# Solutions to Chapter 2 Problems

*Fundamentals of Digital Communication*

**Problem 2.1:** Rather than doing the details of the convolution, we simply sketch the shapes of the waveforms. For a signal $s = s_c + js_s$ and a filter $h = h_c + jh_s$, the convolution

$$y = s * h = (s_c * h_c - s_s * h_s) + j(s_c * h_s + s_s * h_c)$$

For $h(t) = s_{mf}(t) = s^*(-t)$, rough sketches of $\mathrm{Re}(y)$, $\mathrm{Im}(y)$ and $|y|$ are shown in Figure 1. Clearly, the maximum occurs at $t = 0$.

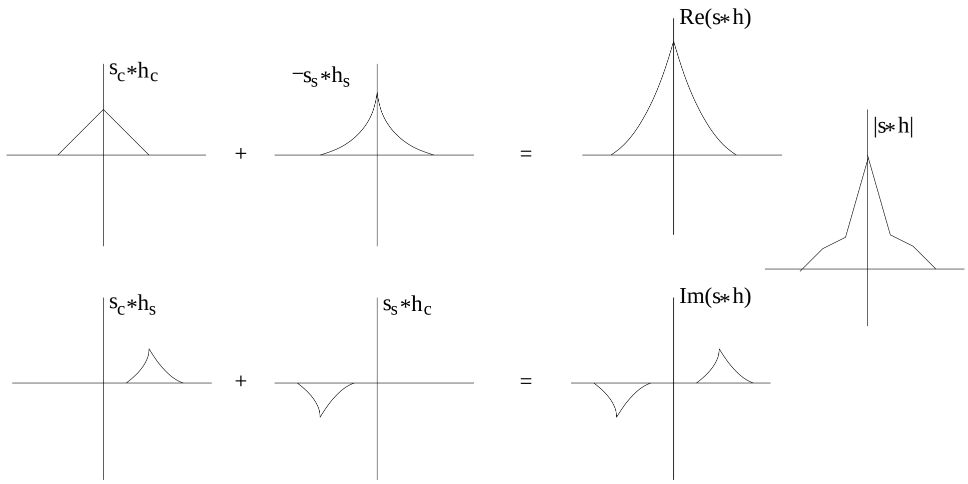



Figure 1: The convolution of a signal with its matched filter yields at peak at the origin.

**Problem 2.2:**
(a) Multiplication in the time domain corresponds to convolution in the frequency domain. The two sinc functions correspond to boxcars in the frequency domain, convolving which gives that $S(f)$ has a trapezoidal shape, as shown in Figure 2.
(b) We have

$$u(t) = s(t)\cos(100\pi t) = s(t)\frac{e^{j100\pi t} + e^{-j100\pi t}}{2} \leftrightarrow U(f) = \frac{S(f-50) + S(f+50)}{2}$$

The spectrum $U(f)$ is plotted in Figure 2.

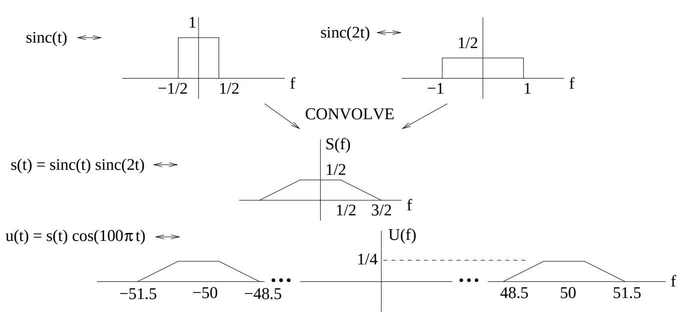



Figure 2: Solution for Problem 2.2.

**Problem 2.3:** The solution is sketched in Figure 3.
(a) We have $s(t) = I_{[-5,5]} * I_{[-5,5]}$. Since $I_{[-5,5]}(t) \leftrightarrow 10\text{sinc}(10f)$, we have $S(f) = 100\text{sinc}^2(10f)$.
(b) We have

$$u(t) = s(t)\sin(1000\pi t) = s(t)\frac{e^{j1000\pi t} - e^{-j100\pi t}}{2j} \leftrightarrow U(f) = \frac{S(f-50) - S(f+50)}{2j}$$

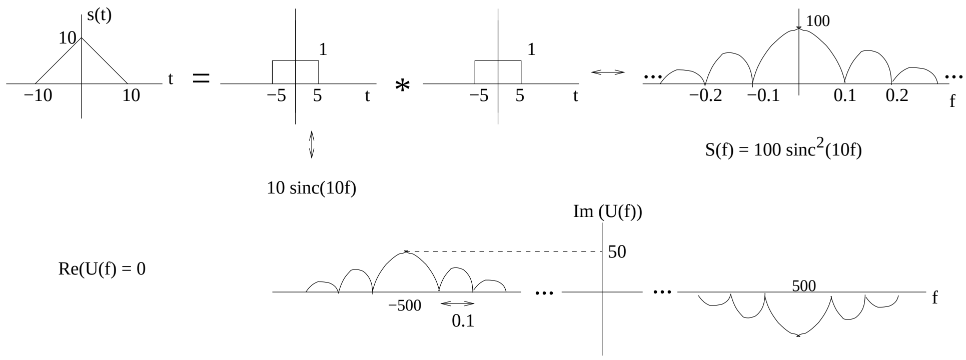



Figure 3: Solution for Problem 2.3.

**Problem 2.4:** Part (a) is immediate upon expanding $||s - ar||^2$.
(b) The minimizing value of $a$ is easily found to be

$$a_{min} = \frac{\langle s, r\rangle}{||r||^2}$$

Substituting this value into $J(a)$, we obtain upon simplification that

$$J(a_{min}) = ||s||^2 - \frac{\langle s, r\rangle^2}{||r||^2}$$

The condition $J(a_{min}) \geq 0$ is now seen to be equivalent to the Cauchy-Schwartz inequality.
(c) For nonzero $s$, $r$, the minimum error $J(a_{min})$ in approximating $s$ by a multiple of $r$ vanishes if and only if $s$ is a multiple of $r$. This is therefore the condition for equality in the Cauchy-Scwartz inequality. For $s = 0$ or $r = 0$, equality clearly holds. Thus, the condition for equality can be stated in general as: either $s$ is a scalar multiple of $r$ (this includes $s = 0$ as a special case), or $r$ is a scalar multiple of $s$ (this includes $r = 0$ as a special case).
(d) The unit vector in the direction of $r$ is $u = \frac{r}{||r||}$. The best approximation of $s$ as a multiple of $r$ is its projection along $u$, which is given by

$$\hat{s} = \langle s, u\rangle u = \langle s, \frac{r}{||r||}\rangle \frac{r}{||r||}$$

and the minimum error is $J(a_{min}) = ||s - \hat{s}||^2$.
**Problem 2.5:** We have

$$y(t) = (x * h)(t) = \int As(\tau - t_0)h(t - \tau)d\tau = A\langle s_{t_0}, h_t^*\rangle$$

where $s_{t_0}(\tau) = s(\tau - t_0)$ and $h_t(\tau) = h(t - \tau)$ are functions of $\tau$. (Recall that the complex inner product is defined as $\langle u, v\rangle = \int uv^*$).
(a) Using the Cauchy-Schwartz inequality, we have

$$|y(t)| \leq |A|||s_{t_0}||||h_t^*||$$

It is easy to check (simply change variables in the associated integrals) that $||s_{t_0}|| = ||s||$ and $||h_t^*|| = ||h||$. Using the normalization $||h|| = ||s||$, we obtain the desired result that $|y(t)| \leq A||s||^2$.
(b) Equality is attained for $t = t_0$ if $h_t^* = as_{t_0}$ for $t = t_0$ for some scalar $a$. Since $||s_{t_0}|| = ||h_t^*|| = ||s||$, we must have $|a| = 1$. Thus, we have

$$h^*(t_0 - \tau) = as(\tau - t_0) \quad (|a| = 1)$$

for all $\tau$. (We would get $a = 1$ if we insisted that $y(t_0) = A||s||^2$ rather than $|y(t_0)| = |A|||s||^2$.) Setting $a = 1$, therefore, we have $h(t) = s^*(-t)$.

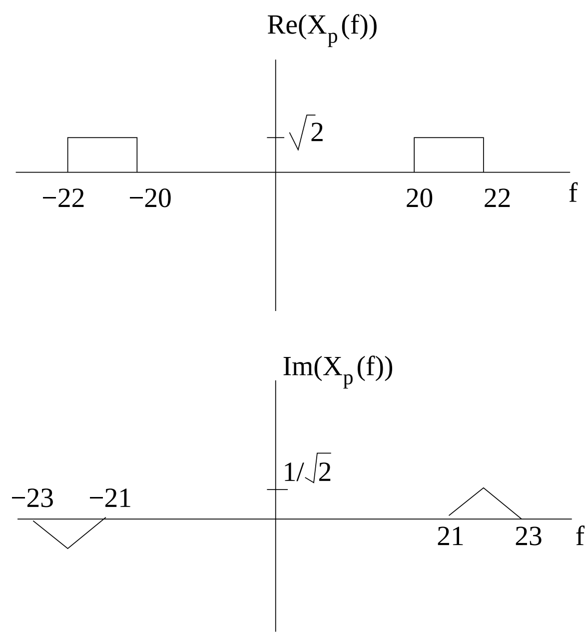



Figure 4: Passband spectrum for Problem 2.6(a).

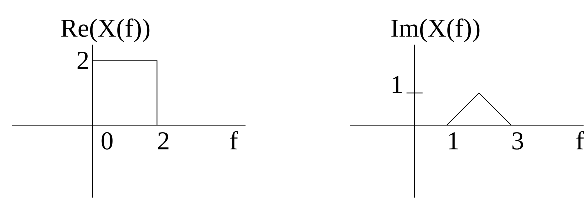



Figure 5: Complex baseband spectrum for Problem 2.6(b).

**Problem 2.6** (a) The real and imaginary parts of $X_p(f)$ are sketched in Figure 4.
(b) Passing $\sqrt{2}x_p(t)\cos 20\pi t$ through a lowpass filter yields $x_c(t)$, the I component with respect to $f_c = 20$. In this case, it is easiest to find the Fourier transform (see Figure 5), and then the time domain expression, for the complex baseband signal $x(t)$, and then take the real part. We see that

$$\mathrm{Re}(X(f)) \leftrightarrow 4\mathrm{sinc}(2t)e^{j2\pi t}$$

$$\mathrm{Im}(X(f)) \leftrightarrow \mathrm{sinc}^2(t)e^{j4\pi t}$$

so that

$$X(f) = \mathrm{Re}(X(f)) + j\mathrm{Im}(X(f)) \leftrightarrow x(t) = 4\mathrm{sinc}(2t)e^{j2\pi t} + j\mathrm{sinc}^2(t)e^{j4\pi t}$$

We can now read off

$$x_c(t) = \text{Re}(x(t)) = 4\text{sinc}(2t)\cos(2\pi t) - \text{sinc}^2(t)\sin(4\pi t)$$

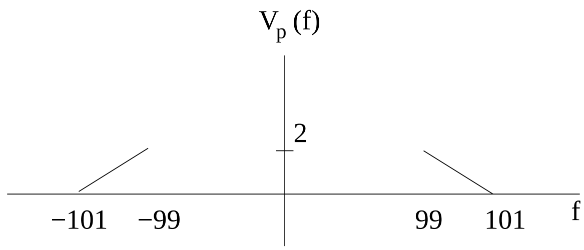



Figure 6: Passband spectrum for Problem 2.7(a).

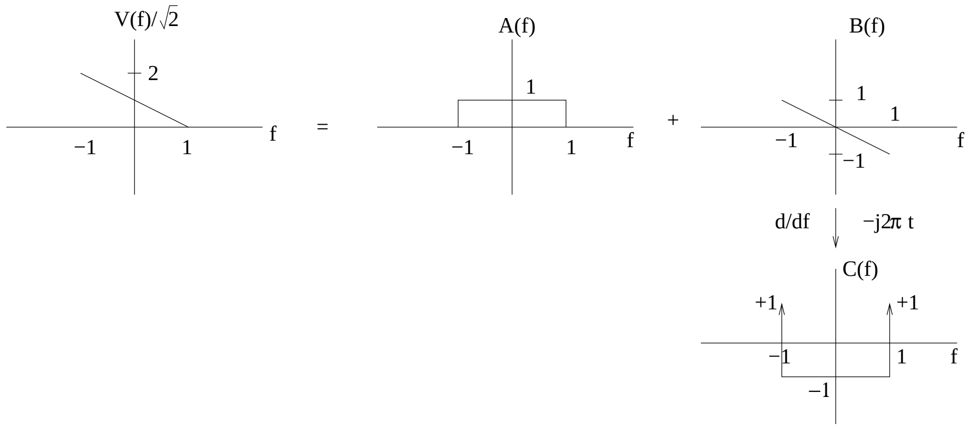



Figure 7: Computing the complex baseband representation in Problem 2.7(c).

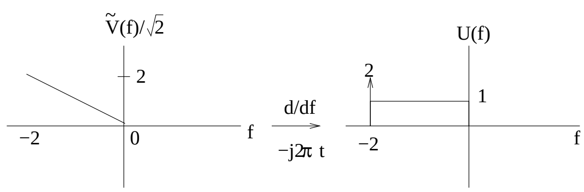



Figure 8: Computing the complex baseband representation in Problem 2.7(d).

**Problem 2.7** (a) The passband spectrum $V_p(f)$ is sketched in Figure 6, where we use the fact that $V(f) = V^*(-f)$, since $v(t)$ is real-valued.
(b) We have $v(t) = v(-t)$ if $V(f) = V(-f)$, which is clearly true here.
(c) The complex baseband spectrum $V(f)$, with respect to $f_0 = 100$, is sketched in Figure 7. In order to compute $v(t)$ with minimal work, we break up $V(f)/\sqrt{2}$ into two components, $A(f)$ and $B(f)$. Clearly, $a(t) = 2\text{sinc}(2t)$. To find $b(t)$, we minimize work further by differentiating in the frequency domain to obtain $C(f) = d/df\ B(f)$. We have $c(t) = 2\cos 2\pi t - 2\text{sinc}(2t) = -j2\pi t b(t)$, which implies that

$$v(t)/\sqrt{2} = a(t) + b(t) = 2\text{sinc}(2t) + \frac{\cos 2\pi t - \text{sinc}(2t)}{-j\pi t}$$

We therefore obtain that the I and Q components are given by

$$v_c(t) = \mathrm{Re}(v(t)) = 2\sqrt{2}\mathrm{sinc}(2t)$$

$$v_s(t) = \mathrm{Im}(v(t)) = \sqrt{2}\frac{\cos 2pit - \mathrm{sinc}(2t)}{\pi t}$$

Note that we could have done this even more easily by directly differentiating $V(f)$. We illustrate this in the next part.
(d) Denoting the complex baseband representation with respect to $f_0 = 101$ as $\tilde{v}$, we have that

$$v_p(t) = \mathrm{Re}\left(\sqrt{2}\tilde{v}(t)e^{j202\pi t}\right) = \mathrm{Re}\left(\sqrt{2}v(t)e^{j200\pi t}\right)$$

so that

$$\tilde{v}(t) = v(t)e^{-j2\pi t}$$

Substituting into (c) and reading off real and imaginary parts gives us the desired result. Alternatively, let us proceed from scratch, as shown in Figure 8. The complex baseband representation $\tilde{V}(f)$ is shown. To minimize computations, we differentiate in the frequency domain to get $U(f)$, as shown. Clearly,

$$u(t) = 2e^{-j4\pi t} + 2\mathrm{sinc}(2t)e^{-j2\pi t} = -j2\pi t\tilde{v}(t)/\sqrt{2}$$

We can now simplify to read off $\tilde{v}(t)$ and the corresponding I and Q components.

**Problem 2.8** (a) For a frequency reference $f_c = 50$, clearly the complex envelope for $u_p$ is given by $u(t) = \mathrm{sinc}(2t)$. For $v_p$, we write

$$v_p(t) = \sqrt{2}\mathrm{sinc}(t)\sin(101\pi t + \frac{\pi}{4}) = \sqrt{2}\mathrm{Re}\left(\mathrm{sinc}(t)(-je^{j(101\pi t+\frac{\pi}{4})}\right) = \sqrt{2}\mathrm{Re}\left(\mathrm{sinc}(t)e^{j(\pi t-\frac{\pi}{4})}e^{j100\pi t}\right)$$

(where we have used $-j = e^{-j\pi/2}$), from which we can read off

$$v(t) = \mathrm{sinc}(t)e^{j(\pi t-\frac{\pi}{4})}$$

(b) Note that $U(f)$ occupies a band of length 2, while $V(f)$ occupies a band of length 1. Thus, $u_p$ has a bandwidth of 2, and $v_p$ has a bandwidth of 1.
(c) The inner product

$$\langle u_p, v_p\rangle = \mathrm{Re}\left(\langle u, v\rangle\right) = \mathrm{Re}\left(\int U(f)V^*(f)df\right)$$

But $U(f) = \frac{1}{2}I_{[-1,1]}(f)$ and $V(f) = e^{-j\frac{\pi}{4}}I_{[0,1]}(f)$, so that

$$\int U(f)V^*(f)df = \frac{1}{2}e^{j\frac{\pi}{4}}$$

We therefore obtain that

$$\langle u_p, v_p\rangle = \frac{1}{2\sqrt{2}}$$

(d) We have $Y(f) = \frac{1}{\sqrt{2}}U(f)V(f) = \frac{1}{2\sqrt{2}}V(f)$ (since $U(f)$ takes the constant value $\frac{1}{2}$ over the support of $V(f)$). This implies that $y_p(t) = \frac{1}{2\sqrt{2}}v_p(t) = \frac{1}{2}\mathrm{sinc}(t)\sin(101\pi t + \frac{\pi}{4})$.

**Problem 2.9:** (a) Since $u(t) = u^*(t)$, we have $U(f) = U^*(-f)$. For $-1 < f < 0$, we have $U(f) = U^*(-f) = \left(e^{j\pi(-f)}\right)^* = e^{j\pi f}$. Thus, $U(f) = e^{j\pi f}I_{[-1,1]}(f)$. We therefore have $\mathrm{Re}(U(f)) = \cos\pi f I_{[-1,1]}(f)$ and $\mathrm{Im}(U(f)) = \sin\pi f I_{[-1,1]}(f)$, as sketched in Figure 9.

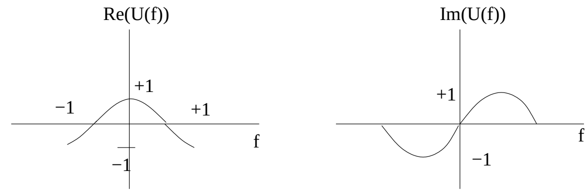



Figure 9: Baseband spectrum in Problem 2.9(a).

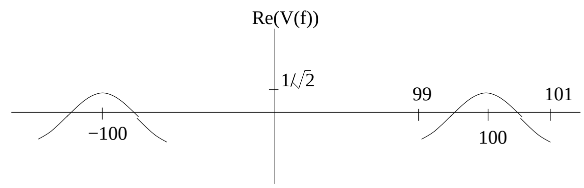



Figure 10: Passband spectrum in Problem 2.9(c).

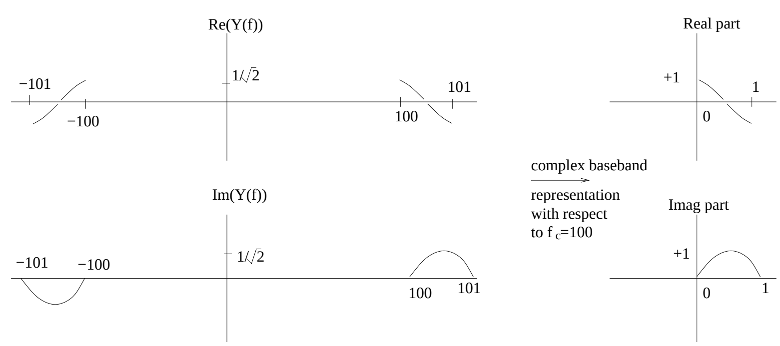



Figure 11: Passband spectrum after highpass filtering in Problem 2.9(d), along with its complex baseband representation with respect to $f_c = 100$.

