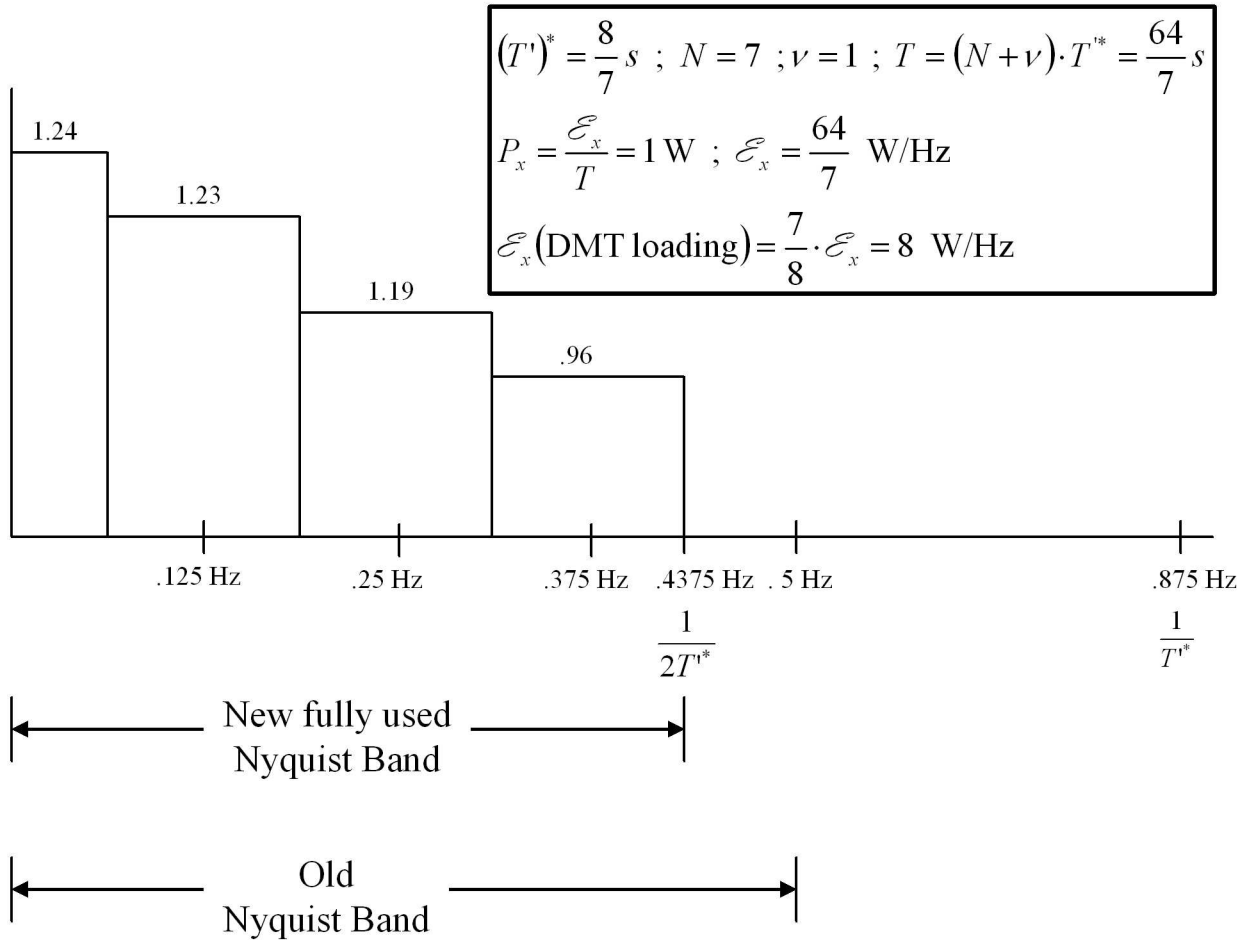


Figure 5.33: Resampled $1 + .9D^{-1}$ channel.

The following sequence of matlab commands shows the resampling interpolation:

```
>> D=exp(j*[0:100]*(7/8)*.01*pi);
>> P7=sqrt(9/8)*ones(1,101)+.9*D;
>> P7=[P7,conj(P7(101:-1:2))];
>> p7=real(ifft(P7));
>> p=[p7(200:201),p7(1:5)]
p = -0.0946    0.8786    1.1812   -0.0564    0.0368   -0.0273    0.0217

>> P=toeplitz([p(1),p(7:-1:2)]',p)
P =
   -0.0946    0.8786    1.1812   -0.0564    0.0368   -0.0273    0.0217
    0.0217   -0.0946    0.8786    1.1812   -0.0564    0.0368   -0.0273
   -0.0273    0.0217   -0.0946    0.8786    1.1812   -0.0564    0.0368
    0.0368   -0.0273    0.0217   -0.0946    0.8786    1.1812   -0.0564
   -0.0564    0.0368   -0.0273    0.0217   -0.0946    0.8786    1.1812
    1.1812   -0.0564    0.0368   -0.0273    0.0217   -0.0946    0.8786
    0.8786    1.1812   -0.0564    0.0368   -0.0273    0.0217   -0.0946
>> P=sqrt(1/.181)*P
P =
   -0.2223    2.0652    2.7764   -0.1326    0.0865   -0.0643    0.0511
    0.0511   -0.2223    2.0652    2.7764   -0.1326    0.0865   -0.0643
   -0.0643    0.0511   -0.2223    2.0652    2.7764   -0.1326    0.0865
    0.0865   -0.0643    0.0511   -0.2223    2.0652    2.7764   -0.1326
   -0.1326    0.0865   -0.0643    0.0511   -0.2223    2.0652    2.7764
    2.7764   -0.1326    0.0865   -0.0643    0.0511   -0.2223    2.0652
    2.0652    2.7764   -0.1326    0.0865   -0.0643    0.0511   -0.2223
```

This channel will now have the water-fill input energies allocated to it, as executed by the following matlab commands:

```
>> J7=hankel([zeros(1,6),1]')
J7 =
     0     0     0     0     0     0     1
     0     0     0     0     0     1     0
     0     0     0     0     1     0     0
     0     0     0     1     0     0     0
     0     0     1     0     0     0     0
     0     1     0     0     0     0     0
     1     0     0     0     0     0     0
>> Q7=(1/sqrt(7))*J7*fft(J7);
>> rXX=diag([1.23,1.19,.96,.96,1.19,1.23,1.24])
rXX =
    1.2300         0         0         0         0         0         0
         0    1.1900         0         0         0         0         0
         0         0    0.9600         0         0         0         0
         0         0         0    0.9600         0         0         0
         0         0         0         0    1.1900         0         0
         0         0         0         0         0    1.2300         0
         0         0         0         0         0         0    1.2400
>> rxx=real(Q7'*rXX*Q7)
rxx =
    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
```

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

```
>> Gxbar=(J7*chol(J7*rxx*J7)*J7)'
Gxbar =
    1.0629    0.0741   -0.0345    0.0078    0.0155   -0.0385    0.0687
         0    1.0654    0.0717   -0.0339    0.0088    0.0129   -0.0340
         0         0    1.0660    0.0715   -0.0344    0.0101    0.0107
         0         0         0    1.0660    0.0716   -0.0348    0.0107
         0         0         0         0    1.0661    0.0711   -0.0340
         0         0         0         0         0    1.0668    0.0687
         0         0         0         0         0         0    1.0690
>> Gx=Gxbar*inv(diag(diag(Gxbar)))
Gx =
    1.0000    0.0695   -0.0323    0.0073    0.0146   -0.0361    0.0643
         0    1.0000    0.0673   -0.0318    0.0083    0.0121   -0.0318
         0         0    1.0000    0.0670   -0.0323    0.0094    0.0100
         0         0         0    1.0000    0.0671   -0.0326    0.0100
         0         0         0         0    1.0000    0.0666   -0.0318
         0         0         0         0         0    1.0000    0.0643
         0         0         0         0         0         0    1.0000
>> rvv=diag(diag(Gxbar))*diag(diag(Gxbar))
rvv =
    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
```

The proximity of the input with resampling to a white input in that G_x is close to a diagonal matrix, converging to $G_x(D) = 1 + .07D$. The CDFE then follows in the usual manner:

```
>> Pt=P*Gx
Pt =
   -0.2223    2.0497    2.9225   -0.0138    0.0018    0.0051   -0.0093
    0.0511   -0.2188    2.0486    2.9223   -0.0140    0.0021    0.0045
   -0.0643    0.0466   -0.2168    2.0482    2.9217   -0.0141    0.0024
    0.0865   -0.0582    0.0440   -0.2162    2.0493    2.9178   -0.0139
   -0.1326    0.0773   -0.0541    0.0431   -0.2180    2.0539    2.9048
    2.7764    0.0604   -0.0121   -0.0340    0.0833   -0.3179    2.2322
    2.0652    2.9199   -0.0126    0.0043   -0.0012    0.0019   -0.1730
>> Rf=Pt'*Pt
Rf =
   12.0545    5.7128   -0.5798   -0.0891    0.2463   -0.8985    5.4559
    5.7128   12.7905    5.4878   -0.5457    0.0084   -0.0155   -0.1648
```

```
   -0.5798    5.4878   12.7898    5.4905   -0.5559    0.0432   -0.2013
   -0.0891   -0.5457    5.4905   12.7846    5.4880   -0.5544    0.0696
    0.2463    0.0084   -0.5559    5.4880   12.7908    5.4639   -0.4687
   -0.8985   -0.0155    0.0432   -0.5544    5.4639   12.8333    5.2158
    5.4559   -0.1648   -0.2013    0.0696   -0.4687    5.2158   13.4509
>> Rbinv=Rf+inv(rvv)
Rbinv =
   12.9397    5.7128   -0.5798   -0.0891    0.2463   -0.8985    5.4559
    5.7128   13.6714    5.4878   -0.5457    0.0084   -0.0155   -0.1648
   -0.5798    5.4878   13.6698    5.4905   -0.5559    0.0432   -0.2013
   -0.0891   -0.5457    5.4905   13.6646    5.4880   -0.5544    0.0696
    0.2463    0.0084   -0.5559    5.4880   13.6705    5.4639   -0.4687
   -0.8985   -0.0155    0.0432   -0.5544    5.4639   13.7119    5.2158
    5.4559   -0.1648   -0.2013    0.0696   -0.4687    5.2158   14.3259
>> Gbar=chol(Rbinv)
Gbar =
    3.5972    1.5881   -0.1612   -0.0248    0.0685   -0.2498    1.5167
         0    3.3390    1.7202   -0.1516   -0.0301    0.1141   -0.7707
         0         0    3.2687    1.7583   -0.1509   -0.0592    0.4188
         0         0         0    3.2480    1.7704   -0.1352   -0.2297
         0         0         0         0    3.2416    1.7630   -0.0388
         0         0         0         0         0    3.2414    1.7723
         0         0         0         0         0         0    2.8391
>> G=inv(diag(diag(Gbar)))*Gbar
G =
    1.0000    0.4415   -0.0448   -0.0069    0.0190   -0.0694    0.4216
         0    1.0000    0.5152   -0.0454   -0.0090    0.0342   -0.2308
         0         0    1.0000    0.5379   -0.0462   -0.0181    0.1281
         0         0         0    1.0000    0.5451   -0.0416   -0.0707
         0         0         0         0    1.0000    0.5439   -0.0120
         0         0         0         0         0    1.0000    0.5468
         0         0         0         0         0         0    1.0000
>> S0=diag(diag(Gbar))*diag(diag(Gbar))
S0 =
   12.9397         0         0         0         0         0         0
         0   11.1492         0         0         0         0         0
         0         0   10.6847         0         0         0         0
         0         0         0   10.5495         0         0         0
         0         0         0         0   10.5077         0         0
         0         0         0         0         0   10.5065         0
         0         0         0         0         0         0    8.0605
>> snr=10*log10(det(rvv*S0)^(1/8)-1)
snr  =    8.9118
>> ibar=.5*log(det(rvv*S0)^(1/8))/log(2)
ibar =    1.5674
```

Thus, the higher SNR must be interpreted in terms of 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 7.6 dB again. In this case, the good DFE performance is obtained by analog resampling rather than the digital interpolation. 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 $G(D)$ of the MMSE-DFE for this new sampling rate on this channel of $1/T = 7/8$, which looks close to $1 + .55D - .04D^2$. The combination of matched filter and feedforward matrix is

```
>> W=inv(S0)*inv(G')*Pt'
W =
   -0.0172    0.0039   -0.0050    0.0067   -0.0102    0.2146    0.1596
    0.1926   -0.0216    0.0067   -0.0087    0.0122   -0.1045    0.1801
    0.1690    0.2036   -0.0242    0.0091   -0.0122    0.0667   -0.0893
   -0.0843    0.1651    0.2076   -0.0258    0.0112   -0.0428    0.0591
    0.0565   -0.0824    0.1635    0.2094   -0.0271    0.0284   -0.0387
   -0.0391    0.0568   -0.0827    0.1638    0.2092   -0.0241    0.0291
    0.0642   -0.0701    0.0906   -0.1264    0.2248    0.1007   -0.0727
```

which appears to be $W(D) \approx .16 + .2D^{-1}$ plus other smaller terms. 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.

As $\Gamma \to 0$, the CDFE and DMT perform the same when the resampling/interpolation have been executed for the CDFE, both have nonsingular and the same $R_{\boldsymbol{xx}}$. DMT would clearly be preferred, because it has no feedback section and can be implemented with $Nlog_2(N)$ operations as compared to N^2 for the CDFE. Also, DMT has a higher SNR for any non-zero gap, and thus would be better in any practical system where gaps must be greater than 0 dB. The CDFE may see more internal values of the matrix G and the feedforward filter becoming zero or negligible, thus allowing a highly channel dependent reduction in some of the computation. In Example 5.5.3, with 6 feedforward taps and 3 feedback taps, the receiver complexity per 8-sample 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 as usual. In Example 5.5.2, the CDFE receiver complexity is$1.5N^2 + N$ per symbol, or 104 operations per sample (but does avoid the resampling, which was not included in the complexity calculation for Example 5.5.3.) 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 dB on the $1 + .9D^{-1}$ 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 for 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 sample, while the CDFE would need to have the capability to perform up to N operations per sample. 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 ($\Gamma = 0$ 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 5.5.3 shows a bit, 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. However, $\Gamma = 0$ dB implies infinite coding and decoding delay, so a better delay would then be mitigated.

Thus, claims that the reader may encournter that a MMSE-DFE with QAM performs the same, but is less complex are unfounded in transmission funamentals – instead in reality, a very special effort

has to be made to make the performance the same, and in which case the MMSE-DFE is always more complex. For some reason, this misunderstanding abounds today without foundation despite efforts by the author and others to correct the misinformed. The sole benefit of the CDFE, or equivalently the MMSE-DFE, is the possibility of a shorter delay if the filter settings converge to a constant setting with many zeros in the trailing positions. Furthermore, systems with large margin, like the 6 dB used in DSL, then necessarily have effectively larger gaps that cannot go to zero dB even with good codes. Thus, it is not possible for the CDFE to match the performance of DMT with non zero margins.

5.5.5 Relationships ZF and MMSE GDFE's

This subsection parallels Subsection 5.2.6, finding an interesting relationship between ZF and MMSE GDFEs under 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.

This result is conveniently expressed in terms of a square non-singular equivalent channel H so that after any channel singular elimination and input singularity elimination, the white-noise equivalent channel is

$$\boldsymbol{z} = H\boldsymbol{u} + \boldsymbol{n} \tag{5.297}$$

where $R_{\boldsymbol{nn}} = I$. Singular value decomposition, as well as QR factorization, are possible on the non-singular H:

$$H = D \cdot G \cdot Q^* = F\Lambda M^* \quad , \tag{5.298}$$

where D is nonsingular diagonal, G is monic lower triangular, and Q is an orthogonal matrix.[25] F, Λ, and M are the nonsingular square matrices that arise from singular value decomposition of the nonsingular square H.