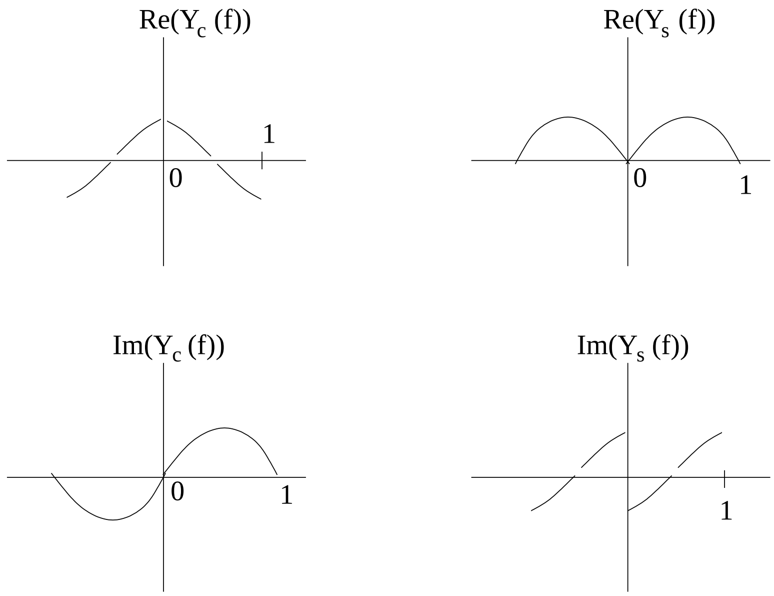



Figure 12: I and Q components for the highpass filtered waveform. The I component is the same as the original message $u(t)$, up to scale factor.

(b) We have $u(t) = 2\text{sinc}(2(t + \frac{1}{2}))$.
(c) We have

$$V(f) = \sqrt{2}\frac{U(f-100) + U(f+100)}{2}$$

as sketched in Figure 10.
(d) After highpass filtering, we obtain the waveform $y(t)$ whose spectrum is shown in Figure 11. The corresponding complex baseband spectrum, relative to $f_c = 100$ is also shown in the figure. Calling the complex envelope $\tilde{Y}(f)$, we have $y_c(t) = \text{Re}(\tilde{y}(t)) = \frac{\tilde{y}(t)+\tilde{y}^*(t)}{2}$, so that

$$Y_c(f) = \frac{\tilde{Y}(f) + \tilde{Y}^*(-f)}{2}$$

This yields

$$\text{Re}(Y_c(f)) = \frac{\text{Re}\big(\tilde{Y}(f)\big)+\text{Re}\big(\tilde{Y}(-f)\big)}{2}$$
$$\text{Re}(Y_c(f)) = \frac{\text{Im}\big(\tilde{Y}(f)\big)-\text{Im}\big(\tilde{Y}(-f)\big)}{2}$$

Similarly,

$$Y_s(f) = \frac{\tilde{Y}(f) - \tilde{Y}^*(-f)}{2j}$$

This yields

$$\text{Re}(Y_c(f)) = \frac{\text{Im}\big(\tilde{Y}(f)\big)+\text{Im}\big(\tilde{Y}(-f)\big)}{2}$$
$$\text{Re}(Y_c(f)) = \frac{-\text{Re}\big(\tilde{Y}(f)\big)+\text{Re}\big(\tilde{Y}(-f)\big)}{2}$$

The spectra are shown in Figure 12, where we have stopped worrying about scale factors.
(e) The output of the LPF is $y_c(t)/\sqrt{2}$, which is the same as the original waveform $u(t)$, up to scale factor.
**Remark:** This problem illustrates the principle underlying single sideband amplitude modulation. Highpass filtering gets rid of one of the sidebands in the passband waveform to be sent

over the medium. However, when we downconvert the signal back to baseband, we recover the original message waveform as the I component.

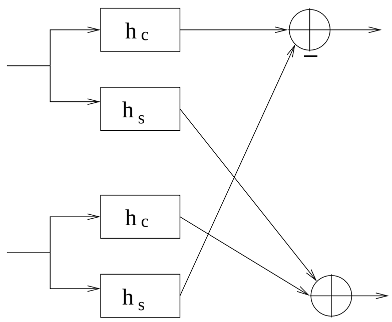



Figure 13: Real baseband operations for Problem 2.10(a), where the filters are specified in Figure 15.

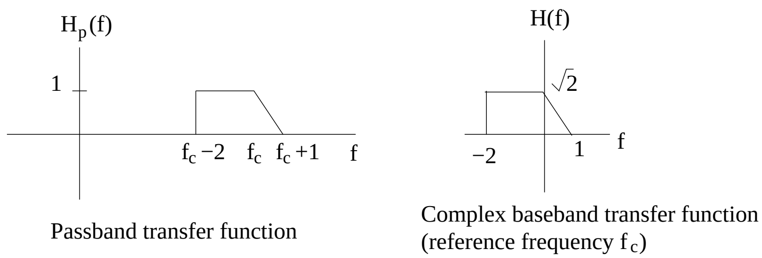



Figure 14: Passband filter and corresponding complex baseband filter with respect to $f_c$ for Problem 2.10(a).

**Problem 2.10:** The input to the baseband box are the I and Q components of $u_p(t)$ with respect to $f_1$ (plus double frequency terms that get filtered out by the baseband filters used in the box). The output of the box should be the I and Q components of $y_p$ with respect to $f_2$.
(a) For $f_1 = f_2 = f_c$, the baseband box need only implement the complex baseband filtering operation (ignoring scale factors):

$$y = y_c + jy_s = u * h = (u_c + ju_s) * (h_c + jh_s)$$

so that $y_c = u_c * h_c - u_s * h_s$ and $y_s = u_c * h_s + u_s * h_c$, as shown in Figure 13. The transfer functions $H(f)$, $H_c(f)$ and $H_s(f)$ are specified in Figures 14 and 15.
(b) When $f_1 = f_c + \frac{1}{2}$ and $f_2 = f_c - \frac{1}{2}$, we have several choices on how to design the real baseband box. We choose to use the same filters as in (a), at a reference frequency of $f_c$, converting from reference frequency $f_1$ to reference frequency $f_c$ at the input, and from reference frequency $f_c$ to reference frequency $f_2$ at the output. Specifically, we have at the input

$$u_p(t) = \text{Re}\left(u_1(t)e^{j2\pi(f_c+0.5)t}\right) = \text{Re}\left(u(t)e^{j2\pi f_c t}\right)$$

which implies $u(t) = u_1(t)e^{j\pi t}$. At the output, we have

$$y_p(t) = \text{Re}\left(y(t)e^{j2\pi f_c t}\right) = \text{Re}\left(y_2(t)e^{j2\pi(f_c-0.5)t}\right)$$

which implies $y_2(t) = y(t)e^{j\pi t}$. Thus, the conversion from $u_1$ to $u$ at the input, and the conversion from $y$ to $y_2$ at the output, requires identical blocks, as shown in Figure 16, where the complex

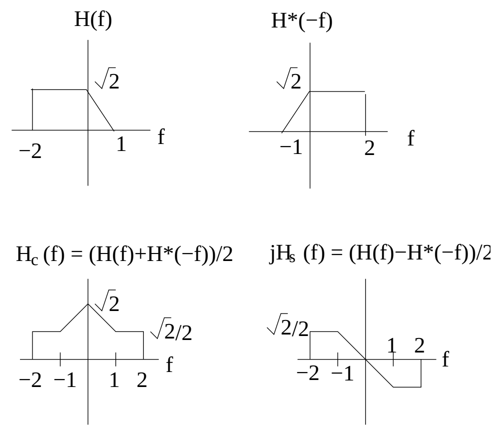



Figure 15: The transfer functions of the real baseband filters used in Problem 2.10(a).

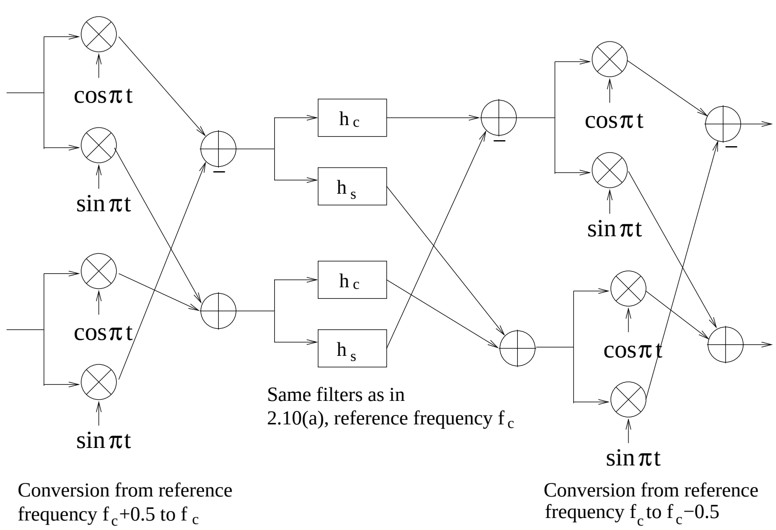



Figure 16: One possible design of real baseband operations for Problem 2.10(b).

operations are written out as real baseband operations. It is left as an exercise to work out alternative solutions in which the filters are expressed with respect to reference frequency $f_1$ (in which case the output must be converted from reference frequency $f_1$ to reference frequency $f_2$) or $f_2$ (in which case the input must be converted from reference frequency $f_1$ to reference frequency $f_2$). The baseband filters $h_c$ and $h_s$ in these cases will be different from those in (a).

**Problem 2.11:** We have $s_p(t) = \cos 2\pi f_c t = \frac{1}{2} e^{j2\pi f_c t} + \frac{1}{2} e^{-j2\pi f_c t}$. Thus, any filter around $\pm f_c$, no matter how narrow, measures a power of $\frac{1}{4}$. Furthermore, there is no power in any band that excludes $\pm f_c$. This implies that

$$S_{s_p}(f) = \frac{1}{4}\delta(f - f_c) + \frac{1}{4}\delta(f + f_c)$$

For the autocorrelation function, consider

$$s_p(t)s_p(t-\tau) = \cos 2\pi f_c t \cos 2\pi f_c (t-\tau) = \frac{1}{2}\cos 2\pi f_c \tau + \frac{1}{2}\cos(4\pi f_c t - 2\pi f_c \tau)$$

The second term is a sinusoid at $2f_c$ with no DC component, so that its time average is zero. We therefore obtain that

$$R_{s_p}(\tau) = \overline{s_p(t)s_p(t-\tau)} = \frac{1}{2}\cos 2\pi f_c \tau$$

Check that $R_{s_p}(\tau)$ and $S_{s_p}(f)$ form a Fourier transform pair.
(b) The complex envelope is $s(t) = \frac{1}{\sqrt{2}}$. This yields $S_s(f) = \frac{1}{2}\delta(f)$, since the measured power in any band around DC is $\frac{1}{2}$. Also, $s(t)s(t-\tau) = \frac{1}{2}$, so that

$$R_s(\tau) = \overline{s(t)s(t-\tau)} = \frac{1}{2}$$

Clearly, the relation (2.70) holds; that is,

$$S_{s_p}(f) = \frac{1}{2}\left(S_s(f - f_c) + S_s(f + f_c)\right)$$

**Problem 2.12:** (a) Yes. As shown in the text, we have

$$S_n(f) = 2S^+_{n_p}(f + f_c)$$

(b) Let us consider the autocorrelation function

$$\begin{aligned} R_n(\tau) = \overline{n(t)n^*(t-\tau)} &= \overline{\left(n_c(t) + jn_s(t)\right)\left(n_c(t-\tau) - jn_s(t-\tau)\right)} \\ &= R_{n_c}(\tau) + R_{n_s}(\tau) + j\left(-R_{n_c,n_s}(\tau) + R_{n_s,n_c}(\tau)\right) \\ &= R_{n_c}(\tau) + R_{n_s}(\tau) + j\left(-R_{n_s,n_c}(-\tau) + R_{n_s,n_c}(\tau)\right) \end{aligned}$$

Clearly, knowing $S_n(f)$, or equivalently, $R_n(\tau)$ does not provide enough information by itself for inferring the second order statistics for $n_c$ and $n_s$. However, if we knew, for example, that $R_{n_c}(\tau) = R_{n_s}(\tau)$, and that $R_{n_s,n_c}(\tau) = -R_{n_s,n_c}(-\tau)$, then we could indeed infer these quantities.
(c) For $n_p$ to be WSS, its mean should be constant, and its autocorrelation should only depend on time differences. The mean is given by

$$\mathbb{E}[n_p(t)] = \mathbb{E}[n_c(t)]\cos 2\pi f_c t - \mathbb{E}[n_s(t)]\sin 2\pi f_c t$$

In order for the mean to be independent of time, we must have $\mathbb{E}[n_c(t)] = \mathbb{E}[n_s(t)] \equiv 0$. We now compute the statistical autocorrelation function of $n_p$:

$$\begin{aligned}
R_{n_p}(t_1,t_2) &= \mathbb{E}\left[n_p(t_1)n_p(t_2)\right] \\
&= \mathbb{E}\left[\left(\sqrt{2}n_c(t_1)\cos 2\pi f_c t_1 - \sqrt{2}n_s(t_1)\sin 2\pi f_c t_1\right)\left(\sqrt{2}n_c(t_2)\cos 2\pi f_c t_2 - \sqrt{2}n_s(t_2)\sin 2\pi f_c t_2\right)\right]
\end{aligned}$$

We now multiply terms out and use the following trigonometric identities:

$$\begin{aligned}
2\cos 2\pi f_c t_1 \cos 2\pi f_c t_2 &= \cos 2\pi f_c(t_1-t_2) + \cos 2\pi f_c(t_1+t_2) \\
2\sin 2\pi f_c t_1 \sin 2\pi f_c t_2 &= \cos 2\pi f_c(t_1-t_2) - \cos 2\pi f_c(t_1+t_2) \\
-2\cos 2\pi f_c t_1 \sin 2\pi f_c t_2 &= \sin 2\pi f_c(t_1-t_2) - \sin 2\pi f_c(t_1+t_2) \\
-2\sin 2\pi f_c t_1 \cos 2\pi f_c t_2 &= -\sin 2\pi f_c(t_1-t_2) - \sin 2\pi f_c(t_1+t_2)
\end{aligned}$$

Substituting, we obtain

$$\begin{aligned}
R_{n_p}(t_1,t_2) &= \left[R_{n_c}(t_1,t_2) + R_{n_s}(t_1,t_2)\right]\cos 2\pi f_c(t_1-t_2) - \left[R_{n_s,n_c}(t_1,t_2) - R_{n_c,n_s}(t_1,t_2)\right]\sin 2\pi f_c(t_1-t_2) \\
&+ \left[R_{n_c}(t_1,t_2) - R_{n_s}(t_1,t_2)\right]\cos 2\pi f_c(t_1+t_2) - \left[R_{n_s,n_c}(t_1,t_2) + R_{n_c,n_s}(t_1,t_2)\right]\sin 2\pi f_c(t_1+t_2)
\end{aligned}$$

For $n_p$ to be WSS, the sinusoids with the $(t_1+t_2)$ terms must drop out, which requires that

$$R_{n_c}(t_1,t_2) \equiv R_{n_s}(t_1,t_2), \qquad R_{n_s,n_c}(t_1,t_2) \equiv -R_{n_c,n_s}(t_1,t_2)$$

Furthermore, in order for the final expression to depend on $t_1 - t_2$ alone, we must have that the second order statistics of $n_c$, $n_s$ depend on $t_1 - t_2$ alone. That is, $R_{n_c}(t_1,t_2) = R_{n_c}(t_1-t_2,0)$, $R_{n_c,n_s}(t_1,t_2) = R_{n_c,n_s}(t_1-t_2,0)$, and so on. That is, $n_c$ and $n_s$ must be jointly WSS, zero mean random processes, satisfying the following conditions (changing notation to express these second order statistics as a function of $\tau = t_1 - t_2$ alone):

$$R_{n_c}(\tau) = R_{n_s}(\tau), \quad R_{n_s,n_c}(\tau) = -R_{n_c,n_s}(\tau) = -R_{n_s,n_c}(-\tau) \tag{1}$$

Under these conditions, we obtain that

$$R_{n_p}(\tau) = 2R_{n_c}(\tau)\cos 2\pi f_c\tau - 2R_{n_s,n_c}(\tau)\sin 2\pi f_c\tau$$

Under the conditions (1), the autocorrelation function of the complex envelope $n$ can be easily computed, and is given by the following:

$$R_n(\tau) = \mathbb{E}[n(t)n^*(t-\tau)] = 2R_{n_c}(\tau) + 2jR_{n_s,n_c}(\tau)$$

**Problem 2.13:** Note the typo: what is specified in the displayed equation is the passband PSD $S_{n_p}(f)$.
(a) We have

$$S_n(f) = 2S_{n_p}^+(f+f_c) = N_0 I_{[-W/2,W/2]}(f)$$

(b) Note that $n_c$, $n_s$ satisfy the conditions for $n_p$ to be WSS derived in Problem 2.12:

$$R_{n_c}(\tau) = R_{n_s}(\tau), \quad R_{n_s,n_c}(\tau) = -R_{n_c,n_s}(\tau) = -R_{n_s,n_c}(-\tau) \equiv 0$$

since they are zero mean and i.i.d. Under these conditions, we have $n_p$, $n$ are WSS with

$$R_n(\tau) = 2R_{n_c}(\tau)$$

so that

$$S_{n_c}(f) = S_{n_s}(f) = \frac{1}{2}S_n(f) = \frac{N_0}{2}I_{[-W/2,W/2]}(f)$$

**Problem 2.14:** (a) Consider $v(t-a) = s(t-D-a)$, where $a$ is an arbitrary real number. We can write

$$D + a = kT + \tilde{D}$$

where $k$ is an integer, and $\tilde{D} = (D+a) \bmod T$ takes values in $[0,T]$. It is easy to see that, if $D$ is uniform over $[0,T]$ and independent of $s(t)$, then so is $\tilde{D}$. By the cyclostationarity of $s$, we know that $\tilde{s}(t) = s(t-kT)$ is statistically indistinguishable from $s(t)$. Furthermore, $\tilde{s}$ and $\tilde{D}$ are independent. Hence, $v(t-a) = \tilde{s}(t-\tilde{D})$ is statistically indistinguishable from $v(t) = s(t-D)$, since the joint distribution of $\tilde{s}$ and $\tilde{D}$ is the same as that of $s$ and $D$.
(b) The mean function is given by

$$m_v(t) = \mathbb{E}[v(t)] = \mathbb{E}[s(t-D)] = \frac{1}{T}\int_0^T m_s(t-\alpha)d\alpha$$

conditioning on $D = \alpha$ and then removing the conditioning. Making a change of variables $\nu = t - \alpha$, we obtain

$$m_v(t) = \frac{1}{T}\int_{t-T}^{t} m_s(\nu)d\nu = \frac{1}{T}\int_0^T m_s(\nu)d\nu$$

where the final equality follows from noting that $m_s$ is periodic with period $T$, so that its average over any interval of length $T$ is the same. Thus, the mean function $m_v(t)$ is a constant.
The autocorrelation function is given by

$$R_v(t_1,t_2) = \mathbb{E}[v(t_1)v^*(t_2)] = \mathbb{E}[s(t_1-D)v^*(t_2-D)] = \frac{1}{T}\int_0^T R_s(t_1-\alpha, t_2-\alpha)d\alpha$$

conditioning again on $D = \alpha$ and then removing the conditioning. Making a change of variables $t = t_2 - \alpha$, we obtain

$$R_v(t_1,t_2) = \frac{1}{T}\int_{t_2-T}^{t_2} R_s(t_1-t_2+t, t)dt = \frac{1}{T}\int_0^T R_s(t_1-t_2+t,t)dt$$

where the last equality follows from noting that $R_s(t+\tau,t)$ is periodic in $t$ with period $T$ by wide sense cyclostationarity, so that its average of any interval of length $T$ is the same. This implies that $R_v(t_1,t_2)$ depends only on the time difference $t_1 - t_2$. We have therefore shown that $v$ is WSS.
(c) (i) By cyclostationarity, $s(t)s^*(t-\tau)$ have the same statistics as $s(t+T)s^*(t+T-\tau)$.
(ii) The time-averaged autocorrelation is given by

$$\hat{R}_s(\tau) = \lim_{K\to\infty}\frac{1}{KT}\int_{-KT/2}^{KT/2} s(t)s^*(t-\tau)dt = \lim_{K\to\infty}\frac{1}{KT}\sum_{k=-K/2}^{K/2-1}\int_{kT}^{(k+1)T} s(t)s^*(t-\tau)dt$$

Setting $u = t - kT$ in the $k$th integral, we have

$$\hat{R}_s(\tau) = \lim_{K\to\infty}\frac{1}{KT}\sum_{k=-K/2}^{K/2-1}\int_0^T s(u+kT)s^*(u+kT-\tau)du$$

Assuming we can interchange limit and integral, we have

$$\hat{R}_s(\tau) = \frac{1}{T}\int_0^T \lim_{K\to\infty}\left[\frac{1}{K}\sum_{k=-K/2}^{K/2-1} s(u+kT)s^*(u+kT-\tau)\right]du$$

For each $u$, the random variables $\{s(u+kT)s^*(u+kT-\tau)\}$ are identically distributed, presumably with correlations dying off fast enough that the law of large numbers applies. Assuming it does, we have

$$\lim_{K\to\infty} \frac{1}{K} \sum_{k=-K/2}^{K/2-1} s(u+kT)s^*(u+kT-\tau) = \mathbb{E}[s(u)s^*(u-\tau)] = R_s(u, u-\tau)$$

In that case, we obtain that

$$\hat{R}_s(\tau) = \frac{1}{T}\int_0^T R_s(u, u-\tau)du = R_v(\tau)$$

That is, under suitable ergodicity assumptions, the time-averaged autocorrelation function of a cyclostationary process equals the statistical autocorrelation of its stationarized version.