The best water-filling input covariance is determined by vector coding as

$$R_{\boldsymbol{uu}} = M\left[K \cdot I - \Lambda^{-2}\right] M^* \quad . \tag{5.299}$$

Cholesky Decomposition of $Q^* R_{\boldsymbol{uu}} Q$ produces

$$Q^* R_{\boldsymbol{uu}} Q = G_x S_x G_x^* \quad , \tag{5.300}$$

(where G_x is monic lower triangular and S_x is diagonal) from which emerges the relation

$$R_{\boldsymbol{uu}} = M\left[K \cdot I - \Lambda^{-2}\right] M^* = Q \cdot G_x \cdot S_x \cdot G_x^* \cdot Q^* \quad . \tag{5.301}$$

Figure 5.34 illustrates the transmitter and simplified receiver. The overall channel is determined by monic upper triangular

$$G_{tot} = G \cdot G_x \quad , \tag{5.302}$$

and the "white" input $\boldsymbol{v}$ has $R_{\boldsymbol{vv}} = S_x$. A ZF-GDFE then uses a feedback section of G and the corresponding SNRs are

$$\mathrm{SNR}_n = S_{x,n} \cdot D_n^2 \tag{5.303}$$

on each of the subchannels. Basic matrix multiplication produces

$$H^*H = M \cdot \Lambda^2 \cdot M^* = Q \cdot G^* \cdot D^2 \cdot G \cdot Q^* \tag{5.304}$$

with corresponding inverse

$$[H^*H]^{-1} = H^{-1}H^{-*} = M \cdot \Lambda^{-2} \cdot M^* = Q \cdot G^{-1} \cdot D^{-2} \cdot G^{-*} \cdot Q^* \quad . \tag{5.305}$$

Substitution of (5.305) into (5.301) yields

$$K \cdot I = Q \cdot G_x \cdot S_x \cdot G_x^* \cdot Q^* + Q \cdot G^{-1} \cdot D^{-2} \cdot G^{-*} \cdot Q^* \tag{5.306}$$

$$K \cdot G \cdot G^* = G_{tot} \cdot S_x \cdot G_{tot}^* + D^{-2} \tag{5.307}$$

[25] The Matlab command [Q,R]=qr(A) produces A=Q*R where R is upper triangular. By executing [Q,R] = qr(A'), one obtains A=R'*Q' where R' is lower triangular.

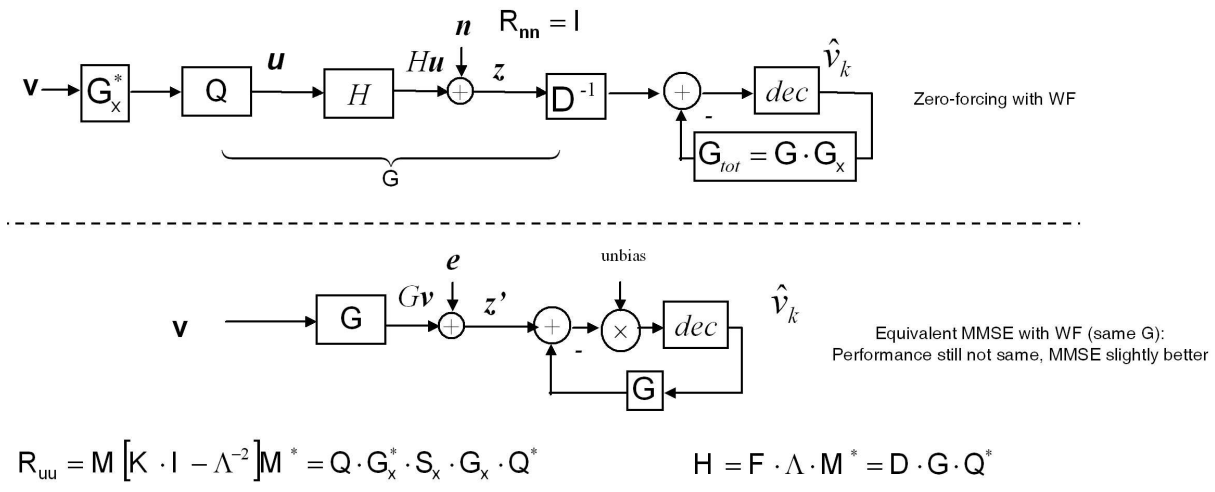


Figure 5.34: Relationship diagram for the ZF-GDFE and GDFE with water-filling input choice.

Equation 5.307 relates that the G of QR factorization essentially determines the shape of the channel plus noise under water-filling after equalization by the lossless transformations as in Figure 5.34. Effectively the MMSE-GDFE is determined by this same G, since this structure has maximum mutual information. Despite the common shape, the ZF-GDFE will perform worse than the MMSE-GDFE unless $G_x = I$. However, QR factorization and Cholesky factorization are much easier to execute in a real system than is singular-value decomposition or eigenvector decomposition. So if a nonsingular square H can be found easily, Equation (5.307) implements a direct computation of necessary quantities and input energies

$$K \cdot G^* G - D^{-2} = G_{tot}^* \cdot S_x \cdot G_{tot} \quad . \tag{5.308}$$

The left side of (5.308) uses the quantities G and D of QR factorization, and K, which for water-filling is easily determined as

$$K = \bar{\mathcal{E}}_{\boldsymbol{x}} - \frac{1}{N^*} \sum_{n=1}^{N^*} \frac{1}{\lambda_n^2} = \bar{\mathcal{E}}_{\boldsymbol{x}} - \frac{1}{N^*} \text{trace} \left\{ [H^* H]^{-1} \right\} \quad . \tag{5.309}$$

Direct Cholesky factorization of the quantity computed on the left in (5.308) produces G_{tot} and $G_x = G_{tot} \cdot G^{-1}$. No singular value decomposition is then necessary. Either a ZF or a MMSE-GDFE can then be implemented. The reader is reminded that $N^* = N$ for this simplification to be of use. This result can be useful only when the water-filling input uses all dimensions of the channel.

worst-case noise

Worst-case noise autocorrelation can be found easily from optimization theory with the constraint that the diagonal elements of the noise autocorrelation matrix $R_{\boldsymbol{nn}}$ are fixed – that is fixed power of the noise on each dimension (or fixed covariance elements/matrices among sets of dimensions along a block diagonal of a matrix). Such worst case noise has a correspondence with the broadcast channel of Chapter 14 where no receiver inter-dimensional coordination is allowed (since the dimensions are different users who may be in physically separated locations).

Correlation between noises of different users cannot be mitigated by an allowable receiver if no receiver-side coordination is permitted, for instance as might occur if user receivers were in different locations. This subsection first simply derives an expression for the worst-case noise that is well-known,

but very difficult to work with as a designer. Instead, the GDFE is subsequently used to provide a different interpretation of worst-case noise, as in the previous subsection. The latter, when combined with water-filling, will avoid the difficult in computation for the designer altogether.

Existence of Worst-Case Noise The mutual information for any vector channel $I(\boldsymbol{x};\boldsymbol{y})$ is a function of the noise, and

$$I(\boldsymbol{x};\boldsymbol{y}) = \frac{1}{2}\log_2 \frac{|PR_{\boldsymbol{xx}}P^* + R_{\boldsymbol{nn}}|}{|R_{\boldsymbol{nn}}|} \quad , \tag{5.310}$$

which is convex in $R_{\boldsymbol{nn}}$. It is minimized of course by an infinite variance noise. If the noise variance on each dimension is held fixed at the set diagonal values of $R^o_{\boldsymbol{nn}}$, so essentially the constraint is that the diagonal of $R_{\boldsymbol{nn}}$ is fixed, then the minimizing noise covariance can be founded from the Lagrangian

$$\min_{R_{\boldsymbol{nn}}} \frac{1}{2}\log_2 |PR_{\boldsymbol{xx}}P^* + R_{\boldsymbol{nn}}| - \frac{1}{2}\log_2 |R_{\boldsymbol{nn}}| + \sum_{n=1}^{N} \lambda_n \left(R_{\boldsymbol{nn}}(n) - R^o_{\boldsymbol{nn}}(n)\right) \quad . \tag{5.311}$$

Then careful setting[26] of the gradient with respect to all the elements in $R_{\boldsymbol{nn}}$ to zero and organization of the result into matrix form produces

$$[PR_{\boldsymbol{xx}}P^* + R_{\boldsymbol{nn}}]^{-1} - R^{-1}_{\boldsymbol{nn}} = -D \quad . \tag{5.312}$$

where D is a diagonal matrix with non-negative real entries

$$D = \text{diag}\{\lambda_n\} \quad . \tag{5.313}$$

Previous EE379C student Mehdi Mohseni has provided a software routine using "interior point" methods that computes iteratively the worst-case noise. This program's channel matrix has dimensions $(K \cdot N) \times m$, which requires some explanation: The input is construed as coming from m independent sources (lines or "antennas" as the comments of the program indicate). If intersymbol interference is included, then the m sources may be the dimensions of a single input (such as the $N + \nu$ inputs of a packet in time as often modeled in this chapter or combinations of many such packets of inputs).

The signals arising from these m sources produce KN output dimensions. This program's N here is not always the same N as packet length, but instead generalizes that concept and refers to N dimensions for each of K groups of such dimensions (in multi-user use in Chapter 14, each group represents a different "user.") The N could represent the output dimensions for instance for each of K output positions/groups. For the most common case in this chapter the program input H is set equal to P and N is set to the number of rows of P while then $m = N + \nu$.

Mohseni's Worst-Case Noise Autocorrelation Matrix

```
% the channel matrix is K*N by m, m is the number of transmitter's antennas,
% K is the number of output dimension groups and N is the number of receiver
% dimensions for each such group.
% Rx is the m by m input covariance matrix
% Rn is the K*N by K*N worst case noise minimizing
% 0.5 * log(det(HRx*H'+Rn)/det(Rn)), Rn has its N by N diganoal sub-blocks
% equal to identity matrix

function [Rn, sumRate] = wcnoise(Rx, H, N)
```

[26] Matrix derivatives become easier to understand and perform if a Taylor's Series representation of $\log_2(x) = \frac{1}{\ln(2)}\sum_{n=1}^{\infty}\frac{(-1)^{n+1}}{n}x^n$ is thought of with x a symmetric non-negative-definite matrix $x = Q\Lambda Q^*$, where Λ is a diagonal matrix of non-negative eigenvalues. In such a form, the derivative is taken with respect to each eigenvalue like a scalar since the Q is invariant to the power of x_n, that is invariant to the value of n. Thus, the derivative of $log_2(R)$ is simply R^{-1}.

```
[n,m] = size(H);
K = n / N;
A = zeros(n,n,n*(n+1)/2);
for i = 1:n
   A(i,i,i) = 1;
end

count = n+1;

for i = 2:n
   for j = 1:i - 1
      A(i,j,count) = 1;
      A(j,i,count) = 1;
      count = count+1;
   end
end

map = zeros(n,n);
for i = 1:K
   map((i-1) * N + 1:i * N,(i-1) * N + 1:i * N) = ones(N,N);
end

NT_max_it = 1000;                                % Maximum number of Newton's
                                                 % method iterations
dual_gap = 1e-5;
mu = 10;                                         % step size for t
alpha = 0.001;                                   % back tracking line search parameters
beta = 0.5;

count = 1;
nerr = 1e-4;                                     % acceptable error for inner loop
                                                 % Newton's method

v_0 = zeros(n*(n+1)/2,1);                        % Strictly feasible point;
v_0(1:n) = 0.5 *  ones(n,1);
v = v_0;
t =1;
l_v = 1;                                         % lambda(v) for newton's method termination

while (1+n)/t > dual_gap
   t =  t * mu;
   l_v = 1;
   count = 1;
   while l_v > nerr & count < NT_max_it

      f_val = 0;                                 % calculating function value
      Rn = zeros(n,n);
      Rnprime = zeros(n,n);

      for i = 1:n*(n+1)/2                        % computing Rn
         Rn = Rn + v(i) * A(:,:,i);
      end

      for i = 1:K
```

```
    Rnprime((i-1) * N + 1:i * N,(i-1) * N + 1:i * N) =
       Rn((i-1) * N + 1:i * N,(i-1) * N + 1:i * N);
end

f_val = t * log(det(H * Rx * H' + Rn)) - (t + 1) * log(det(Rn))
       - log(det(eye(n) - Rnprime));

S = inv(H * Rx * H' + Rn);
Q = inv(eye(n) - Rnprime);
Rn_inv = inv(Rn);
g = zeros(n*(n+1)/2,1);
h = zeros(n*(n+1)/2,n*(n+1)/2);

for i = 1:n*(n+1)/2
   g(i) = t * trace(A(:,:,i) * S) - (t + 1) * trace(A(:,:,i) * Rn_inv)...
        + (sum(sum(A(:,:,i) .* map)) ~= 0) * trace(A(:,:,i) * Q) ;  % gradient
end


for i = 1:n*(n+1)/2
   for j = 1:n*(n+1)/2
      h(i,j) = -t * trace(A(:,:,i) * S * A(:,:,j) * S) + (t + 1) *
           trace(A(:,:,i) * Rn_inv * A(:,:,j) * Rn_inv)...
               +(sum(sum(A(:,:,i) .* map)) ~= 0) * (sum(sum(A(:,:,j) .* map)) ~= 0)
                * trace(A(:,:,i) * Q * A(:,:,j) * Q); %hessian
   end
end


dv = -h\g;                                  % search direction


s = 1;                                      % checking v = v+s*dx feasible
                                            % and also back tracking algorithm

v_new = v + s * dv;

f_new = 0;

Rn_new = zeros(n,n);

for i = 1:n*(n+1)/2
   Rn_new = Rn_new + v_new(i) * A(:,:,i);
end

for i = 1:K
    Rnprime((i-1) * N + 1:i * N,(i-1) * N + 1:i * N) =
       Rn_new((i-1) * N + 1:i * N,(i-1) * N + 1:i * N);
end

f_new = t * log(det(H * Rx * H' + Rn_new)) -
     (t + 1) * log(det(Rn_new)) - log(det(eye(n) - Rnprime));
```

```
feas_check = 1;
if real(eig(Rn_new)) > zeros(n,1)
    feas_check = 1;
else
    feas_check = 0;
end

if real(eig(eye(n) - Rnprime)) > zeros(n,1)
    feas_check = 1;
else
    feas_check = 0;
end

feas_check = feas_check * (f_new < f_val + alpha * s * g' * dv);


while feas_check ~= 1
   s = s * beta

   v_new = v + s * dv;
   f_new = 0;

   Rn_new = zeros(n,n);

   for i = 1:n*(n+1)/2
      Rn_new = Rn_new + v_new(i) * A(:,:,i);
   end

   for i = 1:K
       Rnprime((i-1) * N + 1:i * N,(i-1) * N + 1:i * N) =
          Rn_new((i-1) * N + 1:i * N,(i-1) * N + 1:i * N);
   end

   f_new = t * log(det(H * Rx * H' + Rn_new)) -
           (t + 1) * log(det(Rn_new)) - log(det(eye(n) - Rnprime));

   feas_check = 1;
   if real(eig(Rn_new)) > zeros(n,1)
       feas_check = 1;
   else
       feas_check = 0;
   end
   if real(eig(eye(n) - Rnprime)) > zeros(n,1)
       feas_check = 1;
   else
       feas_check = 0;
   end

   feas_check = feas_check * (f_new < f_val + alpha * s * g' * dv);
end

v = v + s * dv;                          % update v
l_v = -g'*dv                             % lambda(v)^2 for Newton's method
```

```
      count = count + 1;                          % number of Newtons method iterations
   end
end

Rn = zeros(n,n);
for i = 1:n*(n+1)/2
   Rn = Rn + v(i) * A(:,:,i);
end
for i = 1:K
    Rn((i-1) * N + 1:i * N,(i-1) * N + 1:i * N) = eye(N);
end
sumRate = 0.5 * log(det(H * Rx * H' + Rn)/det(Rn));
```

This program appears in usable form on the course web page. The convexity of the mutual information in $R_{\boldsymbol{nn}}$ (and the program!) establishes the existence of the worst-case noise.