**Problem 2.15:** The solution steps are clearly spelt out in the problem statement.

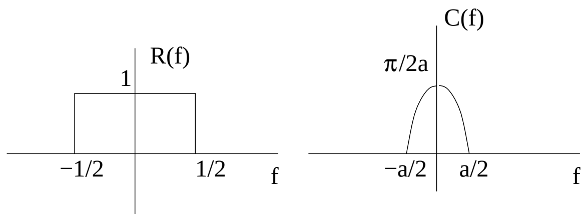



Figure 17: The spectra $R(f)$ and $C(f)$ for Problem 2.16.

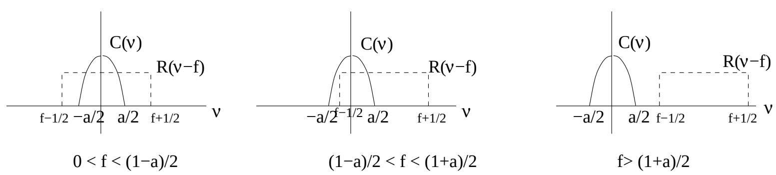



Figure 18: Convolution of $R(f)$ and $C(f)$ in Problem 2.16(b).

**Problem 2.16:** (a) The spectra are sketched in Figure 17.
(b) By the symmetry of $R$ and $C$, we have that $S = R * C$ is also symmetric. We therefore only evaluate the convolution for $f > 0$.

$$S(f) = \int R(f-\nu)C(\nu)d\nu = \int R(\nu - f)C(\nu)d\nu$$

using the symmetry of $R$. The range of integration is determined as shown in Figure 18. For $f \le (1-a)/2$, we have $R(\nu - f) = 1$ wherever $C(\nu) > 0$, so that

$$S(f) = \int_{-a/2}^{a/2} C(\nu)d\nu = 1$$

For $(1-a)/2 < f < (1+a)/2$, we have

$$S(f) = \int_{f-\frac{1}{2}}^{a/2} C(\nu)d\nu = \frac{\pi}{2a}\int_{f-\frac{1}{2}}^{a/2} \cos(\pi\nu/a)d\nu = \frac{1}{2}\left[1 - \sin\left(\frac{\pi}{a}(f-\frac{1}{2})\right)\right]$$

To see that this is indeed a raised cosine shape, set $\tilde{f} = \frac{\pi}{a}(f - (1-a)/2)$, so that $\frac{\pi}{a}(f - \frac{1}{2}) = \tilde{f} - \frac{\pi}{2}$. For $(1-a)/2 < f < (1+a)/2$, we have $0 < \tilde{f} < \frac{\pi}{2}$, and

$$S(\tilde{f}) = \frac{1}{2}\left[1 - \sin(\tilde{f} - \frac{\pi}{2})\right] = \frac{1}{2}\left[1 + \cos\tilde{f}\right]$$

For $f > (1+a)/2$, there is no overlap between $R(\nu - f)$ and $C(\nu)$, so that $S(f) = 0$.
(c) The time domain pulse $s(t) = r(t)c(t)$, where $r(t) = \mathrm{sinc}(t)$ and

$$c(t) = \int C(f)e^{j2\pi ft}df = \tfrac{\pi}{2a}\int_{-a/2}^{a/2}\tfrac{1}{2}\left(e^{j\pi f/a} + e^{-j\pi f/a}\right)e^{j2\pi ft}df$$

$$= \tfrac{\pi}{4a}\left\{\tfrac{e^{j(\frac{\pi}{a}+2\pi t)f}}{j(\frac{\pi}{a}+2\pi t)} + \tfrac{e^{j(-\frac{\pi}{a}+2\pi t)f}}{j(-\frac{\pi}{a}+2\pi t)}\right\}_{f=-a/2}^{f=a/2}$$

$$= \tfrac{\pi}{4a}\left\{\tfrac{e^{j(\frac{\pi}{2}+\pi at)} - e^{j(-\frac{\pi}{2}-\pi at)}}{j(\frac{\pi}{a}+2\pi t)} + \tfrac{e^{j(-\frac{\pi}{2}+\pi at)} - e^{j(\frac{\pi}{2}-\pi at)}}{j(-\frac{\pi}{a}+2\pi t)}\right\}$$

$$= \tfrac{\pi}{4a}\left\{\tfrac{2j\cos\pi at}{j(\frac{\pi}{a}+2\pi t)} + \tfrac{-2j\cos\pi at}{j(-\frac{\pi}{a}+2\pi t)}\right\}$$

$$= \tfrac{\cos\pi at}{1-4a^2t^2}$$

Thus, the time domain pulse is given by

$$s(t) = \mathrm{sinc}(t)\frac{\cos\pi at}{1-4a^2t^2}$$

(d) The raised cosine pulse $s(t)$ decays as $\frac{1}{t^3}$. Setting $g(t) = s(t/T)$, consider a linearly modulated waveform

$$x(t) = \sum_n b[n]g(t-nT) = \sum_n b[n]s((t-nT)/T) = \sum_n b[n]s(t/T - n)$$

Fixing $t$, we realize that $s(t/T - n)$ decays roughly as $1/n^3$. The convergence of $\sum_n 1/n^3$ implies that the sum in the preceding equation converges to a finite value for any $t$, assuming that the symbol magnitudes $|b[n]|$ are bounded.

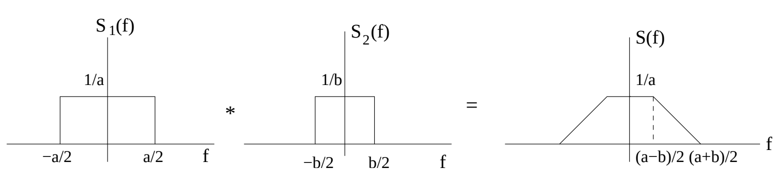



Figure 19: The spectrum $S(f)$ in Problem 2.17(a).

**Problem 2.17:** (a) Letting $s_1(t) = \mathrm{sinc}(at)$, $s_2(t) = \mathrm{sinc}(bt)$, we have $s(t) = s_1(t)s_2(t) \leftrightarrow S(f) = (S_1 * S_2)(f)$. The spectra $S_1$. $S_2$ and $S$ are sketched in Figure 19.
(b) The symbol rate is given by

$$\frac{1}{T} = \frac{1200 \text{ bits/sec}}{\log_2 4 \text{bits/symbol}} = 600 \text{ symbols/sec}$$

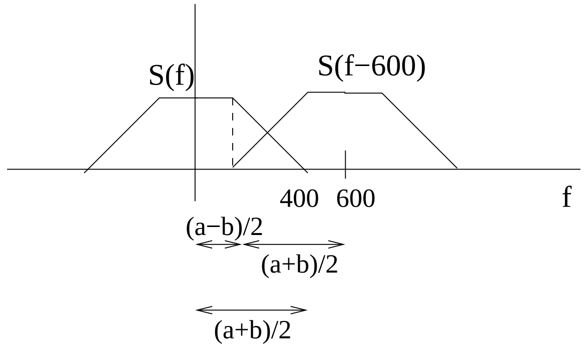



Figure 20: Choosing $a$ and $b$ for Nyquist signaling in Problem 2.17(b).

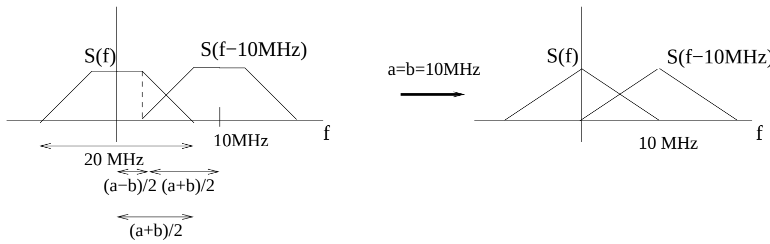



Figure 21: Choosing $a$ and $b$ for Nyquist signaling in Problem 2.17(c).

As shown in Figure 20, in order to exactly fill the channel bandwidth, we need $\frac{a+b}{2} = 400$, and in order to satisfy the Nyquist criterion for the given symbol rate,

$$\frac{a-b}{2} + \frac{a+b}{2} = 600$$

which yields $a = 600$, $b = 200$.
(c) We now have a passband system of bandwidth 20 MHz, and a symbol rate

$$\frac{1}{T} = \frac{60 \text{ Mbps}}{\log_2 64 \text{ bits/symbol}} = 10 \text{ Msymbols/sec}$$

Figure 21 depicts the bandwidth occupancy in complex baseband (note that 20 MHz is the 2-sided bandwidth for the complex baseband system). We now have $\frac{a+b}{2} = 10MHz$ and

$$\frac{a-b}{2} + \frac{a+b}{2} = 10MHz$$ due to Ts

which yields $a = b = 10MHz$.
(d) Since sinc($at$), sinc($bt$) both decay as $1/t$, the pulse $s(t)$ exhibits a $1/t^2$ decay. Since $\int \frac{1}{t^2} dt$ is finite, the superposition of waveforms of the form $s(t - nT)$ adds up to a finite value at any point of time.

**Problem 2.18:** The pulse $p(t)$ is sketched in Figure 22.
(a) **True**, since $p(kT) = \delta_{k0}$.
(b) **False.** For $p(t)$ to be square root Nyquist, we must have $\int p(t)p^*(t - kT)dt = 0$ for any nonzero integer $k$. This is clearly not true, since $\int p(t)p^*(t - T)dt > 0$.

**Problem 2.19:** The spectrum $P(f)$ is sketched in Figure 23. The time domain pulse is therefore of the form $p(t) = \text{sinc}(at)\text{sinc}(bt)$ (up to scale factor), where $(a+b)/2 = 1.25MHz$ and $(a-b)/2 =$

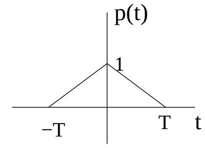



Figure 22: The pulse for Problem 2.18.

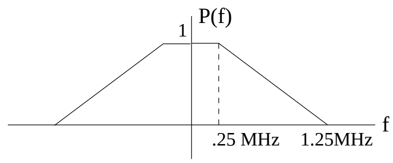



Figure 23: The spectrum $P(f)$ for Problem 2.19.

$0.25MHz$, so that $a = 1.5$ MHz and $b = 1$ MHz.
(a) **True.** 3 Mbps using 8PSK corresponds to a symbol rate of $3/\log_2 8 = 1$ Msymbols/sec. Since $b$ is 1 MHz, the sinc$(bt)$ term gives the desired zeroes at integer multiples of $T$.
(b) **True.** 4.5 Mbps using 8PSK corresponds to a symbol rate of 1.5 Msymbols/sec. Since $a$ is 1.5 MHz, the sinc$(at)$ term gives the desired zeroes at integer multiples of $T$.

**Problem 2.20: True,** since $p(t)p^*(t-kT) \equiv 0$ for $k$ a nonzero integer, which implies that $\int p(t)p^*(t-kT)dt = 0$.

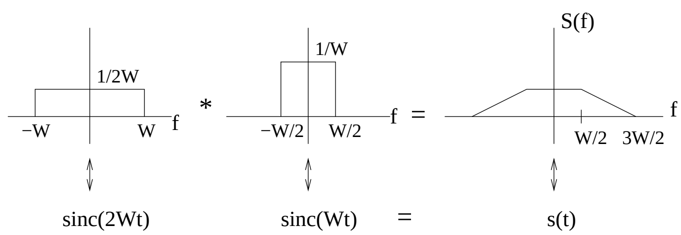



Figure 24: The product of two sinc pulses gives a trapezoidal spectrum in Problem 2.21(c).

**Problem 2.21:** We have

$$r_n = y((n-1)T) = \sum_{k=1}^{100} b_k s((n-1)T - (k-1)T) = \sum_{k=1}^{100} b_k s((n-k)T)$$

(a) For $r_n = b_n$, we need that $s(kT) = \text{sinc}(2WkT) = \delta_{k0}$, which requires that $2WT$ is an integer, or $T = \frac{L}{2W}$, for $L$ a positive integer. The smallest value of $T$, and hence the fastest rate, corresponds to $L = 1$.
(b) Choosing $T = \frac{1}{2W}$, we have, for timing offset $0.25T$, that

$$r_n = y((n-1)T + 0.25T) = \sum_{k=1}^{100} b_k s((n-k)T + 0.25T) = \sum_{k=1}^{100} b_k \text{sinc}(n-k+0.25)$$

Thus,

$$r_{50} = \sum_{k=1}^{100} b_k \text{sinc}(50-k+0.25) = \sum_{m=-50}^{49} b_{50-m} \text{sinc}(m+0.25)$$

For $m > 0$, the sign of $\text{sinc}(m + 0.25)$ is $(-1)^m$. On the other hand, the sign of $b_{50-m}$ is $(-1)^{50-m-1} = (-1)^{m+1}$, so that the term $b_{50-m}\text{sinc}(m + 0.25)$ has negative sign. Similarly, for $m < 0$, the sign of $\text{sinc}(m+0.25)$ is $(-1)^{m+1}$, whereas the sign of $b_{50-m}$ is $(-1)^{50-m} = (-1)^m$, so that the product $b_{50-m}\text{sinc}(m+0.25)$ again has negative sign. Thus, the ISI terms for $m \neq 0$ are all negative. On the other hand, the desired term for $m = 0$, $b_{50}\text{sinc}(0.25) = \text{sinc}(0.25) = .90$ is positive. Numerical summation shows that the sum of the ISI terms is -2.06, which is significantly larger in magnitude than the desired term. We therefore obtain that $r_{50} = -1.16$ has a sign opposite to that of $b_{50}$.
(c) For $s(t) = \text{sinc}(Wt)\text{sinc}(2Wt)$, the two smallest values of $T$ for which the pulse is Nyquist are $T = \frac{1}{2W}$ and $T = \frac{1}{W}$.
(d) Choosing $T = \frac{1}{2W}$, we have, for timing offset $0.25T$, that

$$r_n = y((n-1)T+0.25T) = \sum_{k=1}^{100} b_k s((n-k)T+0.25T) = \sum_{k=1}^{100} b_k \text{sinc}(n-k+0.25)\text{sinc}((n-k+0.25)/2)$$

which gives

$$r_{50} = \sum_{k=1}^{100} b_k \text{sinc}(50-k+0.25)\text{sinc}((50-k+0.25)/2) = \sum_{m=-50}^{49} b_{50-m}\text{sinc}(m+0.25)\text{sinc}((m+0.25)/2)$$

For $m > 0$, the sum of the ISI terms comes to -0.07. For $m > 0$, the sum of the ISI terms is -0.24. The desired term, on the other hand, is 0.8774. Thus, the sum of the ISI and desired term is $r_{50} = 0.57$, which has the same sign as $b_{50}$.
Note that the bit sequence in (b) no longer creates the worst-case ISI, because of the signs of the ISI coefficients are now different. If we chose the bit sequence to create the worst-case ISI, then the ISI terms for $m > 0$ would sum to $-0.13$, and the ISI terms for $m < 0$ would sum to $-0.30$, which is still not large enough in magnitude to switch the sign of the desired term. In this case, we would obtain $r_{50} = 0.45$, which is still the same sign as $b_{50}$. This clearly illustrates how the faster decay in the signaling pulse translates to increased robustness to timing mismatch.
(d) The spectrum of the signaling pulse in (c) is sketched in Figure 24. Clearly, the excess bandwidth is 50%. As we have seen in (c), the faster time decay associated with the excess bandwidth significantly reduces the severity of the ISI due to timing mismatch.

**Problem 2.22:** (a) We have

$$s(t - kT) = \sum_{n=-\infty}^{\infty} b[n]p(t - kT - nT) = \sum_{m=-\infty}^{\infty} b[m - k]p(t - mT)$$

setting $m = k + n$. Comparing with the expression for $s(t) = \sum_n b[n]p(t - nT)$, we note that the only difference is that the sequence $\{b[n]\}$ has been replaced by its shift, $\{b[n - k]\}$. These two sequences are statistically identical if $\{b[n]\}$ is stationary, which means that the random process $s(t)$ and the random process $s(t - kT$ are statistically identical. This proves the cyclostationarity of $s$ with respect to the interval $T$.
(b) For WSS $\{b[n]\}$, we have $\mathbb{E}[b[n]] \equiv m_b$ (constant mean) and $\mathbb{E}[b[n]b^*[m]] = R_b[n - m]$. Consider the mean function

$$\begin{aligned}\mathbb{E}[s(t - T)] &= \textstyle\sum_{n=-\infty}^{\infty} \mathbb{E}[b[n]]p(t - T - nT) = m_b \sum_{n=-\infty}^{\infty} p(t - (n + 1)T) \\ &= m_b \textstyle\sum_{m=-\infty}^{\infty} p(t - mT) = \mathbb{E}[s(t)]\end{aligned}$$

setting $m = n + 1$. Thus, the mean function of $s(t)$ is periodic with period $T$. Now, consider the autocorrelation function

$$\begin{aligned}R_s(t_1 - T, t_2 - T) &= \mathbb{E}\left[\textstyle\sum_n \sum_m b[n]p(t_1 - T - nT)b^*[m]p^*(t_2 - T - mT)\right] \\ &= \textstyle\sum_n \sum_m R_b[n - m]p(t_1 - (n + 1)T)p^*(t_2 - (m + 1)T) \\ &= \textstyle\sum_n \sum_m R_b[n - m]p(t_1 - nT)p^*(t_2 - mT) = R_s(t_1, t_2)\end{aligned}$$

replacing $n+1$ by $n$, and $m+1$ by $m$ in the variables being summed. This completes the proof that $s(t)$ is wide sense cyclostationary.
(c) We know from Problem 2.14(b) that $v$ is WSS with autocorrelation function

$$R_v(\tau) = \frac{1}{T}\int_0^T R_s(\tau+t,t)dt$$

Now,

$$\begin{aligned} R_s(\tau+t,t) &= \mathbb{E}\left[\sum_n \sum_m b[n]p(t+\tau-nT)b^*[m]p^*(t-mT)\right] \\ &= \sum_{n,m} R_b[n-m]p(t+\tau-nT)p^*(t-mT) = \sum_{n,k} R_b[k]p(t+\tau-nT)p^*(t-nT+kT) \end{aligned}$$

setting $k=n-m$ and eliminating $m=n-k$. Substituting into the expression for $R_v$, we get

$$R_v(\tau) = \frac{1}{T}\sum_{n,k} R_b[k]\int_0^T p(t+\tau-nT)p^*(t-nT+kT)dt = \frac{1}{T}\sum_k R_b[k]\sum_n \int_{-nT+\tau}^{(-n+1)T+\tau} p(t)p^*(t+kT-\tau)dt$$

where we have substituted $t-nT+\tau$ by $t$ in the $n$th integral. We therefore obtain that

$$R_v(\tau) = \frac{1}{T}\sum_k R_b[k]\int_{-\infty}^{\infty} p(t)p^*(t+kT-\tau)dt$$

Defining the autocorrelation of the signaling pulse, $R_p(u) = \int p(t)p^*(t-u)dt \leftrightarrow |P(f)|^2$, we have

$$R_v(\tau) = \frac{1}{T}\sum_k R_b[k]R_p(\tau-kT)$$

Taking the Fourier transform, we obtain that

$$S_v(f) = \frac{1}{T}\sum_k R_b[k]|P(f)|^2 e^{-j2\pi fkT} = \frac{|P(f)|^2}{T}\sum_k R_b[k]e^{-j2\pi fkT}$$

This yields the desired result

$$S_v(f) = \frac{|P(f)|^2}{T}S_b(e^{j2\pi fT})$$

(d) (i) $b[n]$ takes values $0, \pm 2$.
(ii) $b[n] = a[n] - a[n-1]$ is zero mean WSS with

$$R_b[m] = \begin{cases} 2, & m=0 \\ -1, & m=\pm 1 \\ 0, & \text{else} \end{cases}$$

so that $S_b(z) = 2-(z+z^{-1})$. For $f=0$, $S_b(e^{j2\pi fT}) = S_b(1) = 0$. Thus, $S_v(0)=0$. That is, the line code creates a spectral null at DC.
(iii) For $b[n] = a[n] + ka[n-1]$, we have

$$R_b[m] = \begin{cases} 1+k^2, & m=0 \\ k, & m=\pm 1 \\ 0, & \text{else} \end{cases}$$

so that $S_b(z) = 1+k^2+k(z+z^{-1})$. In order to have a spectral null at $f=\frac{1}{2T}$, we must have $S_b(e^{j2\pi fT}) = S_b(-1) = 1+k^2-2k = 0$. That is, $(k-1)^2=0$, or $k=1$. This corresponds to the line code $b[n] = a[n] + a[n-1]$.

**Problem 2.23:** The steps are clearly laid out in the problem statement. We therefore state the final result

$$S_s(f) = \frac{|P(f)|^2}{T} S_{\tilde{b}}(e^{j2\pi fT}) + \sum_k |a[k]|^2 \delta\left(f - \frac{k}{T}\right)$$

where $S_{\tilde{b}}(z) = \sum_k C_b[k] z^{-k}$ is the PSD of the zero mean version of the symbol sequence, and where $\{a[k]\}$ is the Fourier series for $\bar{s}(t) = \bar{b} \sum_n p(t - nT)$.
(d) For the unipolar NRZ waveform, $p(t) = I_{[0,T]} \leftrightarrow P(f) = \mathrm{sinc}(fT)e^{-\pi fT}$. The symbols $\{b[n]\}$ are i.i.d., taking values 0 and 1 with equal probability, so that $\bar{b} = \frac{1}{2}$ and $C_b[k] = \frac{1}{4}\delta_{k0}$. We therefore have $\bar{s}(t) \equiv \frac{1}{2}$, so that $a[k] = \frac{1}{2}\delta_{k0}$. The overall PSD is therefore

$$S_s(f) = \frac{T}{4}\mathrm{sinc}^2(fT) + \frac{1}{2}\delta(f)$$

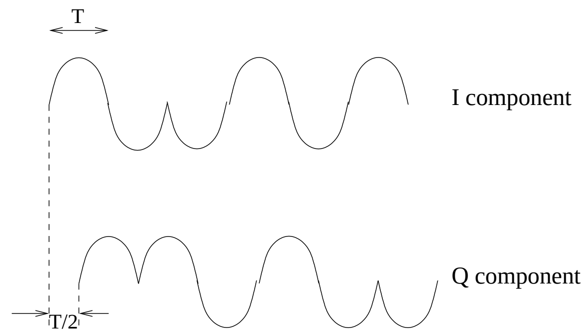



Figure 25: The I and Q components of a typical realization of an MSK waveform.

**Problem 2.24:** (a) As shown in Figure 25, the I and Q components of an MSK waveform are offset from each other by $\frac{T}{2}$.
(b) Over any interval $[mT/2, (m+1)T/2]$, the I and Q components are sinusoids offset in phase by $\pm\frac{\pi}{2}$. Since $\sin^2(x) + \cos^2(x) = 1$, the envelope is constant.
(c) The I and Q components have identical PSDs (the time offset does not affect the PSD). The I and Q components are independent, since the bits sent over the I and Q channels are independent. The PSDs therefore add up, and we obtain that

$$S_s(f) = \mathbb{E}[|b_c[n]|^2]\frac{|G_{TX}(f)|^2}{T} + \mathbb{E}[|b_s[n]|^2]\frac{|G_{TX}(f)|^2}{T} = 2\frac{|G_{TX}(f)|^2}{T}$$

The PSD is not changed by considering the symmetric version of the transmitted pulse: $g_{TX}(t) = \cos \pi t/T I_{[-T/2,T/2]}$. Setting $T = 1$ for convenience, we obtain

$$\begin{aligned}
G_{TX}(f) &= \int_{-1/2}^{1/2} \frac{e^{j\pi t} + e^{-j\pi t}}{2} e^{-j2\pi ft} dt \\
&= \frac{1}{2}\left[\frac{e^{j\pi(1-2f)t}}{j\pi(1-2f)} + \frac{e^{-j\pi(1+2f)t}}{-j\pi(1+2f)}\right]_{t=-1/2}^{t=1/2} \\
&= \frac{1}{2}\left[\frac{e^{j\pi/2}e^{-j\pi f} - e^{-j\pi/2}e^{j\pi f}}{j\pi(1-2f)} + \frac{e^{-j\pi/2}e^{-j\pi f} - e^{j\pi/2}e^{j\pi f}}{-j\pi(1+2f)}\right] \\
&= \frac{1}{2\pi}\left[\frac{e^{-j\pi f} + e^{j\pi f}}{1-2f} + \frac{e^{-j\pi f} + e^{j\pi f}}{1+2f}\right] \\
&= \frac{2}{\pi}\frac{\cos \pi f}{1-4f^2}
\end{aligned}$$

When we timescale by $T$, we have

$$G_{TX}(f) = \frac{2T}{\pi} \frac{\cos \pi f T}{1 - 4f^2T^2}$$

The PSD is given by

$$S_s(f) = \frac{8T^2}{\pi^2} \frac{\cos^2 \pi f T}{(1 - 4f^2T^2)^2} \quad \textbf{PSD of MSK}$$

(d) We can set $T = 1$ once more to compute 99% energy containment bandwidth:

$$\int_{-B_1/2}^{B_1/2} |G_{TX}(f)|^2 df = 0.99 \int_{-\infty}^{\infty} |G_{TX}(f)|^2 df = 0.99 \int_{-\infty}^{\infty} |g_{TX}(t)|^2 dt = 0.99/2$$

Using symmetry, we arrive at the following equation (which must be numerically solved):

$$\int_0^{B_1/2} \left( \frac{2}{\pi} \frac{\cos \pi f}{1 - 4f^2} \right)^2 df = 0.99/4$$

This yields $B_1 = 1.2$. This is much better than the corresponding result $B_1 = 10.2$ for a rectangular pulse. Of course, for signaling at rate $\frac{1}{T}$, the bandwidth simply scales to $\frac{B_1}{T}$.

(e) In order to normalize the area under the PSD curve, we should fix the energy of the transmit pulse to be the same in each case. Setting $T = 1$, for MSK as considered above, the energy is $\frac{1}{2}$. A rectangular pulse of the same energy is given by $g_{TX}(t) = \frac{1}{\sqrt{2}} I_{[-1/2,1/2]} \leftrightarrow \frac{1}{\sqrt{2}} \mathrm{sinc}(f)$. In each case, $S_s(f) = 2|G_{TX}(f)|^2$, so that

$$S_s(f) = \begin{cases} \frac{8}{\pi^2} \frac{\cos^2 \pi f}{(1-4f^2)^2} & \textbf{MSK} \\ \mathrm{sinc}^2(f) & \textbf{OQPSK} \end{cases}$$

Plotting these yields Figure 2.17.

**Problem 2.25:** For $f_0, f_1 \gg \frac{1}{T}$, the timelimited waveforms $s_0$ and $s_1$ are approximately passband. Using $f_0$ as the frequency reference, we can rewrite $s_0$ and $s_1$ as

$$s_0 = \mathrm{Re}\left( e^{j(2\pi f_0 t + \phi_0)} I_{[0,T]} \right) = \sqrt{2} \mathrm{Re} \left( \frac{1}{\sqrt{2}} e^{j\phi_0} I_{[0,T]} e^{j2\pi f_0 t} \right)$$

$$s_0 = \mathrm{Re}\left( e^{j(2\pi f_1 t + \phi_1)} I_{[0,T]} \right) = \sqrt{2} \mathrm{Re} \left( \frac{1}{\sqrt{2}} e^{j(2\pi (f_1 - f_0) t + \phi_1)} I_{[0,T]} e^{j2\pi f_0 t} \right)$$

so that the complex envelopes can be read off as

$$\tilde{s}_0 = \frac{1}{\sqrt{2}} e^{j\phi_0} I_{[0,T]}(t), \quad \tilde{s}_1 = \frac{1}{\sqrt{2}} e^{j(2\pi(f_1 - f_0)t + \phi_1)} I_{[0,T]}(t)$$

We know that $\langle s_0, s_1 \rangle = \mathrm{Re} \langle \tilde{s}_0, \tilde{s}_1 \rangle$. Now,

$$\langle \tilde{s}_0, \tilde{s}_1 \rangle = \int \tilde{s}_0(t) \tilde{s}_1^*(t) dt = \tfrac{1}{2} \int_0^T e^{j\phi_0} e^{-j(2\pi(f_1 - f_0)t + \phi_1)}$$

$$= \tfrac{1}{2} e^{j(\phi_0 - \phi_1)} \left. \tfrac{e^{j2\pi(f_0 - f_1)t}}{j2\pi(f_0 - f_1)} \right|_0^T = \tfrac{1}{2} e^{j(\phi_0 - \phi_1)} \tfrac{e^{j2\pi(f_0 - f_1)T} - 1}{j2\pi(f_0 - f_1)}$$

$$= \tfrac{1}{2} e^{j(\phi_0 - \phi_1)} \tfrac{\cos(2\pi(f_0 - f_1)T) - 1 + j \sin(2\pi(f_0 - f_1)T)}{j2\pi(f_0 - f_1)}$$

(a) For $\phi_0 = \phi_1 = 0$, we have

$$\langle s_0, s_1 \rangle = \mathrm{Re}\left(\langle \tilde{s}_0, \tilde{s}_1 \rangle\right) = \frac{1}{2} \frac{\sin(2\pi(f_0 - f_1)T)}{2\pi(f_0 - f_1)T)}$$

For the signals to be orthogonal, we need $\sin(2\pi(f_0 - f_1)T) = 0$, and the smallest frequency separation for which this happens is given by

$$2\pi(f_0 - f_1)T = \pm\pi$$

That is,

$$|f_0 - f_1| = \frac{1}{2T}$$

(b) For arbitrary $\phi_0$, $\phi_1$, $\mathrm{Re}\left(\langle \tilde{s}_0, \tilde{s}_1 \rangle\right) = 0$ if and only if

$$e^{j2\pi(f_0 - f_1)T} - 1 = 0$$

(set $\phi_0 - \phi_1 = 0$ and $\phi_0 - \phi_1 = \frac{\pi}{2}$ to see that both the real and imaginary parts of the preceding expression must be zero for orthogonality). The minimum frequency separation for which this condition is satisfied is

$$2\pi(f_0 - f_1)T = \pm 2\pi$$

which yields

$$|f_0 - f_1| = \frac{1}{T}$$

Thus, the frequency spacing required for a noncoherent orthogonal system is twice that required for a coherent system.

**Problem 2.26:** (a) The Walsh-Hadamard codes for 8-ary orthogonal signaling are given by

$$\mathbf{H}_3 = \begin{pmatrix} \mathbf{H}_2 & \mathbf{H}_2 \\ \mathbf{H}_2 & -\mathbf{H}_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{pmatrix}$$

(b) Omitted.
(c) We need 8 complex dimensions for 8-ary orthogonal signaling, and the spectral efficiency is $\frac{\log_2 8}{8} = \frac{3}{8}$ bps/Hz. For biorthogonal signaling, we can fit 16 signals into the same number of dimensions, so that the spectral efficiency becomes $\frac{\log_2 16}{8} = \frac{4}{8}$ bps/Hz. The fractional increase in spectral efficiency is 33%, as computed below:

$$\frac{\frac{4}{8} - \frac{3}{8}}{\frac{3}{8}} = \frac{1}{3}$$

**Problem 2.27:** For QPSK (2 bits/symbol), the symbol rate is 5 Msymbols/sec. For an excess bandwidth of 50%, the bandwidth is $5(1 + 0.5) = 7.5$ MHz.
For 64-QAM (6 bits/symbol), the symbol rate is 10/6 Msymbols/sec, and the bandwidth is 10/6 $(1 + 0.5) = 2.5$ MHz.

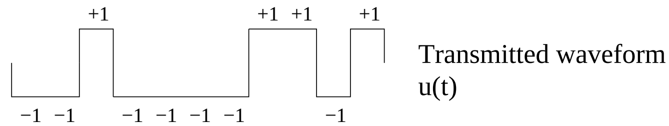



Figure 26: The complex baseband transmitted waveform for Problem 2.28. Only the I component is used.

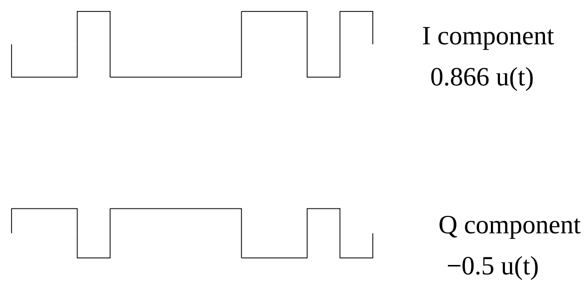



Figure 27: I and Q components in Problem 2.28 after channel phase shift.

For 64-ary noncoherent orthogonal modulation, we have $\frac{\log_2 64}{64} = 6/64$ bits/symbol, so that the bandwidth is $10 \times 64/6 \times (1 + 0.5) = 160$ MHz.

**Problem 2.28:** (a) 0 corresponds to no transition, 1 to a sign change. The transmitted symbol sequence $\{b[n], n = 0, 1, ..., 10\}$ is therefore given by $-1, -1, +1, -1, -1, -1, -1, +1, +1, -1, +1$.
(b) Only the I component is used. The transmitted waveform $u(t)$ is depicted in Figure 26.
(c) The channel phase shift multiplies the complex waveform by $e^{-\pi/6} = \sqrt{3}/2 - j/2$, so that the The received waveform $s(t) = e^{-j\pi/6}u(t) = sqrt3/2u(t) - j/2u(t)$, so that both the I and Q components are scaled versions of $u(t)$, as shown in Figure 27.
(d) The received sample $r[n] = e^{-\pi/6}b[n]$.
(e) We have $r[2]r^*[1] = b[2]b[1] = -1$, which corresponds to $a[2] = 1$ (sign change).

**Problem 2.29:** Let $\theta[n] = \arg(b[n]) \in \{\pm\frac{\pi}{4}, \pm\frac{3\pi}{4}\}$. We know that $\theta[0] = -\frac{\pi}{4}$. For the given information sequence $00, 11, 01, 10, 10, 01, 11, 00, 01, 10$, we obtain, based on Figure 2.22, that $\theta[n], n = 1, 2, ...$ is given by

$$-\frac{\pi}{4}, \frac{3\pi}{4}, \frac{\pi}{4}, \frac{3\pi}{4}, -\frac{3\pi}{4}, \frac{3\pi}{4}, -\frac{\pi}{4}, -\frac{\pi}{4}, -\frac{3\pi}{4}, -\frac{\pi}{4}$$

(b) Consider

$$r[2]r^*[1] = (1 + j)(2 + j) = 1 + 3j$$

the argument of which is closest to a phase difference of $\frac{\pi}{2}$, which corresponds to the information bits 10. Note that this would be an error, because the information bits corresponding to this transition are 11.

# Solutions to Chapter 3 Problems

*Fundamentals of Digital Communication*

**Problem 3.1:**
**(a)** The joint density must integrate to zero, hence we must have

$$1 = K \int_0^\infty \int_0^\infty e^{-(2x^2+y^2)/2} dx dy + K \int_{-\infty}^0 \int_{-\infty}^0 e^{-(2x^2+y^2)/2} dx dy = 2K \int_0^\infty \int_0^\infty e^{-(2x^2+y^2)/2} dx dy$$

where we have used symmetry. The integrals in $x$ and $y$ separate out, and we have

$$\int_0^\infty e^{-(2x^2)/2} dx = \sqrt{2\pi v_1^2} \int_0^\infty \frac{e^{-x^2/(2v_1^2)}}{2\pi v_1^2} dx = \sqrt{2\pi v_1^2}\, \frac{1}{2} = \frac{\sqrt{\pi}}{2}$$

massaging the $x$ integrand into an $N(0, v_1^2)$ density, with $v_1^2 = \frac{1}{2}$. Similarly, we can massage the $y$ integrand into an $N(0, v_2^2)$ density with $v_2^2 = 1$ to get

$$\int_0^\infty e^{-y^2/2} dy = \sqrt{2\pi v_2^2} \frac{1}{2} = \frac{\sqrt{2\pi}}{2}$$

We therefore have $1 = 2K \frac{\sqrt{\pi}}{2} \frac{\sqrt{2\pi}}{2}$, or $K = \sqrt{2}/\pi$.
**(b)** The marginal density of $X$ is

$$p(x) = \int p(x,y) dy = \begin{cases} \int_0^\infty e^{-(2x^2+y^2)/2} dy, & x \geq 0 \\ \int_{-\infty}^0 e^{-(2x^2+y^2)/2} dy, & x < 0 \end{cases}$$

By symmetry, the $y$ integrals evaluate to the same answer for the two cases above, so that $p(x) \sim e^{-x^2}$. Thus, $X \sim N(0, \frac{1}{2})$ (the constant must evaluate out to whatever is needed for $p(x)$ to integrate to one. A similar reasoning shows that $Y \sim N(0, 1)$.
**(c)** The event $X^2 + X > 2$ can be written as

$$X^2 + X - 2 = (X + 2)(X - 1) > 0$$

which happens if $X + 2 > 0, X - 1 > 0$, or $X + 2 < 0, X - 1 < 0$. That is, it happens if $X > 1$ or $X < -2$. Thus,

$$P[X^2 + X > 2] = P[X > 1] + P[X < -2] = Q\left(\frac{1-0}{\sqrt{1/2}}\right) + \Phi\left(\frac{-2-0}{\sqrt{1/2}}\right) = Q(\sqrt{2}) + Q(2\sqrt{2})$$

where we have used $X \sim N(0, 1/2)$ and $\Phi(-x) = Q(x)$.
**(d)** $X$, $Y$ are not jointly Gaussian, since the probability mass is constrained to two of the four quadrants, unlike the joint Gaussian density, for which the probability mass is spread over the entire plane.
**(e)** If $X > 0$, then $Y > 0$ (even though $Y$ can take both positive and negative values). Hence $X$ and $Y$ cannot be independent.
**(f)** From the marginals in (b), we know that $\mathbb{E}[X] = \mathbb{E}[Y] = 0$. However, $\mathbb{E}[XY] > 0$, since all the probability mass falls in the region $xy > 0$. Thus, $\text{cov}(X, Y) = \mathbb{E}[XY] - \mathbb{E}[X]\mathbb{E}[Y] > 0$. So $X$, $Y$ are not uncorrelated.
**(g)** The conditional density is $p(x|y) = p(x, y)/p(y)$. If $y > 0$, this evaluates to $p(x|y) =$

$k_1 e^{-x^2} I_{x \geq 0}$. If $y < 0$, it evaluates to $p(x|y) = k_1 e^{-x^2} I_{x<0}$. Since the probability mass of the conditional density is constrained to part of the real line, it is not Gaussian.