A couple of simple solutions are evident without resort to the program. At very high SNR with white noise, then the noise effectively goes to zero (w.r.t. signal) so that $R_{\boldsymbol{nn}}^{-1} = \infty \cdot I$ and the condition is met. So, white noise is worst-case when the SNR is infinite and any large noise (0 dB SNR) is worst-case. (At very low SNR with any correlation of noise, then the two terms in the subtraction are equal and so zero is a diagonal matrix.)

Connection to the GDFE Worst-case noise diagonalizes the overall feedforward processing of a GDFE for certain inputs. The construction of the input first observes the worst-case noise autocorrelation can be rewritten[27] for some input modulation matrix A as

$$\begin{aligned}
\left[PR_{\boldsymbol{xx}}P^* + R_{wcn}\right]^{-1} - R_{wcn}^{-1} &= -D & (5.314)\\
\left[R_{wcn}^{-1/2}PAA^*P^*R_{wcn}^{-*/2} + I\right]^{-1} &= -R_{wcn}^{*/2}DR_{wcn}^{1/2} & (5.315)\\
R_{wcn}^{-1/2}PA\left[I + A^*P^*PA\right]^{-1}A^*P^*R_{wcn}^{-*/2} &= R_{wcn}^{*/2}DR_{wcn}^{1/2} & (5.316)\\
R_{wcn}^{-1}PAR_bA^*P^*R_{wcn}^{-*} &= D & (5.317)\\
R_{wcn}^{-1}PAG^{-1}S_0^{-1}G^{-*}A^*P^*R_{wcn}^{-*} &= D & (5.318)
\end{aligned}$$

By performing the QR factoriztion

$$R_{wcn}^{-1}P = [0R]Q^* \tag{5.319}$$

where R is an upper triangular square matrix and $Q = [\boldsymbol{q}_1 \; Q_1]$ is an orthogonal matrix. In cases with square nonsingular P, the zero element can be eliminated and only then $Q_1 = Q$. This procedure degenerates when the number of output dimensions exceeds the number of input dimensions, see Chapter 14 for how to handle this situation. Then, when R is nonsingular, the choice

$$A = Q_1R^{-1}D_AG \tag{5.320}$$

(where D_A is a diagonal matrix that satisfies $D_AS_0^{-1}D_A = D$, or equivalently each diagonal entry is computed as $D_A(n) = \sqrt{D(n) \cdot S_0(n)}$) satisfies the worst-case-noise equation, where G remains to be determined. Determination of G occurs through the Cholesky factorization

$$Q_1^*R_{\boldsymbol{xx}}Q_1 = UU^* \tag{5.321}$$

where U is an upper triangular matrix. Then,

$$R^{-1}D_AG = U \tag{5.322}$$

or

$$G = D_A^{-1}RU \quad . \tag{5.323}$$

[27] The matrix inversion lemma $[A + BCD]^{-1} = A^{-1} - A^{-1}B[C^{-1} + DA^{-1}B]^{-1}DA^{-1}$ is used.

The GDFE is now determined, and the overall feedforward processing is

$$W = S_0^{-1} G^{-*} A^* P^* R_{wcn}^{-1} = S_0^{-1} D_A = D_A^{-1} D \quad , \tag{5.324}$$

which is thus clearly diagonal. The requirement that R be nonsingular is equivalent to the channel having a rank that is equal to that of R_{wcn}, which is N (the number of channel output dimensions). When the channel has a rank less than N, the input should be projected (as earlier in this chapter) into a smaller number of input dimensions that is equal to the channel rank. A GDFE attempting to estimate these dimensions will have diagonal total feedforward. It is not possible (in some sense trivially) to diagonalize a feedforward section for a larger number of input dimensions than N. The worst-case noise will certainly exist, but the corresponding input to diagonalize does not.

Theorem 5.5.2 (Worst-Case Noise and the GDFE) *The worst-case noise covariance subject to autocorrelation constraints on sets of dimensions allows at least one GDFE receiver whose entire linear processing (ignoring feedback) is diagonal.*

Proof: *From the convexity of the mutual information in noise covariance matrix, a worst-case noise always exists. Following the above development in (5.314) - (5.324), cleary W is diagonal.* **QED.**

Theorem 5.5.2 holds for any input $R_{\boldsymbol{xx}}$ to the channel (not just water-filling), and a corresponding A can be found to synthesize $R_{\boldsymbol{xx}}$ as well as diagonalize the feedforward processing. An interesting use is that with a precoder implementation of the feedback section at the transmitter that the receiver essentially becomes a trivial scaling of each dimension. Thus, all complexity and also coordination of signal processing among the dimensions occurs at the transmitter.

However, vector-coded inputs (with arbitrary energy assignments, but using the singular vectors M for a transmit matrix) lead to a special case that simplifies the worst-case GDFE determination. First, an exercise for the reader easily shows that any product of an orthogonal matrix and a diagonal matrix can be set equal to the product of another diagonal and another orthogonal matrix,

$$Q_{left} D_{right} = D_{left} Q_{right} \quad . \tag{5.325}$$

Thus, when

$$R_{\boldsymbol{xx}} = M S_x M^* \tag{5.326}$$

the receiver-diagonalizing input matrix A for the corresponding worst-case-noise input is given by

$$A = Q_1 R^{-1} D_A G = Q_1 U = M S_x^{1/2} Q_x \tag{5.327}$$

where Q_x is an unknown orthogonal matrix. Since S_x is diagonal, then the product $S_x^{1/2} Q_x = Q_{left} D_{right}$ then forces the solution for U to be a diagonal. Essentially then, when monic, R and G are the same triangular form or the GDFE is then determined by the "QR" factorization of $R_{wcn} P$. The R_{wcn} can then be determined without iterative descent methods as in Mohseni's WCN program. A direct iterative N-step algorithm for its calculation is left as an exercise for the interested reader.

The ZF and MMSE GDFEs are trivially the same when there is no receiver processing (after biases are removed). Thus worst-case noise can be (with the correct input modulation choice corresponding to A) an additional situation for which the ZF and MMSE GDFE receivers have the same performance. The worst-case noise is a function of the input covariance $R_{\boldsymbol{xx}}$. However, the best input covariance is also a function of the noise $R_{\boldsymbol{nn}}$. The combination best input covariance for a corresponding worst-case noise that diagonalizes the GDFE can be found by an iterative loop that first finds a water-filling solution for the diagonal noise covariance (there will be a corresponding data rate b_0), then finds a worst-case noise for that water-filling solution that will not increase the data rate ($b_1 \leq b_0$). If equality holds, then the solution has been found. If not, then the procedure repeats to find a new water-filling solution for that worst-case noise and will not decrease data rate ($b_2 \geq b_1$) and a subsequent new worst-case noise with $b_3 \leq b_2$.

With $q \geq 1$ an integer. $b_{2q+2} \leq b_{2q}$ and $b_{2q+3} \geq b_{2q+1}$. That is successive upper bounds from water-filling steps are decreasing (because the intermediate worst-case noise step searches all covariances

with the same noise diagonal) and successive lower bounds from worst-case noise steps are increasing (because the water-filling step searches all input covariances with same energy constraint), thus forcing convergence since both problems are convex. The solution is sometimes called a "saddle point." This saddle point is a very special one for what are called "broadcast" multi-user channels in Chapter 14. This observation in GDFE theory lead to former EE379C students Wei Yu and Mehdi MOhseni being the first to master the fundamental limits of the broadcast channel.

An additional simplification occurs when the input

Some Examples of WCN and GDFE

EXAMPLE 5.5.4 (Vector-coded input for WCN) This first example uses the 3×4 channel matrix well known in this text as the $1 + .9D^{-1}$ channel. An input with flat 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.

```
>> P =

    0.9000    1.0000         0         0
         0    0.9000    1.0000         0
         0         0    0.9000    1.0000

>> P=(1/sqrt(.181))*P =

    2.1155    2.3505         0         0
         0    2.1155    2.3505         0
         0         0    2.1155    2.3505

>> C=eye(4) =

     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1

>> [F,L,M]=svd(P)

F= -0.5000    0.7071    0.5000
   -0.7071   -0.0000   -0.7071
   -0.5000   -0.7071    0.5000

L=  4.1270         0         0         0
         0    3.1623         0         0
         0         0    1.7228         0

M= -0.2563    0.4730    0.6140   -0.5776
   -0.6472    0.5256   -0.1861    0.5198
   -0.6590   -0.4730   -0.3508   -0.4678
   -0.2848   -0.5256    0.6822    0.4211

>> Pm=M(:,1:3)*M(:,1:3)' =

    0.6664    0.3002   -0.2702    0.2432
    0.3002    0.7298    0.2432   -0.2189
```

```
   -0.2702    0.2432    0.7811    0.1970
    0.2432   -0.2189    0.1970    0.8227

>> Ctemp=Pm*C =

    0.6664    0.3002   -0.2702    0.2432
    0.3002    0.7298    0.2432   -0.2189
   -0.2702    0.2432    0.7811    0.1970
    0.2432   -0.2189    0.1970    0.8227

>> Rxx=Pm*Pm' =

    0.6664    0.3002   -0.2702    0.2432
    0.3002    0.7298    0.2432   -0.2189
   -0.2702    0.2432    0.7811    0.1970
    0.2432   -0.2189    0.1970    0.8227

>> [Rwcn,rate]=wcnoise(Rxx,P,1)

Rwcn =

    1.0000    0.0694   -0.0282
    0.0694    1.0000    0.0694
   -0.0282    0.0694    1.0000

>> J3 = [
     0     0     1
     0     1     0
     1     0     0 ];

>>J4 = [
     0     0     0     1
     0     0     1     0
     0     1     0     0
     1     0     0     0 ];

>> Ptilde=inv(Rwcn)*P =

    2.1281    2.2118   -0.0990    0.0785
   -0.1527    1.9670    2.2214   -0.1697
    0.0707   -0.0742    1.9584    2.3645

>> [Q,R]=qr(J4*Ptilde'*J3);
>> Q=J4*Q*J4 =

    0.5776   -0.8133    0.0669   -0.0230
   -0.5198   -0.4305   -0.7375    0.0241
    0.4678    0.3067   -0.5297   -0.6375
   -0.4211   -0.2433    0.4136   -0.7697

>> R=J4*R*J3;
>> R=R' =

         0   -2.7323   -1.4039    0.0071
```

```
         0         0   -2.7077   -1.2346
         0         0         0   -3.0719

> R*Q' =

    2.1281    2.2118   -0.0990    0.0785
   -0.1527    1.9670    2.2214   -0.1697
    0.0707   -0.0742    1.9584    2.3645

>> Q1=Q(:,2:4) =

   -0.8133    0.0669   -0.0230
   -0.4305   -0.7375    0.0241
    0.3067   -0.5297   -0.6375
   -0.2433    0.4136   -0.7697

>> Rxxrot=Q1'*Rxx*Q1 =

    1.0000   -0.0000    0.0000
   -0.0000    1.0000    0.0000
    0.0000    0.0000    1.0000

>> Ubar=J3*chol(J3*Rxxrot*J3)*J3 =

    1.0000         0         0
   -0.0000    1.0000         0
    0.0000    0.0000    1.0000

>> Ubar=Ubar';
>> Rup=R(:,2:4) =

   -2.7323   -1.4039    0.0071
         0   -2.7077   -1.2346
         0         0   -3.0719

>> DA=diag(diag(Rup*Ubar)) =

   -2.7323         0         0
         0   -2.7077         0
         0         0   -3.0719

>> G=inv(DA)*Rup*Ubar =

    1.0000    0.5138   -0.0026
         0    1.0000    0.4559
         0         0    1.0000

>> A=Q1*inv(Rup)*DA*G =

   -0.8133    0.0669   -0.0230
   -0.4305   -0.7375    0.0241
    0.3067   -0.5297   -0.6375
   -0.2433    0.4136   -0.7697
```

```
>> A*A'-Rxx = (check)

  1.0e-015 *

    0.4441    0.3331   -0.2776   -0.0833
    0.3331    0.5551   -0.1110   -0.4718
   -0.2776   -0.1110    0.2220   -0.0278
   -0.0833   -0.4718   -0.0278    0.8882

>> D=inv(Rwcn) - inv(P*Rxx*P' + Rwcn) =

    0.8819    0.0000   -0.0000
    0.0000    0.8529    0.0000
   -0.0000    0.0000    0.8819

>> Rbinv=eye(3)+A'*P'*inv(Rwcn)*P*A

    8.4657    4.3497   -0.0221
    4.3497   10.8304    3.9077
   -0.0221    3.9077   12.4876

>> Gbar=chol(Rbinv) =

    2.9096    1.4949   -0.0076
         0    2.9318    1.3367
         0         0    3.2712

>> G=inv(diag(diag(Gbar)))*Gbar =

    1.0000    0.5138   -0.0026
         0    1.0000    0.4559
         0         0    1.0000

(checks)

>> S0=diag(diag(Gbar))*diag(diag(Gbar)) =

    8.4657         0         0
         0    8.5956         0
         0         0   10.7006

>> snr=10*log10(det(S0)^(1/4)-1) =

    6.3169

>> W=1/sqrt(.181)* inv(S0)*inv(G')*A'*Ptilde' =

   -0.7586   -0.0000   -0.0000
   -0.0000   -0.7404    0.0000
    0.0000    0.0000   -0.6748
```

This first example above clearly has a diagonal U because the input autocorrelation was constructed with the M of the channel's singular value decomposition. 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.

```
>> C =
     1     1     1
    -1     1     1
     1    -1    -1
    -1    -1     1

>> Ctemp=Pm*C =

   -0.1472    0.9937    1.4801
    0.0325    1.0057    0.5679
    0.0707   -1.0051   -0.6111
   -0.1636   -0.9954    0.6500

>> Ctilde=Ctemp(:,1:3) =

   -0.1472    0.9937    1.4801
    0.0325    1.0057    0.5679
    0.0707   -1.0051   -0.6111
   -0.1636   -0.9954    0.6500

>> Rin=C*C' =

     3     1    -1    -1
     1     3    -3     1
    -1    -3     3    -1
    -1     1    -1     3

>> Rxx=Pm*Rin*Pm'=

    3.1997    1.8351   -1.9137   -0.0029
    1.8351    1.3350   -1.3556   -0.6372
   -1.9137   -1.3556    1.3888    0.5917
   -0.0029   -0.6372    0.5917    1.4401

>> [Rwcn,rate]=wcnoise(Rxx,P,1)

Rwcn =

    1.0000   -0.0541   -0.0373
   -0.0541    1.0000   -0.0044
   -0.0373   -0.0044    1.0000

rate =    3.1377

>> Ptilde=inv(Rwcn)*P =

    2.1247    2.4761    0.2079    0.0886
    0.1154    2.2500    2.3712    0.0152
    0.0797    0.1023    2.1337    2.3539
```

```
>> [Q,R]=qr(J4*Ptilde'*J3);
>> Q=J4*Q*J4 =

    0.5776   -0.8155   -0.0261   -0.0251
   -0.5198   -0.3421   -0.7821   -0.0322
    0.4679    0.3662   -0.4435   -0.6711
   -0.4211   -0.2894    0.4369   -0.7403

>>  R=J4*R*J3;
>> R=R' =

         0   -2.5294   -2.0456   -0.3381
         0         0   -2.8078   -1.6777
         0         0         0   -3.1797

>> Q1=Q(:,2:4) =

   -0.8155   -0.0261   -0.0251
   -0.3421   -0.7821   -0.0322
    0.3662   -0.4435   -0.6711
   -0.2894    0.4369   -0.7403

>> Rxxrot=Q1'*Rxx*Q1 =

    4.8448    0.8359   -1.4282
    0.8359    0.6633   -1.1041
   -1.4282   -1.1041    1.8554

>> Ubar=J3*chol(J3*Rxxrot*J3)*J3 =

    1.9272         0         0
   -0.1771    0.0792         0
   -1.0485   -0.8106    1.3621

>> Rup=R(:,2:4) =

   -2.5294   -2.0456   -0.3381
         0   -2.8078   -1.6777
         0         0   -3.1797

>> Ubar=Ubar' =

    1.9272   -0.1771   -1.0485
         0    0.0792   -0.8106
         0         0    1.3621

>> DA=diag(diag(Rup*Ubar)) =

   -4.8746         0         0
         0   -0.2224         0
         0         0   -4.3311

>> G=inv(DA)*Rup*Ubar =
```

```
    1.0000   -0.0587   -0.7898
         0    1.0000    0.0420
         0         0    1.0000

>> A=Q1*inv(Rup)*DA*G =

   -1.5717    0.1424    0.8421
   -0.6593   -0.0013    0.9489
    0.7057   -0.1000   -0.9385
   -0.5578    0.0859   -1.0590

>> Rbinv=eye(3)+A'*P'*inv(Rwcn)*P*A =

   24.7620   -1.4532  -19.5561
   -1.4532    1.1382    1.1919
  -19.5561    1.1919   35.8268

>> Gbar=chol(Rbinv) =

    4.9761   -0.2920   -3.9300
         0    1.0261    0.0430
         0         0    4.5145

>> S0=diag(diag(Gbar))*diag(diag(Gbar)) =

   24.7620         0         0
         0    1.0529         0
         0         0   20.3803

>> snr=10*log10(det(S0)^(1/4)-1) =

    5.7991

>> W=1/sqrt(.181)* inv(S0)*inv(G')*A'*Ptilde' =

   -0.4627   -0.0000    0.0000
    0.0000   -0.4964   -0.0000
   -0.0000   -0.0000   -0.4995
```

In this case, the input is not a vector-coding input, so U is not diagonal. However, there still exists an input that diagonalizes the GDFE.