**Problem 3.2:**
**(a)** We have

$$\text{cov}(Y_1, Y_2) = \text{cov}(X_1 + 2X_2, -X_1 + X_2) = -\text{cov}(X_1, X_1) - \text{cov}(X_1, X_2) + 2\text{cov}(X_2, X_2) = -1 + 1 + 8 = 8$$

**(b)** $\mathbf{Y} = (Y_1, Y_2)^T = \mathbf{AX}$, where

$$\mathbf{A} = \begin{pmatrix} 1 & 2 \\ -1 & 1 \end{pmatrix}$$

Thus, we obtain $\mathbf{Y} \sim N(\mathbf{m}_Y, \mathbf{C}_Y)$

$$\mathbf{m}_Y = \mathbf{Am} = \begin{pmatrix} 4 \\ -1 \end{pmatrix}, \quad \mathbf{C}_Y = \mathbf{ACA}^T = \begin{pmatrix} 13 & 8 \\ 8 & 7 \end{pmatrix}$$

Plug these into the standard form of the joint Gaussian density.
**(c)** $P[Y_1 > 2Y_2 + 1] = P[Y_1 - 2Y_2 > 1]$, where $Z = Y_1 - 2Y_2$ is Gaussian with mean $\mathbb{E}[Z] = \mathbb{E}[Y_1] - 2\mathbb{E}[Y_2] = 6$ and variance

$$\text{var}(Z) = \text{cov}(Y_1 - 2Y_2, Y_1 - 2Y_2) = \text{cov}(Y_1, Y_1) + 4\text{cov}(Y_2, Y_2) - 4\text{cov}(Y_1, Y_2) = 9$$

Thus,

$$P[Y_1 > 2Y_2 + 1] = Q\left(\frac{1 - \mathbb{E}[Z]}{\sqrt{\text{var}(Z)}}\right) = Q(\frac{1-6}{\sqrt{9}}) = 1 - Q(5/3)$$

**Problem 3.3:**
**(a)** Rewrite

$$Q(x) = \int_x^\infty \frac{e^{-t^2/2}}{\sqrt{2\pi}}\, dt = e^{-x^2/2} \int_x^\infty \frac{e^{-(t^2-x^2)/2}}{\sqrt{2\pi}}\, dt$$

For $t \geq x \geq 0$, we have

$$t^2 - x^2 = (t+x)(t-x) \geq (t-x)^2$$

Plugging into the integrand, we obtain

$$Q(x) \leq e^{-x^2/2} \int_x^\infty \frac{e^{-(t-x)^2/2}}{\sqrt{2\pi}}\, dt = \frac{1}{2} e^{-x^2/2}$$

proving the desired result.
**(b)** Rewrite

$$\sqrt{2\pi} Q(x) = \int_x^\infty [t e^{-t^2/2}] \frac{1}{t}\, dt$$

Integrate by parts to get

$$\begin{aligned} \sqrt{2\pi} Q(x) &= -\tfrac{1}{t} e^{-t^2/2}|_x^\infty - \int_x^\infty [-e^{-t^2/2}] \tfrac{-1}{t^2}\, dt \\ &= \tfrac{1}{x} e^{-x^2/2} - \int_x^\infty \tfrac{e^{-t^2/2}}{t^2}\, dt \end{aligned} \qquad (1)$$

Since the integral on the last line of (1) is nonnegative, we obtain the upper bound $\sqrt{2\pi} Q(x) \leq \frac{1}{x} e^{-x^2/2}$. Moreover, integrating further by parts, we obtain that

$$\int_x^\infty \frac{e^{-t^2/2}}{t^2}\, dt = -\frac{1}{t^3} e^{-t^2/2}|_x^\infty - \int_x^\infty \frac{3}{t^4} e^{-t^2/2}\, dt \leq \frac{1}{x^3} e^{-x^2/2}$$

Plugging into (1) yields the desired lower bound.

**Problem 3.4:** The Jacobian for the transformation from polar to rectangular coordinates is given by

$$J(X_1, X_2; R, \Phi) = \begin{pmatrix} \frac{\partial X_1}{\partial R} & \frac{\partial X_1}{\partial \Phi} \\ \frac{\partial X_2}{\partial R} & \frac{\partial X_2}{\partial \Phi} \end{pmatrix} = \begin{pmatrix} \cos\Phi & -R\sin\Phi \\ \sin\Phi & R\cos\Phi \end{pmatrix}$$

with determinant $|J(X_1, X_2; R, \Phi)| = R$. This gives

$$p_{R,\Phi}(r,\phi) = p_{X_1,X_2}(x_1,x_2)|J(x_1,x_2;r,\phi)|\ |_{x_1=r\cos\phi, x_2=r\sin\phi} = \frac{r}{2\pi\sigma^2}\exp\left(-\frac{r^2}{2\sigma^2}\right) I_{\{r\geq 0\}} I_{\{0\leq\phi\leq 2\pi\}}$$

This separates out into the marginals $p_\Phi(\phi) = \frac{1}{2\pi} I_{\{0\leq\theta\leq 2\pi\}}$ and $p_R(r) = \frac{r}{\sigma^2}\exp\left(-\frac{r^2}{2\sigma^2}\right) I_{\{r\geq 0\}}$.

**(c)** Setting $Z = R^2$, we have that, for $z \geq 0$, that

$$P[Z > z] = P[R > \sqrt{z}] = \int_{\sqrt{z}}^{\infty} \frac{r}{\sigma^2}\exp\left(-\frac{r^2}{2\sigma^2}\right)\,dr = \exp\left(-\frac{z}{2\sigma^2}\right) \tag{2}$$

Of course, $P[Z > z] = 1$ for $z < 0$. Equation (2) gives the complementary CDF of an exponential random variable with mean $2\sigma^2$. The density of $Z$ is

$$p_Z(z) = -\frac{dP[Z > z]}{dz} = \frac{1}{2\sigma^2}\exp\left(-\frac{z}{2\sigma^2}\right) I_{\{z\geq 0\}}$$

**(d)** Using (2), the event that $Z$ is 20 dB below its mean $2\sigma^2$ has probability

$$P[Z < 10^{-2}2\sigma^2] = 1 - P[Z < 10^{-2}2\sigma^2] = 1 - e^{-10^{-2}} \approx 10^{-2}$$

which does not depend on $\sigma^2$.

**(e)** As before, transformation to polar coordinates yields

$$\begin{aligned} p_{R,\Phi}(r,\phi) &= \tfrac{r}{2\pi\sigma^2}\exp\left(-\tfrac{(r\cos\phi - m_1)^2}{2\sigma^2} - \tfrac{(r\sin\phi - m_2)^2}{2\sigma^2}\right) I_{\{r\geq 0\}} I_{\{0\leq\theta\leq 2\pi\}} \\ &= \tfrac{r}{2\pi\sigma^2}\exp\left(-\tfrac{r^2+m^2}{2\sigma^2}\right)\exp\left(-\tfrac{rm\cos(\phi-\theta)}{2\sigma^2}\right) I_{\{r\geq 0\}} I_{\{0\leq\theta\leq 2\pi\}} \end{aligned}$$

where $m_1 = m\cos\theta$, $m_2 = m\sin\theta$. Clearly, $R$ and $\Phi$ are no longer independent, and $\Phi$ is more likely to take values around $\theta$. Integrating out $\Phi$, we obtain that

$$p_R(r) = \frac{r}{2\pi\sigma^2}\exp\left(-\frac{r^2+m^2}{2\sigma^2}\right) I_0\left(\frac{rm}{\sigma^2}\right) I_{\{r\geq 0\}} \tag{3}$$

where we have used the fact that

$$I_0(x) = \frac{1}{2\pi}\int \exp\left(x\cos(\phi-\theta)\right)d\phi$$

is independent of the offset $\theta$, since we are integrating over a complete period for the integrand, which is periodic with period $2\pi$.

It is worth noting that the Rician density (3) for $R$ depends only on $m = \sqrt{m_1^2 + m_2^2}$ and $\sigma^2$, rather than on the individual values of $m_1$ and $m_2$.

**Problem 3.5**

**(a)** For $a > 0$ and $X_1$, $X_2$ i.i.d. $N(0,1)$, we have

$$P[|X_1| > a, |X_2| > a] = P[|X_1| > a]P[|X_2| > a] = 4Q^2(a)$$

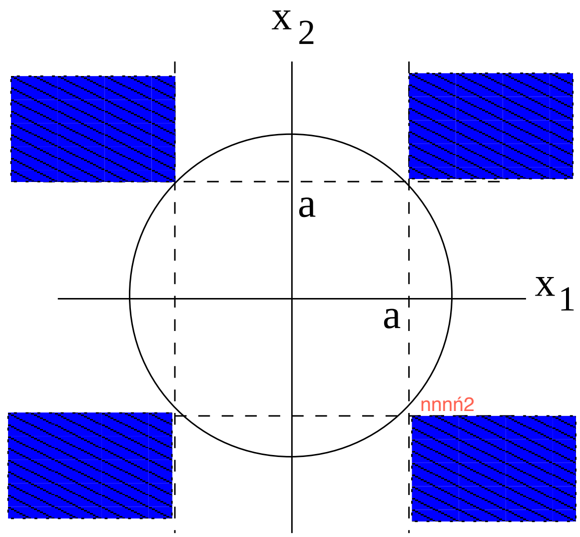



Figure 1: The region corresponding to 3.5(a) is shaded, and is a subset of the region in 3.5(b), which is the outside of the circle.

**(b)** From Problem 3.4, we know that $X_1^2 + X_2^2$ is exponential with mean 2, so

$$P[X_1^2 + X_2^2 > 2a^2] = e^{-2a^2/2} = e^{-a^2}$$

**(c)** See Figure 1.
**(d)** From the figure, we see that the probability in (b) is larger than that in (a) for $a > 0$, so that

$$4Q^2(a) \leq e^{-a^2}$$

which gives the desired bound $Q(a) \leq \frac{1}{2}e^{-a^2/2}$, $a \geq 0$. (from the figure, we see that the inequality is strict for $a > 0$).

**Problem 3.6**
(a) Let $Z = (y * h)(t_0)$, where $h(t) = s(-t)$. Then $Z \sim N(m, v^2)$ if 1 sent, and $Z \sim N(0, v^2)$ if 0 sent, where

$$v^2 = \sigma^2 ||h||^2 = \sigma^2 4 \int_0^1 t^2 dt = \frac{4}{3}\sigma^2$$

$$m = (s * h)(t_0) = \int s(t)s(t - t_0)dt = \int_0^1 t(1-t)dt = \frac{1}{6}$$

Thus, for the ML decision rule

$$P_e = Q\left(\frac{|m|}{2v}\right) = Q\left(\frac{1}{8}\sqrt{\frac{E_b}{N_0}}\right)$$

using the fact that we must have

$$\frac{|m|}{2v} = a\sqrt{\frac{E_b}{N_0}}$$

(why?) where $a$ is a constant determined by substituting $E_b = \frac{1}{2}||s||^2 = \frac{2}{3}$ and $N_0 = 2\sigma^2$.
(b) We can improve the error probability by sampling at $t_0 = 0$. We then have $m = ||s||^2 = \frac{4}{3}$,

while $v^2$ is as before. This gives, reasoning as in (a),

$$P_e = Q\left(\frac{|m|}{2v}\right) = Q\left(\sqrt{\frac{E_b}{N_0}}\right)$$

the usual formula for the performance of optimal reception of on-off keying in AWGN.
(c) For $h(t) = I_{[0,2]}$, we again have the same model for the decision statistic $Z = y * h(t_0)$, but with $v^2 = \sigma^2||h||^2 = 2\sigma^2$, and $m = (s * h)(t_0)$. The performance improves with $|m|$, which is maximized at $t_0 = 2$ ($m = 1$) or $t_0 = 4$ ($m = -1$). We therefore get that, for the ML decision rule,

$$P_e = Q\left(\frac{|m|}{2v}\right) = Q\left(\sqrt{\frac{3E_b}{8N_0}}\right)$$

(d) Note that we can approximate the matched filter $s(-t)$ using linear combinations of two shifted versions of $h(t) = I_{[0,2]}$, by approximating triangles by rectangles. That is, the matched filter shape is approximated as $\tilde{h}(t) = h(t+2) - h(t+4)$. Thus, we can use the decision statistic

$$Z = (y * \tilde{h})(0) = (y * h)(2) - (y * h)(4)$$

We now have $Z \sim N(m, v^2)$ if 1 sent, and $Z \sim N(0, v^2)$ if 0 sent, where

$$v^2 = \sigma^2||\tilde{h}||^2 = 4\sigma^2$$

$$m = (s * \tilde{h})(0) = 2$$

As before, we can enforce the scaling with $\frac{E_b}{N_0}$ to get

$$P_e = Q\left(\frac{|m|}{2v}\right) = Q\left(\sqrt{\frac{3E_b}{4N_0}}\right)$$

3 dB better than the performance in (c), and $10\log_{10}\frac{4}{3} = 1.25$ dB worse than the optimal receiver in (b).

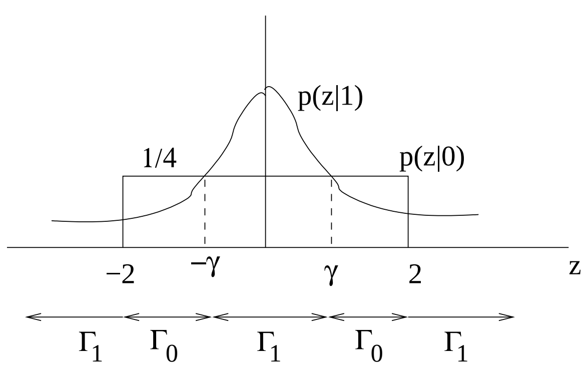



Figure 2: Conditional densities and decision regions for Problem 3.7.

**Problem 3.7:** The conditional densities and decision regions are sketched in Problem 3.7 (not to scale). The threshold $\gamma$ satisfies $p(\gamma|1) = p(\gamma|0)$, or

$$\frac{1}{\sqrt{2\pi}}e^{-\gamma^2/2} = \frac{1}{4}$$

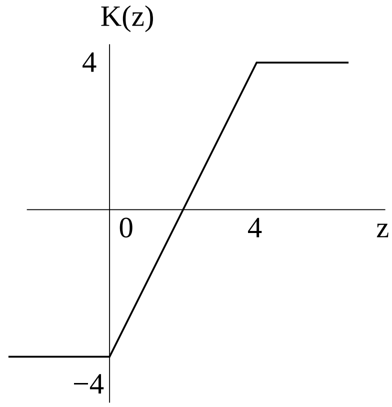



Figure 3: Log likelihood ratio for Problem 3.8.

which yields $\gamma = \sqrt{3\log(2) - \log(\pi)} \approx 0.97$.

**Problem 3.8 (a)** We have $p(z|0) = \frac{1}{2}e^{-|z|}$ and $p(z|1) = \frac{1}{2}e^{-|z-4|}$, so that the log likelihood ratio, sketched in Figure 3, is given by

$$K(z) = \log\frac{p(z|1)}{p(z|0)} = |z| - |z-4| = \begin{cases} 4, & z > 4 \\ 2z-4, & -4 \le z \le 0 \\ -4, & z < 0 \end{cases}$$

**(b)** The conditional error probability given 1 is

$$\begin{aligned} P_{e|1} &= P[Z < 1|H_1] = \int_{-\infty}^{1} p(z|1)dz = \int_{-\infty}^{1} \tfrac{1}{2}e^{-|z-4|}dz \\ &= \tfrac{1}{2}\int_{-\infty}^{1} e^{z-4}dz = e^{-3}/2 = 0.025 \end{aligned}$$

**(c)** The region $z < 1$ can be written as $K(z) < -2$. The MPE rule compares $K(z)$ to $\log\frac{\pi_0}{\pi_1}$. Thus, the rule in (b) is the MPE rule if $\log\frac{\pi_0}{\pi_1} = -2$, which yields $\pi_0 = \frac{1}{e^2+1} \approx 0.12$.

**Problem 3.9 (a)** The likelihood ratio is given by

$$L(y) = \frac{p(y|1)}{p(y|0)} = \frac{e^{-m_1}m_1^y/y!}{e^{-m_0}m_0^y/y!} = \left(\frac{m_1}{m_0}\right)^y e^{-(m_1-m_0)}$$

The ML rule compares the log likelihood ratio to zero. Taking the log above, we have $\log L(y) = y\log\frac{m_1}{m_0} - (m_1 - m_0)$, so that the ML rule reduces to comparing $y$ to a threshold $\gamma = \frac{m_1-m_0}{\log\frac{m_1}{m_0}}$. For $m_1 = 100$, $m_0 = 10$, we have $\gamma \approx 39.1$, so that

$$\delta_{ML}(y) = \begin{cases} 1, & y > 39 \\ 0, & y \le 39 = t \end{cases}$$

where $t = \lfloor\gamma\rfloor$ in general.

**(b)** The conditional error probabilities are given by

$$P_{e|1} = P[Y \le t|H_1] = \sum_{y=0}^{t}\frac{m_1^y}{y!}e^{-m_1}$$

$$P_{e|0} = 1 - P[Y \le t|H_0] = 1 - \sum_{y=0}^{t}\frac{m_0^y}{y!}e^{-m_0}$$

**Problem 3.10 (a)** We have

$$p(y|1) = \frac{e^{-(y-1)^2/8}}{\sqrt{8\pi}}$$
$$p(y|0) = \frac{e^{-(y+1)^2/2}}{\sqrt{2\pi}}$$

The optimal rule consists of comparing the log likelihood ratio to a threshold. The log likelihood ratio can be written as

$$\log L(y) = \log p(y|1) - \log p(y|0) = -(y-1)^2/8 + (y+1)^2/2 - \log 2$$

which has the desired quadratic form.
**(b)** For $\pi_0 = 1/3$, we compare $\log L(y)$ to the threshold $\log \frac{\pi_0}{\pi_1} = -\log 2$. Simplifying, we obtain the MPE rule

$$3y^2 + 10y \underset{H_0}{\overset{H_1}{\gtrless}} 8\log 2 - 3$$

**Problem 3.11**
**(a)** Signal space representations with respect to the given orthonormal basis are:
**Signal Set A:** $\mathbf{s}_1 = (1,0,0,0)^T$, $\mathbf{s}_2 = (0,1,0,0)^T$, $\mathbf{s}_3 = (0,0,0,1)^T$ and $\mathbf{s}_4 = (0,0,0,1)^T$
**Signal Set B:** $\mathbf{s}_1 = (1,0,0,1)^T$, $\mathbf{s}_2 = (0,1,1,0)^T$, $\mathbf{s}_3 = (1,0,1,0)^T$ and $\mathbf{s}_4 = (0,1,0,1)^T$
(b) For Signal Set A, the pairwise distance between any two points satisfies $d^2 = d^2_{min} = 2$, while the energy per symbol is $E_s = 1$. Thus, $E_b = E_s/(\log_2 4) = 1/2$, and $d^2_{min}/E_b = 4$. The union bound on symbol error probability is therefore given by

$$P_e(\text{signal set A}) \le 3Q\left(\sqrt{d^2_{min}/E_b}\sqrt{E_b/2N_0}\right) = 3Q\left(\sqrt{2E_b/N_0}\right)$$

For signal set B, each signal has one neighbor at distance given by $d_1^2 = 4$ and two at distance given by $d_2^2 = d^2_{min} = 2$. The energy per symbol is $E_s = 2$, so that $E_b = 1$. The union bound is given by

$$\begin{aligned} P_e(\text{signal set B}) &\le 2Q\left(\sqrt{d_2^2/E_b}\sqrt{E_b/2N_0}\right) + Q\left(\sqrt{d_1^2/E_b}\sqrt{E_b/2N_0}\right) \\ &= 2Q\left(\sqrt{E_b/N_0}\right) + Q\left(\sqrt{2E_b/N_0}\right) \end{aligned}$$

(c) For exact analysis of error probability for Signal Set B, suppose that the received signal in signal space is given by $\mathbf{Y} = (Y_1, Y_2, Y_3, Y_4)$. Condition on the first signal $\mathbf{s}_1 = (1,0,0,1)^T$ being sent. Then

$$Y_1 = 1 + N_1,\ Y_2 = N_2,\ Y_3 = N_3,\ Y_4 = 1 + N_4$$

where $N_1, ..., N_4$ are i.i.d. $N(0, \sigma^2)$ random variables. A correct decision is made if $\langle \mathbf{Y}, \mathbf{s}_1 \rangle > \langle \mathbf{Y}, \mathbf{s}_k \rangle$, $k = 2, 3, 4$. These inequalities can be written out as

$$\begin{aligned} \langle \mathbf{Y}, \mathbf{s}_1 \rangle = Y_1 + Y_4 > \langle \mathbf{Y}, \mathbf{s}_2 \rangle = Y_2 + Y_3 \\ \langle \mathbf{Y}, \mathbf{s}_1 \rangle = Y_1 + Y_4 > \langle \mathbf{Y}, \mathbf{s}_3 \rangle = Y_1 + Y_3 \\ \langle \mathbf{Y}, \mathbf{s}_1 \rangle = Y_1 + Y_4 > \langle \mathbf{Y}, \mathbf{s}_4 \rangle = Y_2 + Y_4 \end{aligned}$$

The second and third inequalities give $Y_1 > Y_2$ and $Y_4 > Y_3$, and imply the first inequality. Thus, the conditional probability of correct reception is given by

$$P_{c|1} = P[Y_1 > Y_2 \textit{ and } Y_4 > Y_3|1] = P[Y_1 > Y_2|1]P[Y_4 > Y_3|1]$$

since $Y_k$ are conditionally independent given the transmitted signal. Noting that $Y_1 - Y_2$ and $Y_4 - Y_3$ are independent $N(1, 2\sigma^2)$, we have

$$P_{e|1} = 1 - Pc|1 = 1 - \left(1 - Q\left(\frac{1}{\sqrt{2\sigma^2}}\right)\right)^2 = 2Q\left(\frac{1}{\sqrt{2\sigma^2}}\right) - Q^2\left(\frac{1}{\sqrt{2\sigma^2}}\right)$$

Setting $\frac{1}{\sqrt{2\sigma^2}} = a\sqrt{E_b/N_0}$, with $E_b = 1$ and $N_0 = 2\sigma^2$, we have $a = 1$. Further, by symmetry, we have $P_e = P_{e|1}$. We therefore obtain that

$$P_e = 2Q\left(\sqrt{E_b/N_0}\right) - Q^2\left(\sqrt{E_b/N_0}\right), \quad \textbf{exact error probability for signal set B}$$

**Problem 3.12**
**(a)** For 8-PSK, the symbol energy $E_s = R^2$. For the QAM constellations,

$$\begin{aligned} E_s(QAM1) &= \mathrm{I-channel\ avg\ energy} \ + \ \mathrm{Q-channel\ avg\ energy} \\ &= \tfrac{4}{8}[(d_1/2)^2 + (3d_1/2)^2] + (d_1/2)^2 = \tfrac{3}{2}d_1^2 \end{aligned}$$

$$\begin{aligned} E_s(QAM2) &= \mathrm{I-channel\ avg\ energy} \ + \ \mathrm{Q-channel\ avg\ energy} \\ &= [\tfrac{6}{8}(d_2/2)^2 + \tfrac{2}{8}(3d_2/2)^2] + [\tfrac{4}{8}d_2^2 + \tfrac{4}{8}0] = \tfrac{5}{4}d_2^2 \end{aligned}$$

where $d_1 = d_{min}^{(1)}$ and $d_2 = d_{min}^{(2)}$. Since the number of constellation points is the same for each constellation, equal energy per bit corresponds to equal symbol energy, and occurs when

$$R = \sqrt{3/2}\ d_1, \qquad d_2 = \sqrt{6/5}\ d_1$$

**(b)** From (a), for the same $E_b$,

$$d_{min}^{8PSK} = 2R\sin\frac{\pi}{8} = .937d_1 < d_1 < d_2$$

so that, in the high SNR regime, we expect that

$$P_e(8PSK) > P_e(QAM1) > P_e(QAM2)$$

(c) Each symbol is assigned 3 bits. Since 8-PSK and QAM1 are regular constellations with at most 3 nearest neighbors per point, we expect to be able to Gray code. However, QAM2 has some points with 4 nearest neighbors, so we definitely cannot Gray code it. We can, however, try to minimize the number of bit changes between neighbors. Figure 4 shows Gray codes for 8-PSK and QAM1. The labeling for QAM2 is arbitrarily chosen to be such that points with 3 or fewer nearest neighbors are Gray coded.

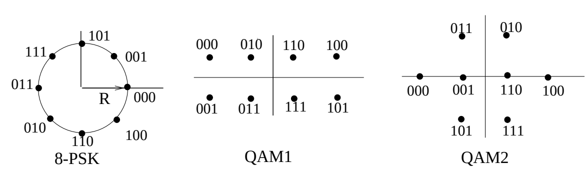



Figure 4: Bit mappings for Problem 3.12.

(d) For Gray coded 8-PSK and QAM1, a symbol error due to decoding to a nearest neighbor causes only 1 out of the 3 bits to be in error. Hence, using the nearest neighbors approximation, $P[bit\ error] \approx \frac{1}{3}P[symbol\ error]$. On the other hand, $P[symbol\ error] \approx \bar{N}_{d_{min}} Q(\frac{d_{min}}{2\sigma})$, where

$\bar{N}_{d_{min}}$ is the average number of nearest neighbors. While the latter is actually an upper bound on the symbol error probability (the nearest neighbors approximation coincides with the intelligent union bound in these cases), the corresponding expression for the bit error probability need not be an upper bound (why?).
For 8-PSK, $d_{min} = 2R\sin\frac{\pi}{8}$ and $E_s = 3E_b = R^2$. Plugging in $\sigma^2 = N_0/2$ and $\bar{N}_{d_{min}} = 2$, we obtain

$$P[bit\ error]_{8PSK} \approx \frac{2}{3}Q\left(\sqrt{\frac{3(1-\frac{1}{\sqrt{2}})E_b}{N_0}}\right)$$

For QAM1, $\bar{N}_{d_{min}} = 5/2$ and $E_s = 3E_b = 3d_1^2/2$, so that

$$P[bit\ error]_{QAM1} \approx \frac{5}{6}Q\left(\sqrt{\frac{E_b}{N_0}}\right)$$

For QAM2, we need to make nearest neighbors approximation specifically for bit error probability. Let $N(\mathbf{b})$ total number of bit changes due to decoding into nearest neighbors when symbol $\mathbf{b}$ is sent. For the labeling given, these are specified by Table 1. Let $\bar{N}_{bit} = \frac{1}{8}\sum_{\mathbf{b}} N(\mathbf{b}) = 11/4$ denote

| $\mathbf{b}$ | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |
|---|---|---|---|---|---|---|---|---|
| $N(\mathbf{b})$ | 1 | 6 | 2 | 2 | 1 | 2 | 6 | 2 |

Table 1: Number of bit changes due to decoding into nearest neighbors for each symbol of QAM2

the average number of bits wrong due to decoding into nearest neighbors. Since each signal point is labeled by 3 bits, the nearest neighbors approximation for the bit error probability is now given by

$$P[bit\ error]_{QAM2} \approx \frac{1}{3}\bar{N}_{bit}Q\left(\frac{d_{min}}{2\sigma}\right) = \frac{11}{12}Q\left(\sqrt{\frac{6E_b}{5N_0}}\right)$$

(We can reduce the factor from 11/12 to 5/6 using an alternative labeling.)

While we have been careful about the factors multiplying the $Q$ function, these are insignificant at high SNR compared to the argument of the $Q$ function, and are often ignored in practice. For $E_b/N_0 = 15dB$, the preceding approximations give the values shown in Table 2. The ordering of error probabilities is as predicted in (b).

| Constellation | 8PSK | QAM1 | QAM2 |
|---|---|---|---|
| P[bit error] | $4.5 \times 10^{-8}$ | $7.8 \times 10^{-9}$ | $3.3 \times 10^{-10}$ |

Table 2: Bit error probabilities at $E_b/N_0$ of 15 dB

**Problem 3.13:**
**(a)** For $R/r = \sqrt{2}$. the constellation takes the rectangular shape shown in Figure 5.
**(b)** We compute the intelligent union bound on the error probability conditioned on two typical signal points shown in Figure 5.

$$P(e|s1) \leq 2Q\left(\frac{d}{2\sigma}\right) = Q\left(\sqrt{\frac{d^2}{E_b}}\sqrt{\frac{E_b}{2N_0}}\right)$$

We now compute $\eta = \frac{d^2}{E_b}$ to express the result in terms of $E_b/N_0$. The energy per symbol is

$$E_s = 2 \times \mathrm{I} - \text{channel energy} = 2\left(\frac{6}{8} \times d^2 + \frac{2}{8} \times 0\right) = 3d^2/2$$

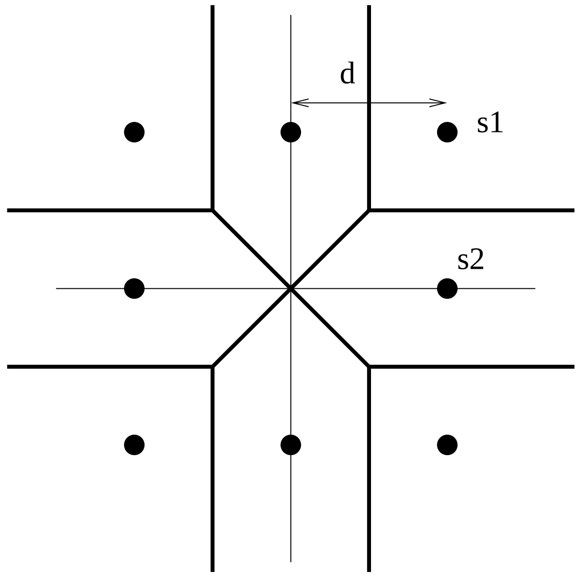



Figure 5: ML decision regions for Problem 3.13(a).

Since $E_b = E_s / \log_2 8$, we have $\eta = \frac{d^2}{E_b} = 2$, which yields

$$P(e|s1) \leq 2Q\left(\sqrt{E_b/N_0}\right)$$

Similarly,

$$P(e|s2) \leq 2Q\left(\frac{d}{2\sigma}\right) + 2Q\left(\frac{\sqrt{2}d}{2\sigma}\right) = 2Q\left(\sqrt{E_b/N_0}\right) + 2Q\left(\sqrt{2E_b/N_0}\right)$$

The average error probability is given by

$$P_e = \frac{1}{2}\left(P(e|s1) + P(e|s2)\right) \leq 2Q\left(\sqrt{E_b/N_0}\right) + Q\left(\sqrt{2E_b/N_0}\right)$$

**(c)** We wish to design the parameter $x = R/r \geq 1$ to optimize the power efficiency, which is given by

$$\eta = \min(d_1^2, d_2^2)/E_b$$

where $d_1$ and $d_2$ shown in Figure 6 are given by

$$d_1^2 = 2r^2$$

$$d_2^2 = \left(R/\sqrt{2}\right)^2 + \left(R/\sqrt{2} - r\right)^2 = R^2 + r^2 - \sqrt{2}Rr = r^2(1 + x^2 - \sqrt{2}x)$$

The energy per symbol is $E_s = (r^2 + R^2)/2$, so that

$$E_b = E_s / \log_2 8 = (r^2 + R^2)/6 = r^2(1 + x^2)/6$$

It is easy to check the following: $\eta_1 = d_1^2/E_b$ decreases with $x$, and $\eta_2 = d_2^2/E_b$ increases with $x$ (for $x \geq 1$). Furthermore, at $x = 1$, $\eta_1 > \eta_2$. This shows that the optimal $x \geq 1$ corresponds to $\eta_1 = \eta_2$, i.e., $d_1^2 = d_2^2$. That is,

$$2r^2 = r^2(1 + x^2 - \sqrt{2}x)$$

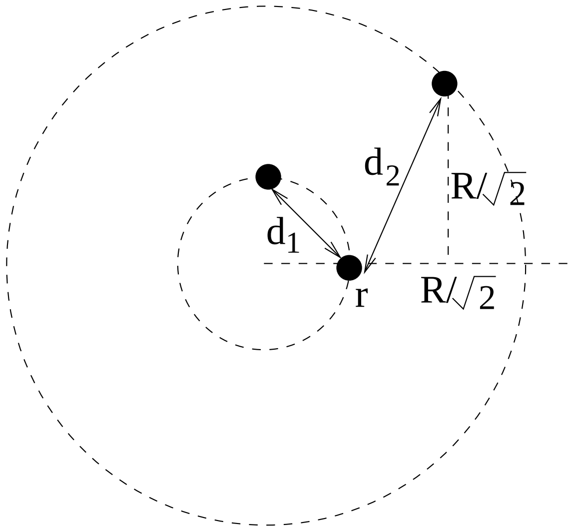



Figure 6: Signal space geometry for Problem 3.13(c).

or

$$x^2 - \sqrt{2}x - 1 = 0$$

The solution to this in the valid range $x \geq 1$ is given by

$$x = R/r = \frac{\sqrt{2} + \sqrt{6}}{2} \approx 1.93$$

The corresponding power efficiency is given by

$$\eta = 2(3 - \sqrt{3}) \approx 2.54$$

which is about 1 dB better than the power efficiency of $\eta = 2$ for $x = \sqrt{2} \approx 1.41$ in part (a).

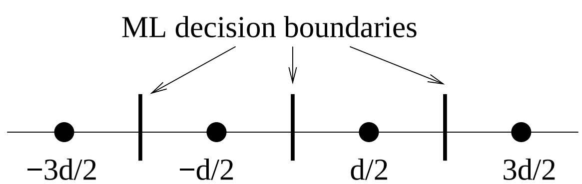



Figure 7: 4PAM constellation and decision boundaries for Problem 3.14.

**Problem 3.14:** The 4PAM constellation is shown in Figure 7. Let $N \sim N(0, \sigma^2)$ denote the noise sample corrupting the received signal. Conditioned on an outer signal point being sent (consider the leftmost point without loss of generality), the error probability is

$$P(e|outer) = P[N > d/2] = Q\left(\frac{d}{2\sigma}\right)$$

Conditioned on one of the two inner points being sent, the error probability is

$$P(e|inner) = P[|N| > d/2] = 2Q\left(\frac{d}{2\sigma}\right)$$

Note that $E_s = \frac{1}{2}\left((d/2)^2 + (3d/2)^2\right)$ and $E_b = E_s/\log_2 4$, so that $d^2/E_b = 8/5$. We therefore can write

$$P_e = \frac{1}{2}\left(P(e|outer) + P(e|inner)\right) = \frac{3}{2}Q\left(\frac{d}{2\sigma}\right) = \frac{3}{2}Q\left(\sqrt{\frac{4E_b}{5N_0}}\right)$$

16QAM can be viewed as a product of 2 4PAM constellations sent in parallel over independent WGN channels, so that a symbol error occurs if either of the 2 4PAM symbols are received incorrectly. Thus,

$$P_e(16QAM) = 1 - (1 - P_e(4PAM))^2 = 2P_e(4PAM) - P_e^2(4PAM)$$

Using the 4PAM error probability bound yields the following bound on the error probability of 16QAM:

$$P_e(16QAM) = 3Q\left(\sqrt{\frac{4E_b}{5N_0}}\right) - \frac{9}{4}Q^2\left(\sqrt{\frac{4E_b}{5N_0}}\right)$$

**Problem 3.15:** The solution steps are laid out in detail in the problem.

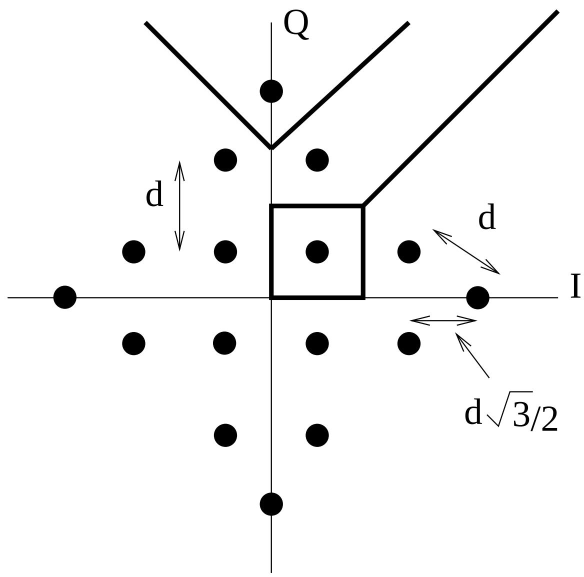



Figure 8: 16-cross constellation for Problem 3.16.

**Problem 3.16: (a)** The ML decision regions for three typical points are shown in Figure 8.
**(b)** As shown in Figure 8, the distances are as in rectangular 16QAM, except for the corner point. The symbol energy

$$\begin{aligned} E_s &= 2 \times \text{I channel energy} = 2\left[\tfrac{8}{16}(d/2)^2 + \tfrac{4}{16}(3d/2)^2 + \tfrac{2}{16}\left((3+\sqrt{3})d/2\right)^2\right] \\ &= d^2\tfrac{17+3\sqrt{3}}{8} = 2.77d^2 \end{aligned}$$

In contrast, rectangular 16QAM has symbol energy

$$E_s = 2 \times \text{I} - \text{channel energy} = 2\left[\frac{8}{16}(d/2)^2 + \frac{8}{16}(3d/2)^2\right] = 2.5d^2$$

Since the minimum distance $d$ is the same, rectangular 16QAM is more power efficient. Cross constellations are typically used when the number of constellation points is an odd power of 2 (e.g. 8QAM, 32QAM), when symmetric rectangular constellations are not possible.

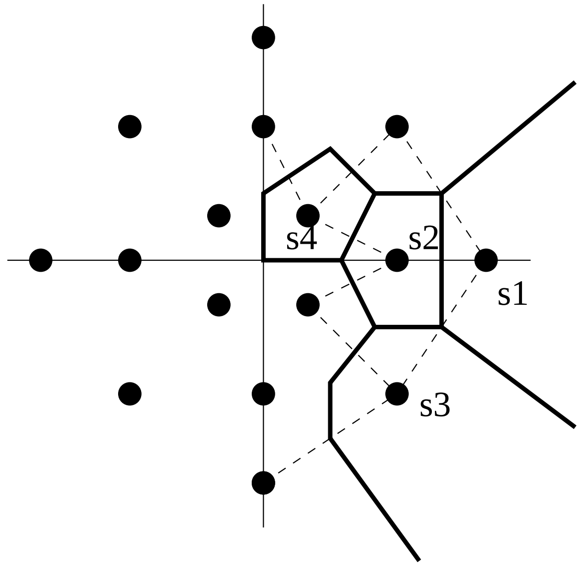



Figure 9: ML decision regions for 4 typical points for the constellation in Problem 3.17.