The examples emphasize that 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.

5.6 Asymptotic Stationary Convergence of certain GDFEs

Sections 5.4 and 5.5 observe that the CDFE with nonsingular, Toeplitz, circulant P and stationary (circulant also) $R_{\boldsymbol{xx}}$ is very special, and appeared in all examples to converge to the MMSE-DFE of Chapter 3 as $N \to \infty$. This section investigates such convergence further. 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 - Subection 5.6.1
2. linear prediction and canonical factorization - Subsection 5.6.2
3. Use of 1 and 2 in the GDFE - Subsection 5.6.3.

5.6.1 Stationary Channels

Strictly speaking, for any finite N, random vectors like the $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{n}$ of the transmission channel $\boldsymbol{y} = P\boldsymbol{x} + \boldsymbol{n}$ are never stationary[28]. When these vectors are considered in succession, if the same autocorrelation matrix for each occurs in every successive block or packet, the vector process can be construed as **block stationary**, which actually is an example of the sample random processes that correspond to the time-indexed elements of these successive random vectors being **cyclostationary** with period equal to $N + \nu$, the dimension of the vectors. Furthermore, singular random processes with less than full rank autocorrelation matrices should not be considered block or cyclo-stationary unless the singularity has been removed as in Section 5.3. The GDFE attains an SNR that is canonical in all cases for the block-stationary case, and the data rate of $\bar{I}$ can only be defined (and thus be achieved) if the transmission system $(R_{\boldsymbol{xx}}, P, R_{\boldsymbol{nn}})$ is block stationary. Strictly speakin, the GDFE only becomes stationary as $N \to \infty$.

However, for the ISI-specific situation investigated throughout Chapters 3 and 4 and often in Chapter 5, another additional type of stationarity is implied. The random processes x_k, y_k, and n_k are presumed stationary. In this case only, additional structure occurs within the GDFE that can lead (when all singularity is removed) to (a set of) stationary DFE(s). In this case, Toeplitz autocorrelation matrices for all processes can occur at each and every block length N (or $N + \nu$), and convergence to fixed structures as $N \to \infty$. This requires care in structuring the successive processing of symbols (or blocks) even when the underlying channel and its input are stationary.

For instance, the FIR channel $P(D)$ 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 $1 + .9D^{-1}$ channel with $\nu = 1$ that has been often investigated in these notes. For this channel and the usual guard period used in VC or the GDFE with a Toeplitz $N \times (N + \nu)$ convolution P matrix does not produce a Toeplitz $R_{\boldsymbol{yy}}$. For instance with $N = 2$ and $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$,

$$R_f = \begin{bmatrix} .81 & .9 & 0 \\ .9 & 1.81 & .9 \\ 0 & .9 & 1 \end{bmatrix} + \sigma^2 I \quad , \tag{5.328}$$

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 N along the way. The repeated insertion of the guard period thus essentially destroys the stationarity that is otherwise naturally present.

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

$$R_{\boldsymbol{yy}} = \begin{bmatrix} 1.81 & .9 \\ .9 & 1.81 \end{bmatrix} + \sigma^2 I \quad , \tag{5.329}$$

which is Toeplitz. Indeed, a cyclic prefix of length ν samples when the channel (or TEQ equalized channel) has length of ν or less will always produce a stationary $R_{\boldsymbol{yy}} = P^* R_{\boldsymbol{xx}} P + R_{\boldsymbol{nn}}$ as long as

[28] The term "stationary" in this chapter means "wide-sense stationarity," and so only means, covariances need be invariant to time

$R_{\boldsymbol{nn}}$ is Toeplitz, and $R_{\boldsymbol{xx}}$ is not only Toeplitz, but cyclic also. [29] Clearly, the cyclic prefix insures the Toeplitz nature of the signal part in $R_{\boldsymbol{yy}}$.[30] Thus, the cyclic prefix has an additional benefit in asymptotic analysis in that Toeplitz matrices will appear nearly throughout the GDFE problem for all N, leading to various Cholesky factorizations appearing to converge for smaller N than would otherwise be necessary. This was clearly evident in early examples in that the CDFE appeared well-behaved as $N \to \infty$ while a more general GDFE appeared to have no easy convergence evident (at the N values used in previous examples).

With multiple transmission bands ($M > 1$), the matrix $\tilde{P} = PA$ of Section 5.5 leads to a $\tilde{P}^*\tilde{P}$ 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 matrix, which is not Toeplitz in the multiband case.

5.6.2 Canonical Factorization for finite- and infinite-length stationary sequences

While Cholesky factorization of a matrix and canonical factorization of an autocorrelation function (or in both cases those that satisfy the Paley Wiener criterion) are clearly related, this chapter now attempts to formalize this relation. This formalization can occur through investigation of linear prediction. From Section 5.1, Figure 5.1 for sequences with stationary autocorrelation $R_{xx}(D)$ that thus satisfy the PW criterion,

$$x(D) = G_x(D) \cdot v(D) \tag{5.330}$$

where

$$R_{xx}(D) = G_x(D) \cdot S_x \cdot G_x^*(D^{-*}) \tag{5.331}$$

and $S_x = E|v_k|^2$ is the MMSE associated with linear prediction of x_k from its past x_{k-1}, ..., $x_{-\infty}$. $G_x(D)$ is causal, monic, and minimum-phase. The linear prediction filter is $1/G_x(D)$, and is also minimum phase, causal, monic, and realizable. The input v_k carries the information of x_k as a white sequence. Canonical factorization for a vector $\boldsymbol{x}$ with Toeplitz autocorrelation matrix $R_{\boldsymbol{xx}}$ is

$$R_{\boldsymbol{xx}} = G_x R_{\boldsymbol{vv}} G_x^* \quad , \tag{5.332}$$

where G_x is upper triangular (causal), monic , and nonsingular (i.e., invertible to upper-triangular-causal inverse).[31]

Cholesky factorization of a stationary sequence is also intimately related to linear prediction. In the finite-length case, the sequence v_k is the MMSE sequence corresponding to estimating x_k from its past samples x_{k-1} ,..., x_0, which follows directly from forming the m^{th}-order linear-prediction error

$$v_k = x_k - g_1 x_{k-1} - ... - g_{k-1} x_0 \quad , \tag{5.333}$$

and from noting via the orthogonality criterion

$$E\left[v_k x_{k-i}^*\right] = 0 \; \forall \; i = 1, ..., k-1 \quad , \tag{5.334}$$

when the g_i's are the entries from the corresponding row of Cholesky factorization with subscript indices that are changed to make the appearance more like a stationary convolutional filter.[32] Clearly then through the linear-prediction problem

$$\lim_{N \to \infty} \text{row}(G_x) \to G_x(D) \quad . \tag{5.335}$$

[29] The product of Hermetian cyclic Toeplitz matrices can be shown to be Hermetian cylic Toeplitz - hint, think circular convolution.

[30] In fact, any noise whitening by cyclic $R_{\boldsymbol{nn}}^{-1/2}$ that is done for equivalent channels needs then also for $R_{\boldsymbol{nn}}$ and $R_{\boldsymbol{nn}}^{-1/2}$ to be cyclic – cyclic $R_{\boldsymbol{nn}}$ 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 $R_{\boldsymbol{nn}}$ 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 $R_{\boldsymbol{nn}}$.

[31] The entries in the rows of G_x need not be minimum phase, but will tend to such as $N \to infty$.

[32] The statisfaction of (5.334) follows directly from $R_{\boldsymbol{vv}}$ being diagonal and thus v_{k-1} ... v_0 or $\boldsymbol{v}_{k-1}$ are all uncorrelated with v_k and thus so must also be x_{k-1} ... x_0 or $\boldsymbol{x}_{k-1}$ since $\boldsymbol{x}_{k-1} = G_x(k-1)\boldsymbol{v}_{k-1}$ from Cholesky factorization of one order lower. Note the non-singularity of $R_{\boldsymbol{xx}}$ ensures Cholesky factorization of all orders.

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

The diagonal matrix $R_{\boldsymbol{vv}}$ must also become constant as $N \to \infty$ because the diagonal values represent the single constant MMSE value

$$\lim_{N\to\infty} R_{\boldsymbol{vv}}(i) = E|v_k|^2 = S_x \quad \forall\, i \geq 0 \quad .^{33} \tag{5.336}$$

This result does not hold for a singular process – that means input singularity must be eliminated, or equivalently the PW must be satisfied. When a water-filling or any other circulant $R_{\boldsymbol{xx}}$ has sets of nonzero DFT values that are separated by zeroed values, the consequent singularity must be eliminated as in Section 5.5.

Then a separate Cholesky filter, each acting on the independent inputs of each band, corresponds to and ultimately converges to each linear prediction filter for each independent nonsingular band. Canonical factorization across the whole band does not work for finite length nor for infinite-length when water-filling produces unused bands. In this case, there must be multiple bands.

5.6.3 Convergence of the Canonical Channel Models and the PDFE

The CDFE must be separated into the M distinct bands of the block triangular G_x, creating a set of $R_f(m)$ for $m = 1, ..., M$ in this section when $M > 1$. The following results will not hold otherwise. The matrix R_f of each band of the CDFE is Toeplitz when $R_{\boldsymbol{nn}}$ is Toeplitz, and P and $R_{\boldsymbol{xx}}$ are circulant as would occur in most higher-performance designs. If $R_{\boldsymbol{nn}}$ is circulant, then so is R_f. Then $R_b^{-1} = R_f + R_{\boldsymbol{vv}}^{-1}$ will approach a Circulant matrix as $N \to \infty$. R_b^{-1}'s only deviation from circulant structure is the non-constant diagonal of $R_{\boldsymbol{vv}}$ that becomes constant as N is large. In fact for most water-filling designs, the diagonal entries of $R_{\boldsymbol{vv}}$ are quite close to constant for relatively small N, because water-filling is often close to spreading equal energy across all the dimensions that are worthwhile using, at least within each band.

The Cholesky factorization of

$$R_b^{-1} = G^* S_0 G \tag{5.337}$$

corresponds to a inverse-order prediction problem (sometimes called reverse or "backward" linear prediction), or equivalently

$$R_b = G^{-1} S_0^{-1} G^{-*} \tag{5.338}$$

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 samples. 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 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

$$\lim_{N\to\infty} \text{ any row of } G = G(D) \quad . \tag{5.339}$$

The MMSE for this factorization must also then converge to a constant so that

$$\lim_{N\to\infty} S_0(i) = s_0 \quad \forall\, i \quad . \tag{5.340}$$

The diagonal entries of the matrix S_0^{-1} also represent the MMSE associated with the GDFE for each time instant, and 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

$$s_0^{-1} = \frac{\frac{\mathcal{N}_0}{2}}{\gamma_0 \cdot \|p\|^2} \tag{5.341}$$

[33] 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.

or

$$s_0 = \frac{\gamma_0 \cdot \|p\|^2}{\frac{\mathcal{N}_0}{2}} \quad . \tag{5.342}$$

The feedforward matrix $S_0^{-1}G^{-*}$ will be thus be anticausal (lower triangular) as $N \to \infty$. The matched filter P^* can be mixed phase, and already has circulant structure directly so is effectively 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 SNR for the CDFE then converges in each band to

$$\begin{aligned}
\lim_{N\to\infty} \text{SNR}_{cdfe} &= \lim_{N\to\infty} \left[|R_{\boldsymbol{vv}}| \cdot |S_0|\right]^{N+\nu} && (5.343)\\
&= \frac{S_x}{s_0} && (5.344)\\
&= \frac{\bar{\mathcal{E}}_{\boldsymbol{x}} \cdot \gamma_0 \cdot \|p\|^2}{\frac{\mathcal{N}_0}{2}} && (5.345)\\
&= \gamma_0 \cdot \text{SNR}_{\text{mfb}} && (5.346)\\
&= 2^{2\bar{I}(X(D);Y(D)} && (5.347)
\end{aligned}$$

The unbiased SNR is as always $\text{SNR}_{cdfe,u} \to \text{SNR}_{\text{mmse-dfe,u}} = \text{SNR}_{\text{mmse-dfe}} - 1$. An overall SNR can be defined based on fraction of bandwidth use of each of the DFEs as in Section 5.2.

Some final comments are in order:

1. All asymptotic results hold ν as a constant (correctly as it is a constant). The corresponding "wasted samples" and any energy "wasted" in the cyclic prefix clearly then go to zero as block length N increases.

2. Error propagation is limited to within each block in all GDFE's, including the CDFE. Shorter block's can thus limit error bursts, while the MMSE-DFE has a potential for infinite catastrophic bursts.

3. A designer could pick N too large, much larger than is necessary to obtain 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 made in this same good-design context. Obviously, a the channel $P(D) = 1$ 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 $3\log_2(N)$ ignoring modulation or interpolation complexity unique to CDFE, and about $5\log_2(N)$ if those are included.

4. Any gap greater than 0 dB will lead 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 maximizing product SNR at all gaps. Beware of certain statements that state the opposite as an "SNR averaging" property of the DFE – correct use of codes and gaps reverses and such argument (and it is errors in the use of the codes and gaps that leads to the misinformed making such high-level arguments.

5. DMT is a maximum likelihood detector and thus canonical and optimal. The CDFE is only canonical and does not have ML performance of the detector, and as well may have then error propagation problems.

6. If modulation and interpolation is carefully implemented (not shown in these notes), it may be possible to reduce the delay from input to output in a CDFE design with respect to DMT.

7. OFDM does not achieve the performance levels of DMT nor CDFE. However, it has been observed that will a very-well-designed "robust" code, it is possible for an OFDM system to obtain the SNR of $2^{2\bar{I}} - 1$ with gap 0 dB for the one special case of flat transmission energy. 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, possibly through many simulataneous different channels to different users) or because the channel changes so fast that the input dynamic spectra could not follow the channel. Thus, a fixed flat spectrum has a certain $\bar{I}$ 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 code for OFDM may be different than the code for the MMSE-DFE. Thus, in time-varying channels, there can be no difference between a flat-input-coded OFDM system and a flat-input-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). Time variation is still an area of open research.

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 $\boldsymbol{R_{xx}}$ that is common to DMT and CDFE) that is a function of the channel and thus must be implemented adaptively.