**Problem 3.17:** **(a)** The ML regions are shown in Figure 9.
**(b)** We find intelligent union bounds for the error probabilities conditioned on the 4 typical points shown in Figure 9. Once this is done, the overall error probability is given by averaging as follows:

$$P_e = \frac{4}{16}\left(P(e|s1) + P(e|s2) + P(e|s3) + P(e|s4)\right)$$

For point s1, the distances to the neighbors determining its ML decision region are 2, $\sqrt{13}$, and $\sqrt{13}$. For point s2, the distances are 2, 3, 3, $\sqrt{5}$, $\sqrt{5}$. For s3, the distances are 3,3,$\sqrt{13}$, $\sqrt{13}$ and $\sqrt{8}$. For s4, the distances are 2, 2, $\sqrt{5}$, $\sqrt{5}$, $\sqrt{8}$. Each neighbor at distance $d$ leads to a term of the form

$$Q\left(\frac{d}{2\sigma}\right) = Q\left(\frac{d}{d_{min}}\sqrt{\frac{d_{min}^2}{E_b}}\sqrt{\frac{E_b}{2N_0}}\right)$$

where $d_{min} = 2$. The symbol energy is given by

$$E_s = 2 \times \text{I channel energy} = 2\left[\frac{2}{16}5^2 + \frac{6}{16}3^2 + \frac{4}{16}1^2\right] = 27/2$$

so that $E_b = E_s/\log_2 16 = 27/8$ and the power efficiency is

$$\frac{d_{min}^2}{E_b} = 32/27$$

We can now write

$$P(e|s1) \leq Q\left(\sqrt{\frac{16E_b}{27N_0}}\right) + 2Q\left(\sqrt{13/4}\sqrt{\frac{16E_b}{27N_0}}\right)$$

$$P(e|s2) \leq Q\left(\sqrt{\frac{16E_b}{27N_0}}\right) + 2Q\left(\frac{3}{2}\sqrt{\frac{16E_b}{27N_0}}\right) + 2Q\left(\sqrt{5/4}\sqrt{\frac{16E_b}{27N_0}}\right)$$

$$P(e|s3) \leq 2Q\left(\frac{3}{2}\sqrt{\frac{16E_b}{27N_0}}\right) + 2Q\left(\sqrt{13/4}\sqrt{\frac{16E_b}{27N_0}}\right) + Q\left(\sqrt{2}\sqrt{\frac{16E_b}{27N_0}}\right)$$

$$P(e|s2) \leq 2Q\left(\sqrt{\frac{16E_b}{27N_0}}\right) + 2Q\left(\sqrt{5/4}\sqrt{\frac{16E_b}{27N_0}}\right) + Q\left(\sqrt{2}\sqrt{\frac{16E_b}{27N_0}}\right)$$

Averaging (and simplifying the arguments of the $Q$ functions), we get

$$P_e \leq Q\left(\sqrt{\frac{16E_b}{27N_0}}\right) + \frac{1}{2}Q\left(\sqrt{\frac{32E_b}{27N_0}}\right) + Q\left(\sqrt{\frac{20E_b}{27N_0}}\right) + Q\left(\sqrt{\frac{4E_b}{3N_0}}\right) + Q\left(\sqrt{\frac{52E_b}{27N_0}}\right)$$

**(c)** The nearest neighbors approximation is simply the first term of the intelligent union bound above:

$$P_e \approx Q\left(\sqrt{\frac{16E_b}{27N_0}}\right)$$

**Problem 3.18:**
**(a)** Letting $N_Q$, $N_I$ denote the iid $N(0, \sigma^2)$ noise components in the $I$ and $Q$ directions. As in the standard QPSK analysis, we may use symmetry to condition on a particular symbol being sent to compute the error and erasure probabilities.

$$p = P[error] = P[error|-1-j\ sent] \leq P[N_I > (1+\alpha)d/2 \text{ or } N_Q > (1+\alpha)d/2] \leq 2P[N_I > (1+\alpha)d/2]$$

so that

$$p \leq 2Q\left((1+\alpha)\sqrt{\frac{2E_b}{N_0}}\right) \tag{4}$$

Similarly,

$$\begin{aligned} q = P[erasure] &= P[erasure| - 1 - jsent] \\ &= P[(1-\alpha)d/2 < N_I < (1+\alpha)d/2 \text{ or } (1-\alpha)d/2 < N_Q < (1+\alpha)d/2] \\ &\leq 2P[(1-\alpha)d/2 < N_I < (1+\alpha)d/2] \end{aligned}$$

so that

$$q \leq 2\left\{Q\left((1-\alpha)\sqrt{\frac{2E_b}{N_0}}\right) - Q\left((1+\alpha)\sqrt{\frac{2E_b}{N_0}}\right)\right\} \tag{5}$$

**(b)** For the exact probabilities, conditioning on $-1-j$ as before, we find

$$\begin{aligned} p &= P[N_I > (1+\alpha)d/2, N_Q < (1-\alpha)d/2] \ + \ P[N_I < (1-\alpha)d/2, N_Q > (1+\alpha)d/2] \\ &\quad + \ P[N_I > (1+\alpha)d/2, N_Q > (1+\alpha)d/2] \\ &= 2Q\left((1+\alpha)\sqrt{\tfrac{2E_b}{N_0}}\right)\left[1 - Q\left((1-\alpha)\sqrt{\tfrac{2E_b}{N_0}}\right)\right] + Q^2\left((1+\alpha)\sqrt{\tfrac{2E_b}{N_0}}\right) \end{aligned}$$

Similarly,

$$\begin{aligned} q &= P[\{(1-\alpha)d/2 < N_I < (1+\alpha)d/2\} \text{ or } \{(1-\alpha)d/2 < N_Q < (1+\alpha)d/2\}] \\ &= 2P[\{(1-\alpha)d/2 < N_I < (1+\alpha)d/2\}] - P^2[\{(1-\alpha)d/2 < N_I < (1+\alpha)d/2\}] \end{aligned}$$

That is,

$$q = 2q_1 - q_1^2$$

where

$$q_1 = Q\left((1-\alpha)\sqrt{\frac{2E_b}{N_0}}\right) - Q\left((1+\alpha)\sqrt{\frac{2E_b}{N_0}}\right)$$

**(c)** Setting $q = 2p$ using the bounds (4)-(5), we obtain

$$Q\left((1-\alpha)\sqrt{\frac{2E_b}{N_0}}\right) = 3Q\left((1+\alpha)\sqrt{\frac{2E_b}{N_0}}\right)$$

Solving numerically for $E_b/N_0 = 10^{4/10}$ (i.e., 4 dB), we obtain $\alpha = .095$.

**Problem 3.19:**

**(a)** Letting $N_I$, $N_Q$ denote the I and Q noise components, it follows that

$$P_{e1} = P[N_I > \sqrt{E_s}\sin\theta] = Q(\sqrt{\frac{2E_s}{N_0}}\sin\theta), \quad P_{e2} = P[N_Q > \sqrt{E_s}\cos\theta] = Q(\sqrt{\frac{2E_s}{N_0}}\cos\theta)$$

**(b)** For $0 \leq \theta \leq \frac{\pi}{4}$, $\cos\theta \geq \sin\theta$, so that $P_{e1} \geq P_{e2}$. We therefore require that $P_{e2}(R2) = P_{e1}(R1) = 10^{-3}$. Letting $E_{si}$ denote the symbol energy received by receiver $Ri$ ($i = 1, 2$), we must have

$$\sqrt{\frac{2E_{s2}}{N_0}}\cos\theta = \sqrt{\frac{2E_{s1}}{N_0}}\sin\theta$$

so that

$$\tan\theta = \sqrt{\frac{E_{s2}}{E_{s1}}} = \sqrt{\left(\frac{1}{2}\right)^4} = \frac{1}{4}$$

i.e., $\theta = \tan^{-1}\frac{1}{4} \approx 14$ degrees.

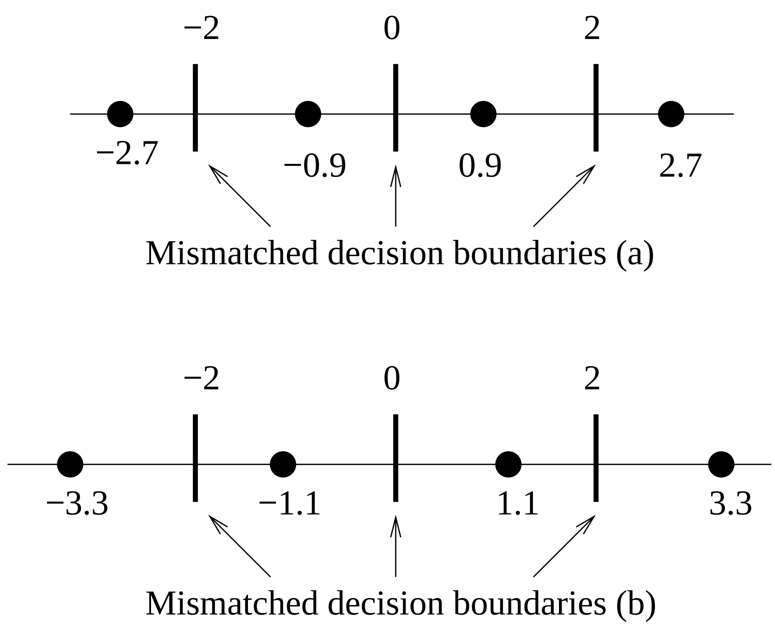



Figure 10: Signal points and decision regions for Problem 3.20

**Problem 3.20**

**(a)** From Figure 10, we can evaluate the error probabilities conditioned on the inner and outer points as follows:

$$P(e|inner) = Q(0.9/\sigma) + Q(1.1/\sigma)$$

$$P(e|outer) = Q(0.7/\sigma)$$

Each of the arguments must take the form $a\sqrt{E_b/N_0}$. For the scaling shown in the figure, $E_s = (0.9^2 + 2.7^2)/2$ and $E_b = E_s/\log_2 4 = 0.9^2(5/2)$. This yields that

$$P(e|inner) = Q\left(\sqrt{4E_b/5N_0}\right) + Q\left(\frac{11}{9}\sqrt{4E_b/5N_0}\right)$$

$$P(e|outer) = Q\left(\frac{7}{9}\sqrt{4E_b/5N_0}\right)$$

Averaging, we get

$$P_e = \frac{1}{2}Q\left(\frac{7}{9}\sqrt{4E_b/5N_0}\right) + \frac{1}{2}Q\left(\sqrt{4E_b/5N_0}\right) + \frac{1}{2}Q\left(\frac{11}{9}\sqrt{4E_b/5N_0}\right)$$

At high SNR, we have a degradation of $20\log_{10}(7/9) = -2.2$ dB due to the mismatch.
**(b)** Proceeding exactly as before, we have

$$P(e|inner) = Q(1.1/\sigma) + Q(0.9/\sigma)$$

$$P(e|outer) = Q(1.3/\sigma)$$

and $E_b = 1.1^2(5/2)$. This yields

$$P(e|inner) = Q\left(\sqrt{4E_b/5N_0}\right) + Q\left(\frac{9}{11}\sqrt{4E_b/5N_0}\right)$$

$$P(e|outer) = Q\left(\frac{13}{11}\sqrt{4E_b/5N_0}\right)$$

with

$$P_e = \frac{1}{2}Q\left(\frac{9}{11}\sqrt{4E_b/5N_0}\right) + \frac{1}{2}Q\left(\sqrt{4E_b/5N_0}\right) + \frac{1}{2}Q\left(\frac{13}{11}\sqrt{4E_b/5N_0}\right)$$

At high SNR, we have a degradation of $20\log_{10}(9/11) = -1.74$ dB due to the mismatch.
**(c)** The AGC circuit is overestimating the received signal power, and therefore scaling the signal points down excessively.

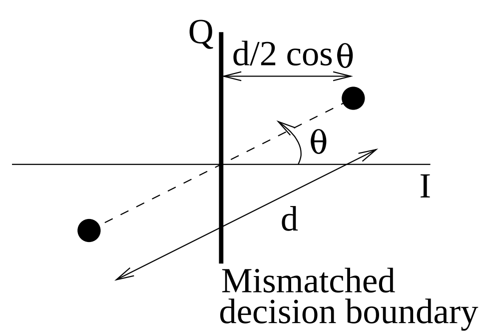



Figure 11: BPSK with phase mismatch (Problem 3.21)

**Problem 3.21**
**(a)** From Figure 11, we see that error occurs if the I-channel noise $N \sim N(0, \sigma^2)$ causes a boundary crossing:

$$P_e = P[N > d/2\ \cos\theta] = Q\left(\frac{d\cos\theta}{2\sigma}\right)$$

Since $E_b = d^2$, we obtain in usual fashion that

$$P_e = Q\left(\sqrt{\frac{2E_b}{N_0}}\ \cos\theta\right)$$

**(b)** The average error probability with mismatch is given by

$$\bar{P}_e = \frac{2}{\pi}\int_{-\pi/4}^{\pi/4} Q\left(\sqrt{\frac{2E_b}{N_0}}\ \cos\theta\right)\ d\theta$$

(numerical results omitted.)

**Problem 3.22**

**(a)** For $b = 0, 1$, the received signal is given by

$$y = (-1)^b A + N(0, \sigma^2)$$

Thus,

$$p(y|0) = \frac{1}{2\pi\sigma^2} e^{-(y-A)^2/2\sigma^2}, \qquad p(y|1) = \frac{1}{2\pi\sigma^2} e^{-(y-A)^2/2\sigma^2}$$

The LLR is given by

$$LLR(b) = \log\frac{P(b=0|y)}{P(b=1|y)} = \log\frac{p(y|0)P[b=0]}{p(y|1)P[b=1]} = \log\frac{P[b=0]}{P[b=1]} + \log\frac{p(y|0)}{p(y|1)}$$

For equiprobable bits, the first term on the extreme right-hand side is zero. Upon simplifying, we get

$$LLR(b) = \log\frac{p(y|0)}{p(y|1)} = \frac{2Ay}{\sigma^2}$$

Since $y$ is conditionally Gaussian, conditioned on the bit sent, so is the LLR. The conditional mean and variance are easily seen to be $(-1)^b\frac{2A^2}{2\sigma^2}$ and $\frac{4A^2}{\sigma^2}$, respectively. These depend only on the SNR $\frac{A^2}{\sigma^2}$. (For uncoded transmission, $E_b = A^2$ and $\frac{A^2}{\sigma^2} = \frac{2E_b}{N_0}$.)

**(b)** For uncoded transmission,

$$P_e = Q\left(\frac{A}{\sigma}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

Numerical search shows that $Q(1.2815) \approx 0.1$, so that an error probability of 10% corresponds to $\frac{A}{\sigma} = 1.2815$. From (a), this implies that the conditional distribution of the LLR, conditioned on 0 sent, is $N(3.285, 6.57)$.

**Problem 3.23 (a)** See Figure 12.

**(b)** We have $E_s = (1^2 + 3^3)/2 = 5$ and $E_b = E_s/\log_2 4 = 5/2$ for the given scaling. We therefore obtain $\sigma^2 = N_0/2 = \frac{5/4}{E_b/N_0}$. For $E_b/N_0 = 4$ (i.e., 6 dB), we obtain $\sigma^2 = 5/16$. The posterior probability of a symbol $s$ is

$$P[s|y] = \frac{p(y|s)P[s \text{ sent}]}{p(y)} = \frac{p(y|s)}{p(y|-3) + p(y|-1) + p(y|+1) + p(y|+3)} \tag{6}$$

for equiprobable transmission. Now,

$$p(y|s) = \frac{1}{2\sigma^2} e^{-(y-s)^2/2\sigma^2} = \alpha e^{\frac{ys - s^2/2}{\sigma^2}}$$

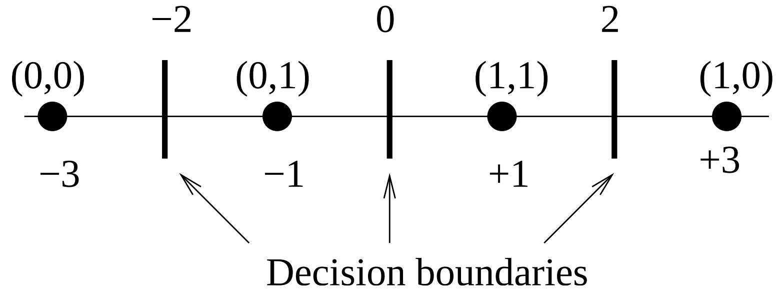



Figure 12: 4PAM bitmaps for Problem 3.23

where the factor $\alpha$ does not depend on $s$, and cancels out in (6). We therefore obtain

$$P[-3|y] = \frac{e^{(-3y-4.5)/\sigma^2}}{e^{(-3y-4.5)/\sigma^2} + e^{(-y-0.5)/\sigma^2} + e^{(y-0.5)/\sigma^2} + e^{(3y-4.5)/\sigma^2}}$$

**(c)** We have $P[b_1 = 1|y] = P[s = +1|y] + P[s = +3|y]$ and $P[b_2 = 1|y] = P[s = -1|y] + P[s = +1|y]$, so that

$$P[b_1 = 1|y] = \frac{e^{(y-0.5)/\sigma^2} + e^{(3y-4.5)/\sigma^2}}{e^{(-3y-4.5)/\sigma^2} + e^{(-y-0.5)/\sigma^2} + e^{(y-0.5)/\sigma^2} + e^{(3y-4.5)/\sigma^2}}$$

$$P[b_2 = 1|y] = \frac{e^{(-y-0.5)/\sigma^2} + e^{(y-0.5)/\sigma^2}}{e^{(-3y-4.5)/\sigma^2} + e^{(-y-0.5)/\sigma^2} + e^{(y-0.5)/\sigma^2} + e^{(3y-4.5)/\sigma^2}}$$

**(d)** The LLRs are given by

$$LLR(b_1) = \log \frac{e^{(-3y-4.5)/\sigma^2} + e^{(-y-0.5)/\sigma^2}}{e^{(y-0.5)/\sigma^2} + e^{(3y-4.5)/\sigma^2}} = -\frac{2y}{\sigma^2} + \log \frac{\cosh\left((y+2)/\sigma^2\right)}{\cosh\left((y-2)/\sigma^2\right)}$$

$$LLR(b_2) = \log \frac{e^{(-3y-4.5)/\sigma^2} + e^{(3y-4.5)/\sigma^2}}{e^{(-y-0.5)/\sigma^2} + e^{(y-0.5)/\sigma^2}} = -\frac{4}{\sigma^2} + \log \frac{\cosh(4y/\sigma^2)}{\cosh(y/\sigma^2)}$$

Note that $LLR(b_1)$ is antisymmetric in $y$, which $LLR(b_2)$ is symmetric, depending on $|y|$ alone.

**Problem 3.24:**

**(a)** From Problem 3.12($d$), $BER \approx \frac{2}{3}Q(0.937\sqrt{\frac{E_b}{N_o}})$. Setting $\frac{E_b}{N_o} = 10^{0.8}$, we get $BER = 0.0062$

**(b)** Hard decision depends only on the angle, not on the amplitude. So the decision is 111.

**(c)** LLR for $b_1$ :

$$\begin{aligned} LLR &= \log \frac{P(b_1 = 0|y)}{P(b_1 = 1|y)} \\ &= \log \frac{P(y|b_1 = 0)}{P(y|b_1 = 1)} \\ &= \log \frac{P(y|000) + P(y|001) + P(y|010) + P(y|011)}{P(y|100) + P(y|101) + P(y|110) + P(y|111)} \end{aligned}$$

Use now $P(y|s_n) = \frac{1}{2\pi}\exp(\frac{-|y-s_n|^2)}{2\sigma^2}$. For our problem, $\sigma^2 = 0.1$, and $s_n = R\exp(j2\pi(n-1)/8)$ (so that 000 is the bit map for $s_1$, 001 for $s_2$, 011 for $s_3$ and so on) where $R = \sqrt{3E_b} = \sqrt{3 * 10^{0.8} * N_0} = \sqrt{3 * 10^{0.8} * 0.2} = 1.95$.

Similarly, find the LLRs for other two bits.

The required LLRs come out to be $-20.7528$ , $-1.6921$ , $-7.9003$

**(d)** Simulation results omitted.

**Problem 3.25** Conditioned on $s_1$ sent, $y = s_1 + n$, and the decision statistics

$$Z_i = \langle s_1 + n, s_i \rangle = E_s \delta_{1i} + N_i$$

where $\{N_i = \langle n, s_i \rangle\}$ are jointly Gaussian, zero mean, with

$$\text{cov}(N_i, N_j) = \sigma^2 \langle s_i, s_j \rangle = \sigma^2 E_s \delta_{ij}$$

Thus, $N_i \sim N(0, \sigma^2 E_s)$ are i.i.d. We therefore infer that, conditioned on $s_1$ sent, the $\{Z_i\}$ are conditionally independent, with $Z_i \sim N(E_s, \sigma^2 E_s)$, and $Z_i \sim N(0, \sigma^2 E_s)$ for $i = 2, ..., M$.

**(b)** To express in scale-invariant terms, replace $Z_i$ by $\frac{Z_i}{\sigma\sqrt{E_s}}$. This gives $Z_1 \sim N(m, 1)$, $Z_2, ..., Z_m \sim N(0, 1)$, conditionally independent, where

$$m = \frac{E_s}{\sigma\sqrt{E_s}} = \sqrt{\frac{E_s}{\sigma^2}} = \sqrt{2E_s/N_0}$$

The conditional probability of *correct reception* is now given by

$$\begin{aligned} P_{c|1} &= P[Z_2 \leq Z_1, ..., Z_M \leq Z_1 | H_1] = \int P[Z_2 \leq x, ..., Z_M \leq x | Z_1 = x, H_1] p_{Z_1|H_1}(x|H_1) dx \\ &= \int P[Z_2 \leq x | H_1] ... P[Z_M \leq x | H_1] p_{Z_1|H_1}(x|H_1) dx \end{aligned}$$

where we have used the conditional independence of the $\{Z_i\}$. The desired expression now follows from plugging in the conditional distributions.