9. CDFE systems exhibit the same "Gaussian" like high peak-to-average ratio of DMT systems (contrary to misinformed popular belief), this is particularly true as ISI becomes severe, or equivalently large N is required. Simply put, this is a function of the "central limit theorem" and the transmit signal being simply a sum of many contributions of independent random variables, 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).

10. Adaptive training of DMT and CDFE systems requires some kind of "loading" algorithm to determine good use of transmit spectra. While this training is well studied for DMT, it is an open area for CDFE presently unless one has a DMT system behind it that simply supplies the information for training the CDFE to the procedures given in this chapter. "Blind" training (i.e., with no feedback from receiver to transmitter loses performance in the case where the initially presumed spectrum is not correct. Beware of statements that DFE systems can be trained blindly from the misinformed who do not understand the effect and need for transmit optimization for best performance.

11. The matched-notching ability of the MS-WMF of the MMSE-DFE that occurs when the noise-equivalent ISI channel has notches is absolutely NOT equivalent to the use of multiple bands. This statement is found throughout the literature on MMSE-DFEs and is simply incorrect every single time it has been said, even if that is often and by many. A single MMSE-DFE will perform worse than multiple MMSE-DFEs on each side of all the notches - no exceptions.

Ideal MMSE-DFE is lower bound for MLSD Performance

The ideal MMSE-DFE (or ideal MMSE-GDFE) upper bounds the performance of a maximum likelihood detector in that the probability of a sequence error is always lower for MLSD. Before accepting this result trivially, one needs to recall that the ideal MMSE-DFE assumes 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 could outperform the optimum detector because of the potentially false presumption of no decision errors.

However, the bound does hold: The probability of sequence error for the ideal DFE case is actually one because the probability of a symbol error is finite non-zero (in other words, the gap is non zero),

so that there is a certainty of at least one error made in an infinite sequence of such symbol-by-symbol decisions. By contrast, the probability of sequence error for MLSD can be made arbitarily small and nonzero if the code is well chosen at rates less than capacity. Even with poor codes, the MLSD probability of error cannot exceed one. Thus, MLSD has a lower probability of sequence error than an ideal DFE whenever the probability of symbol error in the ideal DFE is non-zero. A limiting argument would find that the ideal DFE probability of error is lower bounded by MLSD as the probability of symbol error of the ideal DFE is lowered gradually to zero because if there are no errors in reality, then it is still a suboptimum structure relative to MLSD.

This result is somewhat profound also, albeit it again trivial. The result implies that a receiver design that follows the MS-WMF(matrix) of a MMSE-GDFE with a sequence detector for the consequent equivalent channel output will can perform at least as well as an ideal MMSE-GDFE. Thus, any loss caused by error propagation can be ignored for capacity purposes in that the GDFE feedback section can be replaced by infinite complexity receiver that will at least attain the canonical ideal MMSE-GDFE performance level, albeit with infinite decoding delay. This section essentially establishes that in the limit, there is no loss from capacity by using the feedforward part of a GDFE, a good code, and maximum likelihood detector for the channel shaped by the GDFE feedforward matrix/filter.

In short, Claude Shannon had it right as with so many other of his basic simple and information-age-envisioning results – a multicarrier system is optimum for handling ISI. And, as we see in the last two chapters, quite practical and efficient as well with modern digital signal processing. Wide-band QAM or PAM can be saved to a canonical level, but this is unnecessarily complex – these systems exist as a legacy of analog processing methods of decades past where high precision digital signal processing was not available to efficiently implement what Shannon originally conceived as best. In time, as this legacy dwindles, the need for the wide-band system theories and designs will diminish also as more educated engineers proceed to good designs initially when faced with bandlimited channels.

5.7 Block infinite channels and results

The channel matrix P has been arbitrary in this chapter, but much of the theory for the equivalence conditions for CDFE and DMT have focused on the situation in which P models the time-domain convolution of the input samples x_k with a finite-length channel pulse response p_k. However, at times P has represented a multiple-input-multiple-output transmission system (wired or wireless) for which the matrix P is more arbitrary and has in general no "sample-stationary" nor even convolutional stucture. However, all GDFE theory has assumed independent packet transmissions through the channel P – that is no intersymbol interference between "packets" (in the GDFE theory, the packets are symbols).

This section generalizes the case to possible block-channel memory in forming a block-stationary matrix channel pulse response with D-transform:

$$\boldsymbol{P}(D) = P_0 + P_1 \cdot D + P_2 \cdot D^2 + ... + P_{\tilde{\nu}} \cdot D^{\tilde{\nu}} + ... \ , \tag{5.348}$$

where $\tilde{\nu}$ is now used in a block or symbol sense, but is analogous to the ν parameter used to represent the channel length of the time-domain pulse response on single-input-single-output channel. However, $P(D)$ can be infinite in length also if necessary and none of the following theory in this Section changes. Fundamentally, $\tilde{\nu}$ represents a response length in symbol periods. This type of model, for instance, can allow the omission of guard-bands between blocks and the associated excess-dimensionality loss. In most transmission systems without a guard band and some pulse response wiht $N = \nu + 1$ and $\tilde{nu} = 1$,

$$\boldsymbol{P}(D) = P_0 + P_1 \cdot D \tag{5.349}$$

where

$$P_0 = \begin{bmatrix} p_0 & p_1 & ... & p_\nu \\ 0 & p_0 & ... & p_\nu \\ \vdots & \ddots & \ddots & \vdots \\ 0 & ... & 0 & p_0 \end{bmatrix} \tag{5.350}$$

and

$$P_1 = \begin{bmatrix} 0 & ... & 0 & 0 & & \\ p_\nu & 0 & ... & 0 & & \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ p_1 & ... & p_\nu & & & \end{bmatrix} \ . \tag{5.351}$$

Other block lengths, ν's , and structure can be imposed on P_i, depending on the particulars of the application. For wireless MIMO, the $\boldsymbol{P}(D)$ structure allows memory within the channel or intersymbol interference to be modelled along with crosstalk. For a cable of wires, it can represent the matrix channel's memory and the memory of the crosstalk between the lines. Vector D transforms can be associated with the channel input, channel noise, and channel output to produce

$$\boldsymbol{y}(D) = \boldsymbol{P}(D) \cdot \boldsymbol{x}(D) + \boldsymbol{n}(D) \quad . \tag{5.352}$$

Conceptually, all the concepts previously introduced generalize so that in particular

- **singularity elimination** (channel and input) $A \rightarrow A(D)$.
- **equivalent nonsingular channel** $\tilde{P} \rightarrow \tilde{P}(D) = R_{\boldsymbol{nn}}^{-1/2}(D) \cdot \boldsymbol{P}(D) \cdot A(D)$.
- **forward canonical channel matrix** $R_f \rightarrow R_f(D) = \tilde{P}^*(D^{-*})\tilde{P}(D)$.
- **white input** $R_{\boldsymbol{vv}} \rightarrow R_{\boldsymbol{vv}}$ (no D)
- **backward canonical channel matrix** $R_b^{-1} \rightarrow R_b^{-1}(D) = R_f(D) + R_{\boldsymbol{vv}}^{-1}$

Singularity elimination requires that $\tilde{P}(D)$ be $N \times N^*$ where $N^* \leq \tilde{N} \leq N$ defines the number of non-zero energy input dimensions that pass to the channel output. Equivalently $R_f(D)$ is invertible at all frequencies $D = e^{-\jmath\omega T}$ except possible a set of measure zero (i.e., notches are allowed, but no dead-bands

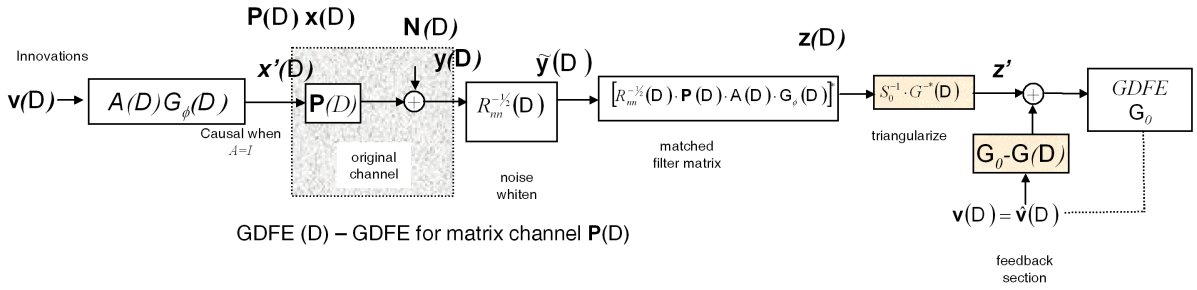


Figure 5.35: GDFE for channel $\boldsymbol{P}(D)$.

in frequency or dead-spatial zones). Mutual information is maintained $\bar{I}(\boldsymbol{x}(D);\boldsymbol{y}(D)) = \bar{I}(\boldsymbol{u}(D);\boldsymbol{z}(D))$ and performance is canonical for any derived GDFE's, which are now a function of D,

$$\begin{aligned} G &\rightarrow G(D) && (5.353)\\ S_0 &\rightarrow S_0 && (5.354)\\ W &\rightarrow W(D) = S_0^{-1}G^{-*}(D^{-*}) && (5.355)\end{aligned}$$

The GDFE then generalizes to the structure of Figure 5.35. However, Cholesky factorization, and also any singular-value or eigenvector decompositions, need re-interpretation.

Definition 5.7.1 (Smith-Canonical Form) *The generalization of Cholesky factorization of any positive definite symmetric ($R(D) = R^*(D^{-*})$) is known in the mathematical and control literature as* **Smith Canonical Factorization**, *a special form of which is guaranteed to exist and provide*

$$R(D) = G(D)\cdot S\cdot G^*(D^{-*}) \tag{5.356}$$

where $G(D) = G_0 + G_1\cdot D + ... + G_{\tilde{\nu}}D^{\nu} + ...$ is monic volume preserving(G_0 has ones on diagonal and $|G(D)| = 1$), block causal, and minimum phase (all roots of $|G(D)| = 0$ are outside or on unit circle, and G_0 is triangular (upper or lower as appropriate). S is positive-definite diagonal matrix with constant entries (no D). For the factorization to exist, the PW criterion generalizes to

$$\int_{-\pi}^{\pi} |\ln R(e^{-\jmath\omega T})| d\omega < \infty \quad . \tag{5.357}$$

and must be satisfied, wher $|\bullet|$ denotes determinant and $\ln(matrix)$ *is defined in the polynomial Taylor series sense.*

The smith-canonical (Cholesky Factorization) form has known algorithms that produce it (see Chapter 10 appendices) symbolically for polynomial $R(D)$. Conceptually, Cholesky factorization is used for the transmitted input in two cases:

1. input triangular factorization $R_{\boldsymbol{uu}} = G_u(D)R_{\boldsymbol{vv}}G_u(D^{-*}$ where $G_{u,0}$ is monic upper triangular
2. and for the feedback section of the GDFE $R_b^{-1}(D) = G^*(D^{-*})\cdot S_0\cdot G(D)$ where G_0 is monic upper triangular.

Eigenvector and/or singular-value decompositions follow directly the case in which $P(D) = P_0$ but must be executed (conceptually) for each and every value of D at least on the unit circle. As far as this author has been able to determine, there is no body of mathematics ever devoted to the concept of such matrix-polynomial eigen-vector or singular-value factorization. Even for any fixed value of D, such factorizations require iterative approximate algorithms for computation, so there is certainly no

closed form means for executing the generalized forms as a function of D. A means for approximating such factorizations is to execute them for equally spaced values (say 100 or so) around the unit circle $D = e^{-j\cdot(i/100)\cdot\pi}$ for $i = -100, ..., 0, ..., 100$. When the resulting 201 matrix factorizations are stored, one can then take the IFFT of each entry of such matrices to get the time-domain equivalent response at the packet-sampling interval and construct the various GDFE matrices. A lot of design work for the designer, but a means for approximating matrix polynomial series.

5.7.1 The Finite Block-Length Stationary MMSE-DFE for packets

The above involved polynomial factorizations can be simplified considerably and the GDFE well approximated by in some sense returning to Chapter 3's MMSE-DFE, but this time for packets. Such a structure presumes a block-stationary nature of the channel so that P_i are all constant matrices, and also a nonsingular input with no energy in the null space of the channel. In the matrix channel case, these may be difficult assumptions to postulate in practice.

The vector of inputs at packet time k to the feedforward matrix filter is $\boldsymbol{Y}_k$, now a vector of N-dimensional packet symbols, thus $N_f \cdot N$ in length. and the corresponding vector of filter coeficients is $\boldsymbol{W}$, so the feedforward filter output is $\boldsymbol{Z}_k = \boldsymbol{W}^*\boldsymbol{Y}_k$, which attempts to assist a block feedback section in estimating the packet symbol $\boldsymbol{X}_{k-\Delta}$. The DFE error signal is then described by

$$\boldsymbol{E}_k = \boldsymbol{B}\boldsymbol{X}_k(\Delta) - \boldsymbol{W}\boldsymbol{Y}_k \quad , \tag{5.358}$$

$\boldsymbol{B}$ is monic and volume preserving. A GDFE for the monic triangular feedback section of B_0 is independently still executed to obtain each packet decision.

$$\boldsymbol{B} \triangleq [B_0\ B_1\ B_2\ ...\ B_{N_b}] \quad , \tag{5.359}$$

where B_0 is triangular (and monic). The mean-square (trace) of the Error-autocorrelation matrix trace($R_{\boldsymbol{E}\boldsymbol{E}}$) is to be minimized over $\boldsymbol{B}$ and $\boldsymbol{W}$. Signals will again be presumed to be complex, but all developments here simplify to the one-dimensional real case directly, although it is important to remember to divide any complex signal's variance by 2 to get the energy per real dimension. The SNR equals $\bar{\mathcal{E}}_{\boldsymbol{x}}/\frac{\mathcal{N}_0}{2} = \mathcal{E}_{\boldsymbol{x}}/\mathcal{N}_0$ in either case.

Optimizing the feedforward and feedback filters For any fixed $\boldsymbol{B}$ in (5.358), the cross-correlation between the error and the vector of channel outputs $\boldsymbol{Y}_k$ should be zero to minimize MSE,

$$E\left[\boldsymbol{E}_k\boldsymbol{Y}_k^*\right] = 0 \quad . \tag{5.360}$$

So,

$$\boldsymbol{W}^*\bar{R}_{\boldsymbol{Y}\boldsymbol{Y}} = \boldsymbol{B}^*\bar{R}_{\boldsymbol{X}\boldsymbol{Y}}(\Delta) \quad , \tag{5.361}$$

and

$$\begin{aligned} \bar{R}_{\boldsymbol{X}\boldsymbol{Y}}(\Delta) &= E\left\{\boldsymbol{X}_k(\Delta)\left[\boldsymbol{x}_k^*\ \boldsymbol{x}_{k-1}^*\ ...\boldsymbol{x}_{k-N_f-\nu+1}^*\right]\boldsymbol{P}^*\right\} & (5.362)\\ &= R_{\boldsymbol{v}\boldsymbol{v}}J_\Delta^*\boldsymbol{P}^* \quad , & (5.363)\end{aligned}$$

where J_Δ^* is a matrix with an $N \times N$ Identity in the Δ^{th} position and zeros elsewhere. The MSE for this fixed value of $\boldsymbol{B}$ then becomes

$$\begin{aligned} R_{\boldsymbol{E}\boldsymbol{E}} &= \boldsymbol{B}^*\left(\bar{\mathcal{E}}_{\boldsymbol{x}}I - \bar{R}_{\boldsymbol{X}\boldsymbol{Y}}(\Delta)\bar{R}_{\boldsymbol{Y}\boldsymbol{Y}}^{-1}\bar{R}_{\boldsymbol{Y}\boldsymbol{X}}(\Delta)\right)\boldsymbol{B} & (5.364)\\ &= \boldsymbol{B}^*\bar{R}_{\boldsymbol{X}/\boldsymbol{Y}}^{\perp}(\Delta)\boldsymbol{B} & (5.365)\\ &= R_{\boldsymbol{v}\boldsymbol{v}}\cdot\boldsymbol{B}^*\left\{I_{N_b\cdot N} - J_\Delta^*\boldsymbol{P}^*\left(\boldsymbol{P}\boldsymbol{P}^* + \frac{l}{\mathrm{SNR}}\cdot\bar{R}_{\boldsymbol{N}\boldsymbol{N}}\right)^{-1}\boldsymbol{P}J_\Delta\right\}\boldsymbol{B} & (5.366)\\ &= l\cdot\frac{\mathcal{N}_0}{2}\left\{\boldsymbol{B}^*J_\Delta^*\left(\boldsymbol{P}^*\boldsymbol{P} + \frac{l}{\mathrm{SNR}}\cdot\bar{R}_{\boldsymbol{N}\boldsymbol{N}}\right)^{-1}J_\Delta\boldsymbol{B}\right\} \quad , & (5.367)\end{aligned}$$

and $\bar{R}^{\perp}_{\boldsymbol{X}/\boldsymbol{Y}}(\Delta) \triangleq \bar{\mathcal{E}}_{\boldsymbol{x}} I - \bar{R}_{\boldsymbol{XY}}(\Delta)\bar{R}^{-1}_{\boldsymbol{YY}}\bar{R}_{\boldsymbol{YX}}(\Delta)$, the autocorrelation matrix corresponding to the MMSE vector in estimating $\boldsymbol{X}(\Delta)$ from $\boldsymbol{Y}_k$. That is, $\bar{R}^{\perp}_{\boldsymbol{X}/\boldsymbol{Y}}$ is the autocorrelation matrix for the error sequence of length N_b that is associated with a "matrix" MMSE-LE. The solution then requires factorization of the inner matrix in to canonical factors, which is executed with Choleksy factorization.

By defining

$$\tilde{\boldsymbol{Q}}(\Delta) = \left\{ J^*_\Delta \left(\boldsymbol{P}^*\boldsymbol{P} + \frac{l}{\mathrm{SNR}} I \right)^{-1} J_\Delta \right\}^{-1} \quad . \tag{5.368}$$

Cholesky factorization of $\tilde{\boldsymbol{Q}}$ proceeds according to

$$\bar{\mathcal{E}}_{\boldsymbol{x}} \tilde{\boldsymbol{Q}} = G^*_\Delta S^{-1}_\Delta G_\Delta \quad , \tag{5.369}$$

where G_Δ is an upper triangular (causal) and monic (ones down the diagonal) matrix and $S_\Delta = diag\{S_0(\Delta)\ S_1(\Delta)\ ...\ S_{N_b}(\Delta)\}$ is a diagonal matrix with smallest element in the upper left corner and positive, real, non-decreasing values down the diagonal. The MSE then is

$$R_{\boldsymbol{EE}} = \boldsymbol{B} G^{-1}_\Delta S_\Delta G^{-*}_\Delta \boldsymbol{B}^* \quad , \tag{5.370}$$

which is minimized when

$$\boldsymbol{B} = \boldsymbol{G}(\Delta) \quad , \tag{5.371}$$

the top row of the upper-triangular matrix G_Δ. The MMSE is thus obtained by computing Cholesky factorizations of $\tilde{\boldsymbol{Q}}$ for all reasonable values of Δ and then setting $\boldsymbol{B} = \boldsymbol{G}(\Delta)$. Then

$$R_{\boldsymbol{vv}} = S_0(\Delta) \quad . \tag{5.372}$$

The packet decision device becomes a within-packet feedback section determined by $Gvec_0(\Delta) = B_0$ (the term with the monic diagonal).

The feedforward filter then becomes

$$\begin{aligned} \boldsymbol{W} &= \boldsymbol{b}\bar{R}_{\boldsymbol{XY}}(\Delta)\bar{R}^{-1}_{\boldsymbol{YY}} && (5.373)\\ &= \boldsymbol{G}(\Delta) J^*_\Delta \boldsymbol{P}^* \left(\boldsymbol{P}\boldsymbol{P}^* + \frac{l}{\mathrm{SNR}} I \right)^{-1} && (5.374)\\ &= \underbrace{\boldsymbol{g} J^*_\Delta \left(\boldsymbol{P}^*\boldsymbol{P} + \frac{l}{\mathrm{SNR}_{\mathrm{mfb}}} I \right)^{-1}}_{\text{feedforward filter}} \underbrace{\boldsymbol{P}^*}_{\text{matched filter}} \quad . && (5.375) \end{aligned}$$

The SNR is

$$\mathrm{SNR}_{\text{mmse-dfe,u}} = |R_{\boldsymbol{vv}} S_0^{-1}(\Delta)|^{1/N} - 1 \quad . \tag{5.376}$$

Bias can be removed in the usual manner.

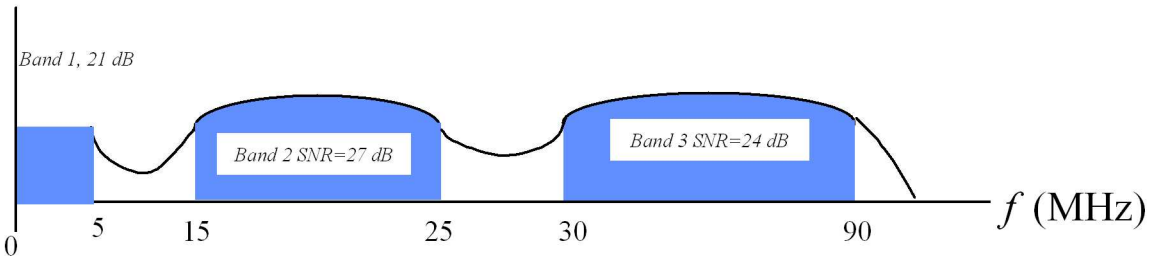


Figure 5.36: Multiband transmission for Problem 5.3

Exercises - Chapter 5

5.1 *Mutual Information and Parallel Independent AWGN Channels*
Consider a set of parallel independent one-dimensional AWGN channels:

a. (2pts) Show that the mutual information for the set of channels is the sum of the mutual information quantities for the set.

b. (3 pts) If the set of parallel channels has a total energy constraint that is equal to the sum of the energy constraints, what energy $\mathcal{E}_n$, $n = 1, ..., N$ should be allocated to each of the channels to maximize the mutual information. You may presume the subchannel gains are given as g_n (so that the individual SNRs would then be $\mathcal{E}_n \cdot g_n$).

c. (2 pts) Find the overall SNR for a single AWGN that is equivalent to the set of channels in terms of mutual information.

5.2 *Innovations*
Find the innovations variance per dimension, entropy and linear prediction filter for the following Gaussian processes (let $T = 1$):

a. (2 pts) A real process with autocorrelation $R_{xx}(D) = .0619 \cdot D^{-2} + .4691 \cdot D^{-1} + 1 + .4691 \cdot D + .0619 \cdot D^2$.

b. (2 pts) A complex discrete-time process with power spectral density $10\,(1.65 + 1.6 \cdot cos(\omega))$.

c. (2 pts) A complex process that has a maximum of only 3 nonzero terms in its autocorrelation function, a notch exactly in the middle of the band (i.e., at normalized frequency 1/4) and total power 2.

5.3 *Multibands*
Three bands of transmission result from infinite-length water-filling as shown below with a gap of 5.8 dB at $P_e = 10^{-6}$. The lowest band uses baseband PAM transmission, and the other two bands use QAM transmission. Each band has an MMSE-DFE receiver with the SNR shown in Figure 5.36.

a. (1 pt) Find the optimum symbol rates $1/T_i^*$ and carrier frequencies $f_{c,i}^*$ for each of the 3 bands.

b. (2 pts) Find b_1, b_2, and b_3 for these 3 bands and the data rate R.

c. (1 pt) Find the overall best SNR and $\bar{b}$ for a single AWGN that is equivalent to the set of channels in terms of mutual information.

d. (1 pt) The noise is such that $\bar{\mathcal{E}}_{\boldsymbol{x}} = -70$ dBm/Hz in all bands. What is the total energy used $\mathcal{E}_x$ =?

Suppose baseband PAM with a symbol rate of $1/T$ =200 MHz is used instead with total energy/power equally divided on all dimensions (i.e., successive PAM symbols are independent).

e. (.5 pt) What is $\bar{\mathcal{E}}_{x,PAM}$ =?

f. (1.5 pts) Find approximate $\text{SNR}_{mmse-dfe,PAM,u}$ and new data rate for this alternative single-band PAM design.

5.4 *CDEF*

An AWGN with intersymbol interference sustains successful transmission using a MMSE-DFE receiver with 256 QAM with Gap of $\Gamma = 8.8$ dB with a margin of 1.2 dB at $\bar{P}_e = 10^{-6}$.

a. (2 pts) What is the mutual information in bits per symbol for this channel? In bits per dimension?

b. (2 pts) Suppose optimization of the transmit filter increased the margin to 2.5 dB. What is the capacity of the channel at this symbol rate?

Suppose we return to the $1 + .9D^{-1}$ channel with $\text{SNR}_{\text{mfb}} = 10$ dB when $T = 1$ with PAM transmission.

c. (3 pts) At what new increased symbol energy per dimension would this channel have $T^* = 1$?

d. (2 pts) What is the capacity of this system in bits per symbol at symbol rate in part c with this new energy level?

e. (1 pt) Suppose this optimized system transmits with code with gap of 6 dB for probability of error 10^{-6} at $\bar{b} = 1$. What is margin at this data rate? (fractional bits per symbol are ok for inclusion in results and use of all formulas.)

5.5 *channel singularity elimination*

Given a binary antipodal white input with $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$ and the channel matrix

$$P = \begin{bmatrix} .9 & 1 & 0 & 0 \\ 0 & .9 & 1 & 0 \\ 0 & 0 & .9 & 1 \end{bmatrix} , \tag{5.377}$$

and with the choice of $C = I$,

a. (2 pts) Find the projection matrix $P_{\tilde{M}}$.

b. (2 pts) Determine the 4×3 matrix $\tilde{C}$ and the input $\tilde{\boldsymbol{u}}$ that characterize the pass-space description of this channel.

c. (2 pts) Determine the autocorrelation matrix $R_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}}$.

d. (2 pts) How much energy is lost into the null space of this channel?

5.6 *Singularity Elimination short problems.*

For each of the following channels and inputs, determine N, $\tilde{N}$, N^*, and ν.

a. (1 pt) $P = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$ and $R_{\boldsymbol{x}\boldsymbol{x}} = I$.

b. (1 pt) $P = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$ and $R_{\boldsymbol{x}\boldsymbol{x}} = I$.

c. (1 pt) $P = \begin{bmatrix} 5 & 6 & 7 & 8 \\ 1 & 3 & 2 & 4 \\ 4 & 3 & 5 & 4 \end{bmatrix}$ and $R_{\boldsymbol{xx}} = I$.

d. (1 pt) $P = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}$ and $R_{\boldsymbol{xx}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$.

e. (1 pt) $P = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{bmatrix}$ and $R_{\boldsymbol{xx}} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$.

f. (1 pt) $P = \begin{bmatrix} 5 & 6 & 7 & 8 \\ 1 & 3 & 2 & 4 \\ 4 & 3 & 5 & 4 \end{bmatrix}$ and $R_{\boldsymbol{xx}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$.

5.7 *MMSE Estimation*

Two zero-mean Gaussian random variables have probability distributions p_x and p_y with joint probability distribution $p_{x,y}$ and nonsingular autocorrelation matrix

$$R_{x,y} = \begin{bmatrix} \mathcal{E}_x & r_{xy} \\ r_{xy}^* & \mathcal{E}_y \end{bmatrix} \tag{5.378}$$

The minimum mean-square estimate of x given y has variance $\sigma_{x/y}^2$

a. (1 pt)

Find $\sigma_{x/y}^2$ in terms of $\mathcal{E}_x$, r_{xy} and $\mathcal{E}_y$.

b. (2 pts)

Find $p_{x/y}$ and show that it is also Gaussian with mean $[r_{xy}/(\mathcal{E}_y] \cdot y$ and variance $\sigma_{x/y}^2$.

c. (2 pts)

Relate the conditional entropy $H_{x/y}$ to $\sigma_{x/y}^2$ and therefore to the MMSE estimate.

d. (1 pt)

Show the results in parts a,b, and c are symmetric in x and y in that the two random variables can be interchanged.

e. (1 pt)

Now let $y = x + n$ where n is zero-mean complex Gaussian and independent of x with variance σ^2. Determine $H_{y/x}$.

f. (3 pts)

Show that $\text{SNR}_x = \frac{\mathcal{E}_x}{\sigma_{x/y}^2} = \text{SNR}_y = \frac{\mathcal{E}_y}{\sigma_{y/x}^2}$. Relate these two identical SNRs to the $\text{SNR} = \frac{\mathcal{E}_x}{\sigma^2}$.

g. (1 pt)

Relate the mutual information to SNR_x and to SNR.

h. (2 pts)

Suppose y becomes a complex vector $\boldsymbol{y}$ but x remains a scalar with $\boldsymbol{y} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} x + \boldsymbol{n}$ where $\boldsymbol{n}$ has independent zero-mean Gaussian components all with variance σ^2. Does your answer to part g change? How might you interpret SNR_x in this case in terms of equalizers you know from Chapter 3?

5.8 *DFE - MT connection.*

This problem explores the connection between DFE and multi-tone-like concepts for transmitter optimization.

a. (4 pts) Show that the following are all equivalent expressions for I. (QAM is assumed)

$$I = \frac{T}{2\pi} \int_{-\frac{\pi}{T}}^{\frac{\pi}{T}} \log_2(1 + SNR(w))dw = \frac{T}{2\pi} \int_{-\frac{\pi}{T}}^{\frac{\pi}{T}} \log_2(1 + SNR_{MFB} \left|Q(e^{jwT})\right|)dw$$

$$= \frac{T}{2\pi} \int_{-\frac{\pi}{T}}^{\frac{\pi}{T}} \log_2(1 + SNR \left|H(e^{jwT})\right|^2 \left|\Phi(e^{jwT})\right|^2)dw$$

where H is the transform of the sampled channel. Assume that the sampling rate is high enough so that there is no aliasing.

b. (4 pts) One can approximate the integral in part (a) by the following sum

$$\bar{I} = \frac{1}{2NT} \times T \sum_{-N+1}^{N} \frac{1}{2} \log_2(1 + SNR \left|H(e^{j2\pi n/2NT})\right|^2 \left|\Phi(e^{j2\pi n/2NT})\right|^2)$$

Do you see a transmitter optimization problem here? Show that the water-fill equations for this problem are :

$$\frac{1}{SNR \left|H(e^{j2\pi n/2NT})\right|^2} + \left|\Phi(e^{j2\pi n/2NT})\right|^2 = Constant$$

5.9 Input Singularity Elimination.

The channel input autocorrelation after elimination of components on the channel singularities is

$$R_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}} = \begin{bmatrix} 1 & .5 & 1 \\ .5 & 1 & .5 \\ 1 & .5 & 1 \end{bmatrix} \quad . \tag{5.379}$$

a. (1 pts) What is N^* for this channel input?

b. (2 pts) Find $\tilde{Q}$ and $P_{\tilde{Q}}$.

c. (4 pts) Find A and N^*-dimensional $\boldsymbol{u}$ such that $R_{\tilde{\boldsymbol{x}}\tilde{\boldsymbol{x}}} = AR_{\boldsymbol{u}\boldsymbol{u}}A^*$, and relate the components of $\boldsymbol{u}$ to those of $\tilde{\boldsymbol{x}}$.

5.10 Canonical Triangles

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 5.23.

a. (2 pts) The forward channel in Figure 5.23 has two right subtriangles embedded within it. From the one describing a decomposition of the noise vector $\boldsymbol{n}'$, one notes that the MMSE estimation filter for $\boldsymbol{n}'$ given $\boldsymbol{z}$ is $R_{\boldsymbol{n}'\boldsymbol{z}}R_{\boldsymbol{z}\boldsymbol{z}}^{-1} = (I - R_f R_b)$. Use this relation, and note that $R_{\boldsymbol{n}'\boldsymbol{z}} = R_{\boldsymbol{n}'\boldsymbol{n}'} = R_f$, to prove that $R_f^{-1} = R_b + R_{\boldsymbol{z}\boldsymbol{z}}^{-1}$.

b. (2 pts) Use the triangle for decomposition of the error vector in the backward triangle to note that $R_b = -R_{\boldsymbol{e}\boldsymbol{n}'} R_{\boldsymbol{n}'\boldsymbol{n}'}^{-1}$ to prove easily that $R_{\boldsymbol{e}\boldsymbol{e}} = R_b$.

c. (2 pts) Follow parts a and b to prove that $R_b^{-1} = R_f + R_{\boldsymbol{u}\boldsymbol{u}}^{-1}$.

d. (2 pts) Use a convenient triangle to show that $R_{\boldsymbol{e}\boldsymbol{n}'}{}^* = -R_f R_b$ and thus that $R_f R_b = (I - R_f R_{\boldsymbol{z}\boldsymbol{z}}^{-1})$.

e. (2 pts) Prove that $R_b R_f = (I - R_b R_{\boldsymbol{u}\boldsymbol{u}}^{-1})$.

f. (3 pts) The matrix SNR for a matrix MMSE-LE is $\boldsymbol{SNR} = R_{\boldsymbol{u}\boldsymbol{u}} R_{\boldsymbol{e}\boldsymbol{e}}^{-1}$. Use the Pythagorean relationship $R_{\boldsymbol{u}\boldsymbol{u}} = R_b R_{\boldsymbol{z}\boldsymbol{z}} R_b + R_{\boldsymbol{e}\boldsymbol{e}}$ to prove the "similar" triangles relationship $\boldsymbol{SNR}$ is also equal to $R_{\boldsymbol{n}'\boldsymbol{n}'}^{-1} R_{\boldsymbol{z}\boldsymbol{z}}$. Why is $I(\boldsymbol{x};\boldsymbol{y}) = I(\boldsymbol{u};\boldsymbol{z}) = \log_2 |\boldsymbol{SNR}|$?

5.11 Matrix Bias.

The relationships for bias in the matrix MMSE-LE and the GDFE generalize easily from their scalar counterparts. Let $\hat{\boldsymbol{u}} = R_b \boldsymbol{z}$ and $\boldsymbol{SNR} = R_{\boldsymbol{u}\boldsymbol{u}} R_{\boldsymbol{e}\boldsymbol{e}}^{-1}$.

a. (1 pt) Show that $E[\hat{\boldsymbol{u}}/\boldsymbol{u}] = R_b R_f \boldsymbol{u} \neq \boldsymbol{u}$ unless $u = 0$.

b. (1 pt) Show that $E[\boldsymbol{e}/\boldsymbol{u}] = (I - R_b R_f)\boldsymbol{u}$.

c. (2 pts) Show that $\boldsymbol{SNR}_u \triangleq R_{\hat{\boldsymbol{u}}\hat{\boldsymbol{u}}} R_{\boldsymbol{e}\boldsymbol{e}}^{-1}$ is equal to $\boldsymbol{SNR} - I$.

d. (2 pts) Show that another expression for the unbiased matrix SNR is $R_{\boldsymbol{n}'\boldsymbol{n}'}^{-1} R_{\hat{\boldsymbol{z}}\hat{\boldsymbol{z}}}$ where $\hat{\boldsymbol{z}} = R_f \boldsymbol{u}$.

e. (2 pts) Show that $I(\boldsymbol{u};\boldsymbol{z}) = \log_2 |\boldsymbol{SNR}|$ for complex signals and $I(\boldsymbol{u};\boldsymbol{z}) = \frac{1}{2}\log_2 |\boldsymbol{SNR}|$ for real signals. Explain the difference in formulae for real and complex channels.

5.12 Non-white inputs to GDFE.

In Example 5.3.3, assume that the binary PAM input on the $N + \nu = 3$ input dimensions is independent and one of the eight possible 3-dimensional vectors $\boldsymbol{x} = [\pm 1,\ \pm 1,\ \pm 1]$.

a. (4 pts) What are the eight corresponding $\boldsymbol{u}$?

b. (2 pts) Find $R_{\boldsymbol{u}\boldsymbol{u}}$.

c. (1 pt) What kind of detector would do best for the backward canonical channel created in terms of probability of error?

d. (2 pts) Compare $2^{2\bar{I}(\boldsymbol{x};\boldsymbol{y})}$ to SNR_{GDFE} for this $\boldsymbol{u}$. What can you say about probability of error if the original 2PAM $\boldsymbol{x}$ is used in this GDFE?

5.13 GDFE's and White Inputs.

A real linear-ISI channel with discrete-time response $1 + D$ and AWGN $\frac{\mathcal{N}_0}{2} = .001$ has initially 8PAM inputs that are independent. $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$.

a. (4 pts) For $N = 3$ and $\nu = 1$, find the CDFE on this channel, including G, feedforward filter matrix , $\text{SNR}_{cdfe,u}$ and $\bar{P}_e$ with equal-energy 16 PAM on each of the 3 used subchannels.

b. (2 pts) Compare the probability of error in part a with the asymptotic probability of error of an 8 PAM MMSE-DFE with $N_f = 3$ and $N_b = 1$ on this channel. Comment on why the latter appears better.

c. (4 pts) Now rework part a for a GDFE (with no cyclic prefix) that instead places 4/3 units of energy on each of the 3 useful dimensions of the input. 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 $\bar{P}_e = 10^{-6}$. Which system in this problem is best and why?

5.14 Comparing the GDFE and the MMSE-DFE.

For the $1 + .9D^{-1}$ with $SNR_{mfb} = 10$ 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 their old EE379A MMSE-DFE Matlab program (dfecolor3.m will be provided on the web).

Initially let $R_{\boldsymbol{xx}} = I$ and let N increase.

a. (5 pts) Write a matlab program to compute the GDFE performance for variable N and $\nu = 1$. Plot $SNR_{gdfe,u}$ versus N for $N = 2, ..., 40$. (Recall $SNR_{gdfe} = 2^{2*\bar{I}(X,Y)}$. Is there an easy way to compute $\bar{I}(X, Y)$??, THINK vector coding)

b. (2 pts) Repeat part a for the MMSE-DFE of Chapter 3 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 the performance of Vector Coding with waterfilling, which equals the performance of the GDFE with optimized input, for $N = 2, ..., 40$ with $\nu = 1$ and compare to parts (a) and (b).

d. (5 pts) Draw conclusions from your plots in parts (a), (b), (c).

5.15 *MIMO Antenna*

For the 6-real-dimensional cyclic channel visited in Chapter 4 with $P = \begin{bmatrix} 1 & .5 & .5 \\ .5 & 1 & .5 \\ .5 & .5 & 1 \end{bmatrix}$ and shown in Figure 5.37 with noise variance (2-side PSD) .01 and a gap of 0 dB, the energy per real dimension is 1. It may help to note that the Cholesky factorization of a matrix that is nearly white is trivial and generally for a 3×3 matrix, trivial linear prediction and the determinant of the matrix can be used to compute Cholesky factorization rapidly if if the matrix is not close to diagonal or matlab is not available.

a. (2 pts) Find a water-filling input covariance matrix $R_{\boldsymbol{xx}} =$? and corresponding Cholesky factorization $R_{\boldsymbol{xx}} = G_x R_{\boldsymbol{vv}} G_x^*$.

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 $P_e = 10^{-6}$?

5.16 *Bonded DMT*

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 5.38. 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 $\mathcal{E}_{1,n}$ and $\mathcal{E}_{2,n}$. 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

$$\boldsymbol{h}_k = \begin{bmatrix} h_{11,k} & h_{12,k} \\ h_{21,k} & h_{22,k} \end{bmatrix} \tag{5.380}$$

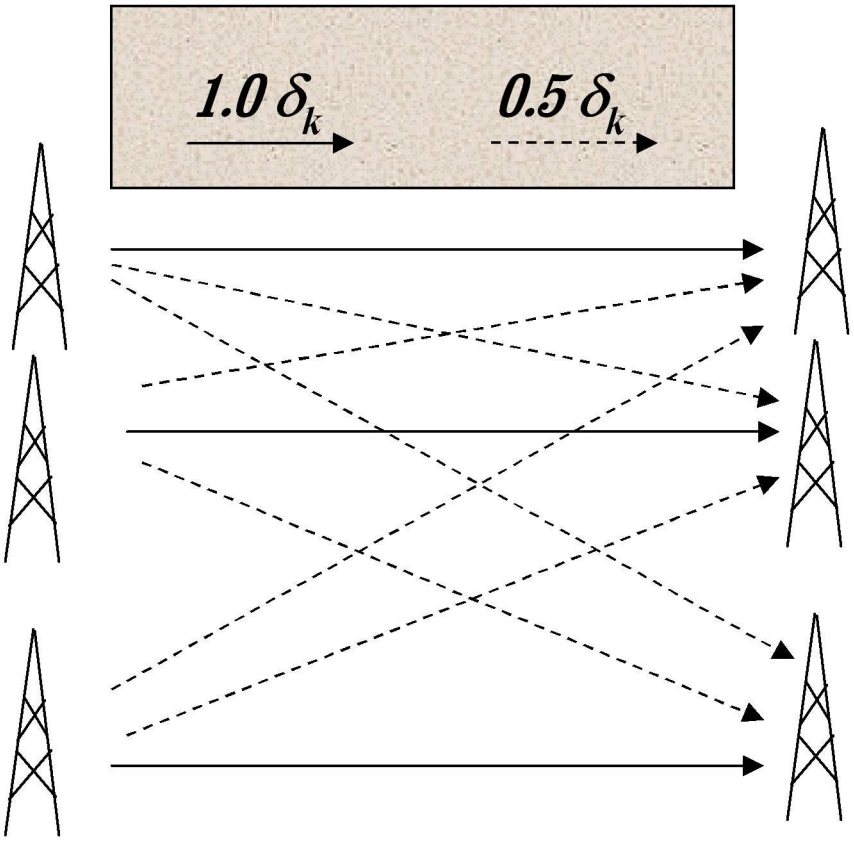


Figure 5.37: MIMO Antennae for Problem 5.15

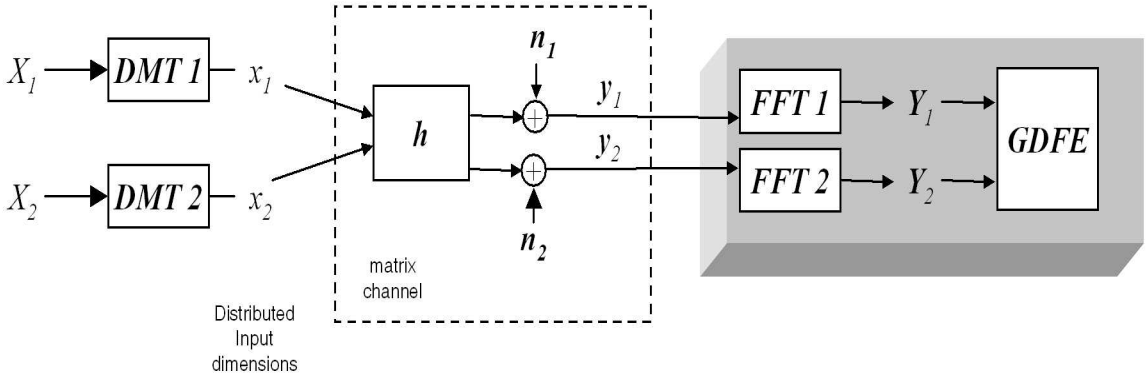


Figure 5.38: Illustration of bonded DMT for Problem 5.16

where each entry is an FIR channel and

$$h_{ij,0} + h_{ij,1} \cdot D + ... + h_{ij,\nu_{ij}} \cdot D^{\nu_{ij}} \quad . \tag{5.381}$$

The channel output is thus

$$\boldsymbol{y}(D) = \begin{bmatrix} y_1(D) \\ y_2(D) \end{bmatrix} = \boldsymbol{h}(D) \begin{bmatrix} x_1(D) \\ x_2(D) \end{bmatrix} + \boldsymbol{n}(D) \quad , \tag{5.382}$$

where $R_{\boldsymbol{nn}}(D) = \sigma^2 \cdot I$.

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 $\boldsymbol{H}_n$ 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 $r_{12,n} = H^*_{11,n} \cdot H_{21,n} + H^*_{21,n} \cdot H_{22,n}$ 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 $\bar{I}_1(\boldsymbol{x}_1; \boldsymbol{y})$ and $\bar{I}_2(\boldsymbol{x}_2; \boldsymbol{y})$, what is the overall SNR of the DMT/GDFE (bonded DMT) system?

5.17 Generalized Cholesky.

Given the discrete-time channel $P(D) = 1 + .8D$ with $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$ and $\frac{\mathcal{N}_0}{2} = .164$ and a gap of 0 dB:

a. (4 pts) Find the best $R_{\boldsymbol{xx}}$ and corresponding $\text{SNR}_{gdfe,u}$ 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 $\nu = 1$?

5.18 Some review questions.

a. (1 pt) For what types of $R_{\boldsymbol{xx}}$ can a CDFE be implemented? Equivalently, for what types of channel matrices P?

b. (1 pt) For what types of $R_{\boldsymbol{xx}}$ can a GDFE be implemented? Equivalently, what are the restrictions on P?

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 à 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).

5.19 A multi-band design.

We are given the channel $P(D) = 1 - D^3$ with $\frac{\mathcal{N}_0}{2} = .01$ and $\bar{\mathcal{E}}_{\boldsymbol{x}} = 1$.

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).

Appendix A

Information Measures

A concept generalizing the number of input bits to an encoder, b, is the **entropy**. Entropy (in this text) is defined for a stationary distribution for $\boldsymbol{x}$, $\mathrm{p}_{\boldsymbol{x}}(i)$, that exists in N-dimensions.[1]

Definition A.0.2 (Entropy) *The* **entropy** *for an N-dimensional sequential encoder with probability distribution $p_{\boldsymbol{x}}(i)$ $i = 0, ..., M-1$ is:*

$$H_{\boldsymbol{x}} \triangleq -\sum_{i=0}^{M-1} p_{\boldsymbol{x}}(i) \log_2 \left[p_{\boldsymbol{x}}(i)\right] \quad \textit{bits/symbol} \tag{A.1}$$

$$= E\left\{\log_2 \left[1/p_{\boldsymbol{x}}\right]\right\} \quad , \tag{A.2}$$

The entropy $H_{\boldsymbol{y}}$ of the channel output is also similarly defined for discrete channel-output distributions.

The entropy of a random variable can be interpreted as the "information content" of that random variable, in a sense a measure of its "randomness" or "uncertainty." It can be easily shown that a discrete uniform distribution has the largest entropy, or information – or uncertainty over all discrete distributions. That is, all values are just as likely to occur. A deterministic quantity ($p_{\boldsymbol{x}}(i) = 1$ for one value of i and $p_{\boldsymbol{x}}(j) = 0\ \forall j \neq i$) has no information, nor uncertainty. For instance, a uniform distribution on 4 discrete values has entropy $H_{\boldsymbol{x}} = 2$ bits/symbol. This is the same as b for 4-level PAM or 4 QAM with uniform input distributions. The entropy of a source is the essential bit rate of information coming from that source. If the source distribution is not uniform, the source does not have maximum entropy and more information could have been transmitted (or equivalently transmitted with fewer messages M) with a different representation of the source's message set. Prior to this chapter, most of the message sets considered were uniform in distribution, so that the information carried was essentially b, the base-2 log of the number of messages. In general, $H_{\boldsymbol{x}} \leq \log_2(M)$, where M is the number of values in the discrete distribution.

In the case of a continuous distribution, the **differential entropy** becomes:

$$H_{\boldsymbol{y}} \triangleq -\int_{-\infty}^{\infty} p_{\boldsymbol{y}}(u) \log_2 \left[\mathrm{p}_{\boldsymbol{y}}(u)\right] du \quad . \tag{A.3}$$

Theorem A.0.1 (Maximum Entropy of a Gaussian Distribution) *The distribution with maximum differential entropy for a fixed variance σ_y^2 is Gaussian.*

Proof: Let us denote the Gaussian distribution as $g_y(v)$, then

$$\log_2 g_y(v) = -\log_2\left(\sqrt{2\pi\sigma_y^2}\right) - \left(\frac{v}{\sqrt{2}\sigma_y}\right)^2 \bullet \left(\ln(2)\right)^{-1} \quad . \tag{A.4}$$

[1] Information measures in this book use a base-2 logarithm and are measured in bits per symbol. Any other base, p, could be used for the logarithm, and then the measures would be in the $p$its/symbol!

For any other distribution $p_y(v)$ with mean zero and the same variance,

$$-\int_{-\infty}^{\infty} p_y(v)\log_2\left(g_y(v)\right)dv = \log_2\left(\sqrt{2\pi\sigma_y^2}\right) + \frac{1}{2\ln(2)} \quad , \tag{A.5}$$

which depends only on σ_y^2. Then, letting the distribution for y be an argument for the entropy,

$$\begin{aligned}
H_y(g_y) - H_y(p_y) &= && \text{(A.6)}\\
&= -\int_{-\infty}^{\infty} g_y(v)\log_2(g_y(v))dv + \int_{-\infty}^{\infty} p_y(v)\log_2(p_y(v))dv && \text{(A.7)}\\
&= -\int_{-\infty}^{\infty} p_y(v)\log_2(g_y(v))dv + \int_{-\infty}^{\infty} p_y(v)\log_2(p_y(v))dv && \text{(A.8)}\\
&= -\int_{-\infty}^{\infty} p_y(v)\log_2\left(\frac{g_y(v)}{p_y(v)}\right)dv && \text{(A.9)}\\
&\geq \frac{1}{\ln 2}\int_{-\infty}^{\infty} p_y(v)\left(1-\frac{g_y(v)}{p_y(v)}\right)dv && \text{(A.10)}\\
&\geq \frac{1}{\ln 2}(1-1) = 0 \quad , && \text{(A.11)}
\end{aligned}$$

or[2]

$$H_y(g_y) \geq H_y(p_y) \quad . \tag{A.12}$$

QED.

With simple algebra,

$$H(g_y) = \frac{1}{2}\log_2(2e\pi\sigma_y^2) \quad . \tag{A.13}$$

For baseband complex signals, H_y, in bits per complex (two-dimensional) symbol is often written $H_y = \log_2\left(\pi e\sigma_y^2\right)$ where σ_y^2 becomes the variance of the complex random variable (which is twice the variance of the variance of the real part of the complex variable, when real and imaginary parts have the same variance, as is almost always the case in data transmission).

Most of the uses of entropy in this text are associated with either the encoder distribution $\mathrm{p}_{\boldsymbol{x}}$, or with the channel output distributions $p_{\boldsymbol{y}}$ or $p_{\boldsymbol{y}/\boldsymbol{x}}$. The normalization of the number of bits per dimension to $\bar{b} = \frac{b}{N}$ tacitly assumes that the successively transmitted dimensions were independent of one another. In the case of independent successive dimensions, $H_{\boldsymbol{x}} = NH_x$. H_x is equal to $\bar{b}$, if the distribution on each dimension is also uniform (as well as independent of the other dimensions). Equivalently in (A.2), $H_{\boldsymbol{x}} = N \cdot H_x$ if each of the dimensions of $\boldsymbol{x}$ is independent.

For instance, 16QAM has entropy $H_{\boldsymbol{x}} = b = 4$ bits/symbol and normalized entropy $\bar{H}_x = \bar{b} = 2$ bits/dimension. However, 32CR has entropy $H_{\boldsymbol{x}} = b = 5$ bits/symbol, but since $p(\pm x = 1) = p(\pm x = 3) = 6/32$ and $p(\pm 5) = 4/32$, the entropy $H_x = 2.56$ bits/dimension $\neq \bar{H}_{\boldsymbol{x}} = \bar{b} = 2.5$ bits/dimension. Note also that the number of one-dimensional distribution values is 6, so $H_x = 2.56 < \log_2(6) = 2.58$.

The differential entropy of a (real) Gaussian random variable with variance $\bar{\sigma}^2$ is

$$H_x = \frac{1}{2}\log_2\left(2\pi e\bar{\sigma}^2\right) \quad \text{bits/symbol.} \tag{A.14}$$

A complex Gaussian variable with variance σ^2 has differential entropy

$$H_{\boldsymbol{x}} = \log_2\left(\pi e\sigma^2\right) \quad \text{bits/symbol.} \tag{A.15}$$

[2]Equation (A.11) uses the bound $\ln(x) \geq x - 1$.

The conditional entropy of one random variable given another is defined according to

$$H_{\boldsymbol{x}/\boldsymbol{y}} \triangleq \sum_{\boldsymbol{v}} \sum_{i=0}^{M-1} \mathrm{p}_{\boldsymbol{x}}(i)\mathrm{p}_{\boldsymbol{y}/\boldsymbol{x}}(\boldsymbol{v},i) \log_2 \frac{1}{\mathrm{p}_{\boldsymbol{x}/\boldsymbol{y}}(\boldsymbol{v},i)} \tag{A.16}$$

$$H_{\boldsymbol{y}/\boldsymbol{x}} \triangleq \sum_{\boldsymbol{v}} \sum_{i=0}^{M-1} \mathrm{p}_{\boldsymbol{x}}(i)\mathrm{p}_{\boldsymbol{y}/\boldsymbol{x}}(\boldsymbol{v},i) \log_2 \frac{1}{\mathrm{p}_{\boldsymbol{y}/\boldsymbol{x}}(\boldsymbol{v},i)} \quad , \tag{A.17}$$

with integrals replacing summations when random variables/vectors have continuous distributions. The definition of conditional entropy averages the entropy of the conditional distribution over all possibilities in some given input distribution. Thus, the conditional entropy is a function of both $p_{\boldsymbol{y}/\boldsymbol{x}}$ and $p_{\boldsymbol{x}}$. Conditional entropy measures the residual information or uncertainty in the random variable given the value of another random variable on average. This can never be more than the entropy of the unconditioned random variable, and is only the same when the two random variables are independent. For a communication channel, the conditional entropy, $H_{\boldsymbol{y}/\boldsymbol{x}}$, is basically the uncertainty or information of the "noise." If the conditional distribution is Gaussian, as is often the case in transmission, the conditional entropy of a scalar x becomes

$$H_{x/\boldsymbol{y}} = \begin{cases} \frac{1}{2}\log_2\left(2\pi e\sigma^2_{mmse}\right) & \text{real } x \\ \log_2\left(\pi e\sigma^2_{mmse}\right) & \text{complex } x \end{cases} \tag{A.18}$$

The MMSE in the above equations is that arising from estimation of x, given $\boldsymbol{y}$. Thus, the conditional entropy then measures the information remaining after the effect of the $\boldsymbol{y}$ has been removed. That is in some sense measuring useless information that a receiver might not be expected to use successfully in estimating $\boldsymbol{x}$.

The information of the "noise," or more generically, the "useless part" of the channel output given a certain input distribution, is not of value to a receiver. Thus, while the entropy of the source is a meaningful measure of the data transmitted, the entropy of the channel output has extra constituents that are caused by the randomness of noise (or other useless effects). Given that it is the output of a channel, $\boldsymbol{y}$, that a receiver observes, only that part of the output that bears the information of the input is of value in recovering the transmitted messages. Thus, the entropy $H_{\boldsymbol{y}} - H_{\boldsymbol{y}/\boldsymbol{x}}$ measures the useful information in the channel output. This information is called the **mutual information**.

Definition A.0.3 (Mutual Information) *The* **mutual information** *for any N-dimensional signal set with probability distribution $p_{\boldsymbol{x}}(i)$ $i = 0, ..., M-1$, and a corresponding channel description $p_{\boldsymbol{y}/\boldsymbol{x}}(\boldsymbol{v}, i)$, is:*

$$I_{\boldsymbol{y},\boldsymbol{x}} \triangleq H_{\boldsymbol{y}} - H_{\boldsymbol{y}/\boldsymbol{x}} \quad . \tag{A.19}$$

The identity,

$$I_{\boldsymbol{y},\boldsymbol{x}} = H_{\boldsymbol{y}} - H_{\boldsymbol{y}/\boldsymbol{x}} = H_{\boldsymbol{x}} - H_{\boldsymbol{x}/\boldsymbol{y}} \quad , \tag{A.20}$$

easily follows from transposing $\boldsymbol{x}$ and $\boldsymbol{y}$. Using probability distributions directly:

$$I_{\boldsymbol{y},\boldsymbol{x}} \triangleq \sum_{\boldsymbol{v}} \sum_{i=0}^{M-1} p_{\boldsymbol{x}}(i)p_{\boldsymbol{y}/\boldsymbol{x}}(\boldsymbol{v},i) \log_2 \left[\frac{p_{\boldsymbol{y}/\boldsymbol{x}}(\boldsymbol{v},i)}{\sum_{m=0}^{M-1} p_{\boldsymbol{y}/\boldsymbol{x}}(\boldsymbol{v},m)p_{\boldsymbol{x}}(m)}\right] \quad \text{bits/symbol} \tag{A.21}$$

$$= E\log_2\left[\frac{p_{\boldsymbol{y}/\boldsymbol{x}}}{\mathrm{p}_{\boldsymbol{y}}}\right] \tag{A.22}$$

$$= E\log_2\left[\frac{p_{\boldsymbol{y},\boldsymbol{x}}}{p_{\boldsymbol{x}} \cdot p_{\boldsymbol{y}}}\right] \tag{A.23}$$

$$= E\log_2\left[\frac{p_{\boldsymbol{x}/\boldsymbol{y}}}{p_{\boldsymbol{x}}}\right] \quad , \tag{A.24}$$

In the case of a continuous distribution on y and/or x, the summation(s) is (are) replaced by the appropriate integral(s).

Appendix B

Cholesky Factorization

This appendix illustrates two approaches to Cholesky factorization that can either simply the direct computation of triangular matrices in Section B.1 or provide a MMSE approach as in Section **??**.

B.1 The Simple Arithmetic Approach

This approach was provided by former EE379C student Chan Soo Hwang.

Definition of Cholesky decomposition Given a symmetric positive definite matrix R, the Cholesky decomposition is an upper triangular matrix G such that

$$R = G^* S G \tag{B.1}$$

B.1.1 Direct Arithmetic Computation of Cholesky decomposition

Cholesky decomposition in this text requires G to be monic, i.e. all the diagonal terms are 1. Therefore, the decomposition is $R = G^* S G$ where S is a diagonal matrix and G is monic upper triangular matrix. Since elementary row operations[1] are equivalent to multiplication by monic triangular matrices, we can obtain G, an upper triangular matrix, by finding a sequence of elementary matrices that transforms R to upper triangular. Application of the same matrix sequence to identity matrix provides $G^* S$.

The squence of transformations alternates between diagonal scaling and monic elementary triangular row operations. The diagonal scaling force the resultant matrix from the immediately preceding step to be monic. The monic row operations are then used to zero all entries below the diagonal in one column of the resultant matrix. Thus, the sequency appears as

$$S_N^{-1} \cdot G_{N-1}^{-1} \cdot S_{N-1}^{-1} \cdot \ldots \cdot G_1 \cdot S_1^{-1} \cdot R = G \ . \tag{B.2}$$

The resultant matrix is G and thus, the uniqueness of Cholesky factorization (non-singular matrix) forces the left side of the equation to be also $S^{-1} \cdot G^{-*} R$. Since the G matrices are all monic, then the overall S diagonal matrix is simply the product of the S_i matrices.

The computation is illustrated through use of an example from the text. Exercise 5.15 uses the following matrix as R_b^{-1} which will be used to derive feedback and feedforward filter of GDFE.

$$R_b^{-1} = \begin{bmatrix} 151 & 125 & 125 \\ 125 & 151 & 125 \\ 125 & 125 & 151 \end{bmatrix} \tag{B.3}$$

Step 1a) Normalize each row by the first column

$$S_1 \quad = \quad \begin{bmatrix} 151 & 0 & 0 \\ 0 & 125 & 0 \\ 0 & 0 & 125 \end{bmatrix} \tag{B.4}$$

[1] An elementary row operation is the replacement of any row by the sum of it and some scalar times another row.

$$R_{1a} = \begin{bmatrix} 1 & 0.828 & 0.828 \\ 1 & 1.21 & 1 \\ 1 & 1 & 1.21 \end{bmatrix} \tag{B.5}$$

Step 1b) Subtract the first row from the second and third row

$$G_1 = \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \tag{B.6}$$

$$R_{1b} = \begin{bmatrix} 1 & 0.828 & 0.828 \\ 0 & 0.382 & 0.172 \\ 0 & 0.172 & 0.382 \end{bmatrix} \tag{B.7}$$

Step 2a) Normalize the second and third row by the second column

$$S_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0.382 & 0 \\ 0 & 0 & 0.172 \end{bmatrix} \tag{B.8}$$

$$R_{2a} = \begin{bmatrix} 1 & 0.828 & 0.828 \\ 0 & 1 & 0.45 \\ 0 & 1 & 2.22 \end{bmatrix} \tag{B.9}$$

Step 2b) Subract the second row from the third row

$$G_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \tag{B.10}$$

$$R_{2b} = \begin{bmatrix} 1 & 0.828 & 0.828 \\ 0 & 1 & 0.45 \\ 0 & 1 & 2.22 \end{bmatrix} \tag{B.11}$$

Step 3) Normalize the third row by the third column

$$S_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1.77 \end{bmatrix} \tag{B.12}$$

$$R_3 = \begin{bmatrix} 1 & 0.828 & 0.828 \\ 0 & 1 & 0.45 \\ 0 & 0 & 1 \end{bmatrix} \tag{B.13}$$

The overall S is then

$$S = \begin{bmatrix} 151 & 0 & 0 \\ 0 & 47.8 & 0 \\ 0 & 0 & 38.1 \end{bmatrix} \tag{B.14}$$

Cholesky decomposition for $R_{\boldsymbol{xx}}$

In Section 5.5, the Cholesky decomposition locates the upper triangle first, and lower triangular matrix last. To do this calculation, the sequence instead then works above the diagonal instead of below. As an example"

$$R_{\boldsymbol{xx}} = \begin{bmatrix} 0.6715 & 0.2956 \\ 0.2956 & 0.7340 \end{bmatrix} \quad . \tag{B.15}$$

Step1) Normalize each row by the first column and Subtract the second row from the first row.

$$S_1 = \begin{bmatrix} 0.2956 & 0 \\ 0 & 0.7340 \end{bmatrix} \tag{B.16}$$

$$G_1 = \begin{bmatrix} 1 & -1 \\ 0 & 1 \end{bmatrix} \tag{B.17}$$

$$R_1 = \begin{bmatrix} 1 & 0 \\ 0.403 & 1 \end{bmatrix} \tag{B.18}$$

Step 2) Normalize the first row

$$S_2 = \begin{bmatrix} 1.87 & 0 \\ 0 & 1 \end{bmatrix} \tag{B.19}$$

$$G = \begin{bmatrix} 1 & 0 \\ 0.403 & 1 \end{bmatrix} \tag{B.20}$$

Then

$$S = \begin{bmatrix} 1.87(.2956) & 0 \\ 0 & .7340 \end{bmatrix} \quad . \tag{B.21}$$

B.2 MMSE interpretation of Cholesky Factorization

The MMSE approach to Cholesky factorization is trivially Generalized Cholesky without the extra rows and zeroed prediction errors.