The conditional probability of *error* is, of course, one minus the preceding expression. But for small error probabilities, the probability of correct reception is close to one, and it is difficult to get good estimates of the error probability. We therefore develop an expression for the error probability that can be directly computed, as follows:

$$P_{e|1} = \sum_{j \neq 1} P[Z_j = \max_i Z_i | H_1] = (M-1) P[Z_2 = \max_i Z_i | H_1]$$

where we have used symmetry. Now,

$$\begin{aligned} P[Z_2 = \max_i Z_i | H_1] &= P[Z_1 \leq Z_2, Z_3 \leq Z_2, ..., Z_M \leq Z_2 | H_1] \\ &= \int P[Z_1 \leq x, Z_3 \leq x, ..., Z_M \leq x | Z_2 = x, H_1] p_{Z_2|H_1}(x|H_1) dx \\ &= \int P[Z_1 \leq x | H_1] P[Z_3 \leq x | H_1] ... P[Z_M \leq x | H_1] p_{Z_2|H_1}(x|H_1) dx \end{aligned}$$

Plugging in the conditional distributions, and multiplying by $M-1$, gives the desired expression for error probability.

**(c)** See Figure 3.20.

**Problem 3.26 (a)** Replace $x - m$ by $x$ in the result of Problem 3.25(b). This change of variables yields the desired expression

$$P_c = \int_{-\infty}^{\infty} [\Phi(x+m)]^{M-1} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \, dx$$

where $m = \sqrt{2E_s/N_0} = \sqrt{2E_b \log_2 M / N_0} = a\sqrt{\log M}$, where $a = \sqrt{2E_b \log 2 / N_0}$.

**(b)** We wish to find

$$A = \lim_{M \to \infty} \Phi^{M-1}(x + a\sqrt{\log M}) = \lim_{M \to \infty} \Phi^M(x + a\sqrt{\log M})$$

Taking logarithms, we obtain

$$
\begin{aligned}
\log A &= \lim_{M\to\infty} \frac{\log \Phi(x+a\sqrt{\log M})}{1/M} = \lim_{M\to\infty} \frac{\frac{1}{\sqrt{2\pi}} e^{-(x+a\sqrt{\log M})^2/2} \frac{a}{2\sqrt{\log M}} \frac{1}{M}}{-\frac{1}{M^2}} \\
&= \lim_{M\to\infty} -M e^{-\frac{a^2 \log M}{2}} \times \text{terms with slow variation in } M
\end{aligned}
$$

That is, the term dominating $\log A$ is $-M^{1-a^2/2}$, which gives

$$
\log A = \begin{cases} -\infty, & 1 - a^2/2 > 0 \\ 0, & 1 - a^2/2 < 0 \end{cases}
$$

The condition $1 - a^2/2 > 0$ corresponds to $\frac{E_b}{N_0} < \log 2$. We therefore get the result that

$$
A = \lim_{M\to\infty} \Phi^{M-1}\left(x + \sqrt{\frac{2E_b \log M}{N_0}}\right) = \begin{cases} 0, & E_b/N_0 < \log 2 \\ 1, & E_b/N_0 > \log 2 \end{cases}
$$

**(c)** Substitute the preceding limit into the expression for $P_c$ to infer the desired result. Exchanging limit and integral in this fashion can be justified by standard tools of real analysis (specifically, the dominated convergence theorem), which are beyond our scope.

**Problem 3.27:** See Section 8.2.1.

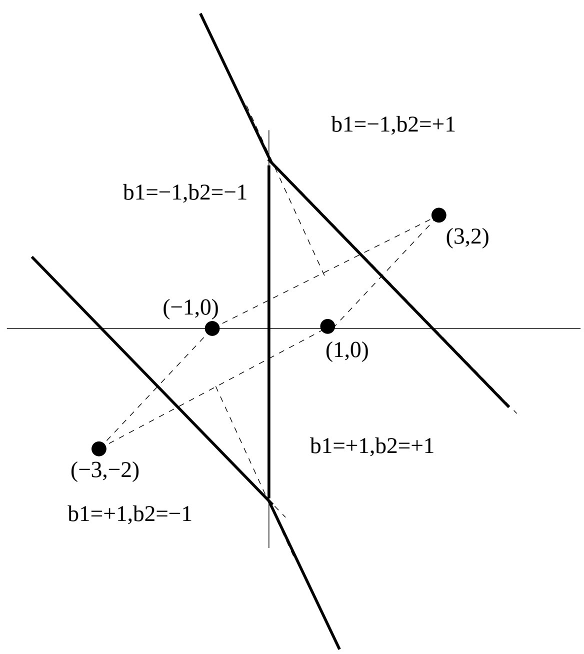



Figure 13: ML decision regions for Problem 3.28

**Problem 3.28** The 2-user system can be written in our $M$-ary hypothesis testing framework as

$$
\mathbf{y} = \mathbf{s}(b_1, b_2) + \mathbf{n}
$$

where $\mathbf{s}(b_1, b_2) = b_1\mathbf{u}_1 + b_2\mathbf{u}_2$ takes one of 4 values. The signal points and the ML decision regions are shown in Figure 13.

**(b)** The decision region for $b_1 = +1$, $b_2 = +1$, which corresponds to the signal point $(1,0)^T$, involves all three neighbors. The distances from the neighbors are given by $d^2 = 4, 8, 20$, and the intelligent union bound is given by the sum of $Q(\frac{d}{2\sigma})$ terms:

$$P(e|+1,+1) \leq Q\left(\frac{1}{\sigma}\right) + Q\left(\frac{\sqrt{2}}{\sigma}\right) + Q\left(\frac{\sqrt{5}}{\sigma}\right)$$

For $b_1 = +1$, $b_2 = -1$, the signal point is $(-3,-2)$. The ML decision region is determined by two neighbors at distances given by $d^2 = 8, 20$, and the intelligent union bound is

$$P(e|+1,-1) \leq Q\left(\frac{\sqrt{2}}{\sigma}\right) + Q\left(\frac{\sqrt{5}}{\sigma}\right)$$

**(c)** Conditioned on $b_1 = +1$, an error occurs if the ML rule chooses one of the two signal points corresponding to $b_1 = -1$. Suppose that $b_1 = +1$. $b_2 = +1$ is sent. An intelligent union bound on choosing one of the two "bad" signal points is given by

$$P(b_1 \text{ wrong}|b_1 = +1, b_2 = +1) \leq Q\left(\frac{1}{\sigma}\right) + Q\left(\frac{\sqrt{2}}{\sigma}\right)$$

Now, suppose that $b_1 = +1$. $b_2 = -1$ is sent. The intelligent union bound on choosing one of the two bad signal points is:

$$P(b_1 \text{ wrong}|b_1 = +1, b_2 = -1) \leq Q\left(\frac{\sqrt{2}}{\sigma}\right)$$

Averaging, we obtain that

$$P_{e,1} = P(b_1 \text{ wrong} \leq \frac{1}{2}Q\left(\frac{1}{\sigma}\right) + Q\left(\frac{\sqrt{2}}{\sigma}\right)$$

**(d)** Figure 14 shows the ML decision regions for the singleuser system. Clearly,

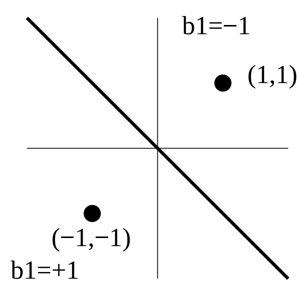



Figure 14: ML decision region in Problem 3.28 if only user 1 is active

$$P_{e,1}^{su} = Q\left(\frac{\sqrt{2}}{\sigma}\right)$$

**(d)** Using $Q(x) \doteq e^{-x^2/2}$, we have $P_{e,1} \doteq e^{-\frac{1}{2\sigma^2}}$, $P_{e,1}^{su} \doteq e^{-\frac{1}{\sigma^2}}$, which gives an asymptotic efficiency of $\frac{1}{2}$.
**(f)** It can be checked that

$$P_{e,2}^{su} = Q\left(\frac{\sqrt{5}}{\sigma}\right) \doteq e^{-\frac{25}{2\sigma^2}}$$

and that this is also the dominant term in the intelligent union bound for error in $b_2$ in the 2-user system. Thus, the asymptotic efficiency for user 2 is one.

**Problem 3.29** The *noise power* is given by

$$P_n = kT10^{FdB/10}B = (1.38 \times 10^{-23}) \times 290 \times 10^{7/10} \times (3 \times 10^6) = 6 \times 10^{-14} \text{ watts}$$

That is, $P_n = -102.2$ dBm.
The *symbol rate* for 50% excess bandwidth is $3MHz/(1+0.5)$, or $R_s = 2$ MHz.
**Bit rate computations:** The ratio of bit rate to symbol rate is 2 for QPSK, 3 for 8PSK, 6 for 64QAM, and $\frac{\log_2 16}{8} = \frac{1}{2}$ for 16-ary orthogonal signaling. The last computation assumes coherent orthogonal signaling, so that 16 orthogonal signals require 8 complex dimensions. This gives bits rates $R_b$ of 4 Mbps (QPSK), 6 Mbps (8PSK), 12 Mbps (64QAM) and 1 Mbps (16-ary orthogonal).
$(Eb/N0)_{reqd}$ **computations:** The required BER is $10^{-8}$. We note that $Q(5.62) \approx 10^{-8}$.
*QPSK*: We have $R_b = 2R_s = 4$ Mb/s. The BER with Gray coding is

$$P_b(QPSK) = Q\left(\sqrt{2E_b/N_0}\right)$$

which yields that $(Eb/N0)_{reqd}$ is about 12 dB for a BER of $10^{-8}$.
*8PSK and 64QAM* With Gray coding, the nearest neighbors approximation to BER is

$$P_b(8PSK) \approx Q\left(\sqrt{\eta_P E_b/2N_0}\right)$$

where $\eta_P = \frac{d_{min}^2}{E_b}$ equals $3(2-\sqrt{2})$ for 8PSK, and 4/7 for 64QAM. This yields that, for BER of $10^{-8}$, $(Eb/N0)_{reqd}$ is about 15.6 dB for 8PSK, and 20.4 dB for 64QAM.
*16-ary orthogonal signaling:* The BER is approximately given by

$$P_b(16 - ary\ orthog) \approx 8Q\left(\sqrt{4E_b/N_0}\right)$$

Noting that $8Q(5.97) \approx 10^{-8}$, we obtain that , for BER of $10^{-8}$, $(Eb/N0)_{reqd}$ is about 9.5 dB.
**Receiver sensitivity computations:**
The receiver sensitivity is given by

$$P_{RX,min}(dBm) = P_n(dBm) + (Eb/N0)_{reqd}(dB) + 10\log_{10} R_b/B$$

This evaluates to -89 dBm (QPSK), -83.6 dBm (8PSK), -75.8 dBm (64QAM) and -97.5 dBm (16-ary orthogonal signaling).

**Problem 3.30**
**(a)** The symbol rate remains 15Mbps, hence the bit rate goes up to 90 Mbps (6 bits/symbol) with 64QAM. The power efficiency of 64QAM is $\eta_P(64QAM) = \frac{4}{7}$, compared to $\eta_P(QPSK) = 4$. The receiver sensitivity therefore goes up by a factor of 7 (i.e., by 8.45 dB), and the range down by a factor of $\sqrt{7}$ to $107/\sqrt{7} = 40.4$ meters.
**(b)** The BER of Gray coded QPSK with Rayleigh fading is given by

$$P_b(QPSK) = \frac{1}{2}\left(1 - (1 + N_0/\bar{E}_b)^{-\frac{1}{2}}\right)$$

For a BER of $10^{-6}$, the $\bar{E}_b/N_0$ required is about 54 dB. This is about 44 dB worse than that for the AWGN channel considered in the example, so that the range attained is about 22 dB smaller, i.e., $107/10^2.2 = 0.68$ meters. That is, insisting on a very low BER with Rayleigh fading makes the system essentially useless.

**Problems 3.31-3.32:** The steps are clearly outlined in the problem statements.

## Solutions to Chapter 4 Problems

*Fundamentals of Digital Communication*

**Problem 4.1:** (i) **Known symbols:** ince $N[k] \sim N(0, \sigma^2)$, we have

$$p(Y[k]|A, b[k]) = \frac{e^{-\frac{(y-Ab[k])^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}$$

so that the cost function to be maximized over $A$ is

$$J(A) = \sum_{k=1}^{K} \log p(Y[k]|A, b[k])) = \sum_{k=1}^{K} \left\{ -\frac{(y - Ab[k])^2}{2\sigma^2} - \frac{1}{2}\log(2\pi\sigma^2) \right\}$$

This is a quadratic function of $A$, and the global maximum can be found by differentiation: $\frac{dJ(A)}{dA} = 0$. Ignoring scalar multiples, this yields

$$\sum_{k=1}^{K} b[k](y - Ab[k]) = 0$$

Since $b^2[k] \equiv 1$, the solution to the preceding equation is given by

$$\hat{A}_{ML} = \frac{1}{K} \sum_{k=1}^{K} Y[k]b[k]$$

(ii) **Unknown symbols:** From (4.8), we have that the conditional density of $Y[k]$, conditioned on $A$, after averaging out $b[k]$ is

$$p(Y[k]|A) = e^{-\frac{A^2}{2\sigma^2}} \cosh\left(\frac{AY[k]}{\sigma^2}\right) \frac{e^{-\frac{Y^2[k]}{2\ \sigma^2}}}{\sqrt{2\pi\sigma^2}}$$

The cost function to be maximized over $A$ is

$$J(A) = \sum_{k=1}^{K} \log p(Y[k]|A)) = -K\frac{A^2}{2\sigma^2} + \sum_{k=1}^{K} \log\cosh\left(\frac{AY[k]}{\sigma^2}\right) \ + \text{ terms independent of } A$$

Setting the derivative with respect to $A$ equal to zero, we obtain

$$-KA/\sigma^2 + sum_{k=1}^{K} \frac{\sinh\left(\frac{AY[k]}{\sigma^2}\right) Y[k]/\sigma^2}{\cosh\left(\frac{AY[k]}{\sigma^2}\right)}$$

which yields that $\hat{A}_{ML}$ is the solution to the equation

$$A = \frac{1}{K} \sum_{k=1}^{K} Y[k] \tanh\left(\frac{AY[k]}{\sigma^2}\right)$$

which, as pointed out in the text, can be viewed as correlating the observations with soft decisions on the symbols.

**Problem 4.2:** (a) The likelihood function conditioned on $A$, $\theta$ and $b[k]$ is given by

$$L(Z[k]|A, \theta, b[k]) = \exp\left(\tfrac{1}{\sigma^2}\left[\mathrm{Re}\langle Z[k], Ae^{j\theta}b[k]\rangle - \tfrac{|Ae^{j\theta}b[k]|^2}{2}\right]\right)$$
$$= \exp\left(\tfrac{1}{\sigma^2}\left[\mathrm{Re}(|Z[k]|e^{j\phi_k}Ae^{-j\theta}b^*[k]) - \tfrac{|A|^2}{2}\right]\right)$$

where $\phi_k = \arg(Z[k])$. Averaging over $b[k]$, which takes values equiprobably in $\{\pm 1, \pm j\}$, we get

$$\begin{aligned} L(Z[k]|A,\theta) &= \tfrac{1}{4}\exp\left(\tfrac{1}{\sigma^2}\left[A|Z[k]|\cos(\theta-\phi_k) - \tfrac{|A|^2}{2}\right]\right) + \tfrac{1}{4}\exp\left(\tfrac{1}{\sigma^2}\left[-A|Z[k]|\cos(\theta-\phi_k) - \tfrac{|A|^2}{2}\right]\right) \\ &\quad + \tfrac{1}{4}\exp\left(\tfrac{1}{\sigma^2}\left[A|Z[k]|\sin(\theta-\phi_k) - \tfrac{|A|^2}{2}\right]\right) + \tfrac{1}{4}\exp\left(\tfrac{1}{\sigma^2}\left[-A|Z[k]|\sin(\theta-\phi_k) - \tfrac{|A|^2}{2}\right]\right) \\ &= \tfrac{1}{2}e^{-\frac{A^2}{2\sigma^2}}\left[\cosh\left(\tfrac{A}{\sigma^2}|Z[k]|\cos(\theta-\phi_k)\right) + \cosh\left(\tfrac{A}{\sigma^2}|Z[k]|\sin(\theta-\phi_k)\right)\right] \end{aligned}$$

where we have dropped constants that do not affect the optimization. (b) Since $Z[1],...,Z[K]$ are conditionally independent given $A, \theta$, we obtain

$$\begin{aligned} \log(Z[1],...,Z[K]|A,\theta) &= \textstyle\sum_{k=1}^K \log(Z[k]|A,\theta) \\ &= -K\tfrac{A^2}{2\sigma^2} + \textstyle\sum_{k=1}^K \log\left(\cosh\left(\tfrac{A|Z[k]|\cos(\theta-\phi_k)}{\sigma^2}\right) + \cosh\left(\tfrac{A|Z[k]|\sin(\theta-\phi_k)}{\sigma^2}\right)\right) = J(A,\theta) \end{aligned}$$

(c) If $\theta$ is replaced by $\theta + \frac{\pi}{2}$, we have $\cos(\theta + \pi/2 - \phi_k) = -\sin(\theta - \phi_k)$ and $\sin(\theta + \pi/2 - \phi_k) = \cos(\theta - \phi_k)$. Since $\cosh(\cdot)$ is an even function, it is easy to see that the log likelihood function in (b) remains unchanged. We can therefore only estimate $\theta$ modulo $\pi/2$, and can restrict attention to $\theta \in [0, \pi/2)$.
(d) We have

$$\mathbb{E}[|Z[k]|^2] = \mathbb{E}[|Ae^{j\theta}b[k] + N[k]|^2] = \mathbb{E}\left[|A|^2 + |N[k]|^2 + Ae^{-j\theta}b^*[k]N[k] + Ae^{j\theta}b[k]N^*[k]\right] = A^2 + 2\sigma^2$$

with the last two terms dropping out because $N[k]$, $b[k]$ are independent, zero mean, random variables. This motivates the estimate

$$\hat{A}^2 = \max\{\frac{1}{K}\sum_{k=1}^K |Z[k]|^2 \ - \ 2\sigma^2, 0\}$$

where we replace the expectation in (d) by an empirical average, and constrain the estimate to be nonnegative.
(e) We are given the following observations:

| k | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| $Z[k]$ | $-0.1 + 0.9j$ | $1.2 + 0.2j$ | $0.3 - 1.1j$ | $-0.8 + 0.4j$ |
| $\lvert Z[k]\rvert$ | | | | |
| $\phi_k$ | | | | |

Table 1: Observations.

There are several ways of computing the estimates. Since $\theta$ is restricted to $[0, \pi/2)$, we can simply carry out a brute force maximization of $J(A, \theta)$ over $\theta$ for fixed $A$. The suboptimal estimate of $A$ in (d) actually works quite well. However, once we have found $\theta$ by brute force search, we can choose $A$ to satisfy $0 = \frac{\partial J(A,\theta)}{\partial A}$, which yields the transcendental equation

$$A = \frac{1}{K}\sum_{k=1}^K |Z[k]| \frac{\cos(\theta-\phi_k)\cosh\left(\frac{A|Z[k]|\cos(\theta-\phi_k)}{\sigma^2}\right) + \sin(\theta-\phi_k)\cosh\left(\frac{A|Z[k]|\sin(\theta-\phi_k)}{\sigma^2}\right)}{\cosh\left(\frac{A|Z[k]|\cos(\theta-\phi_k)}{\sigma^2}\right) + \cosh\left(\frac{A|Z[k]|\sin(\theta-\phi_k)}{\sigma^2}\right)}$$

We obtain $\hat{A} = 1$ and $\hat{\theta} = 0.06$ radians.

**Problem 4.3:** Setting $s(t) = \sum_k b[k]p(t - kT)e^{j\theta}$., the log likelihood function is given by

$$L(y|\theta, \mathbf{b}) = \exp\left(\frac{1}{\sigma^2}\left[\mathrm{Re}\langle y, s\rangle - \frac{||s||^2}{2}\right]\right)$$

The observation-dependent part of the likelihood function is

$$\begin{aligned}\mathrm{Re}\langle y, s\rangle &= \int y(t)s^*(t)dt = \sum_k b^*[k]e^{-j\theta} \int y(t)p^*(t-kT)dt \\ &= \sum_k b^*[k]e^{-j\theta} z[k]\end{aligned}$$

where $z[k] = z(kT) = (y * p_{mf})(t)$, where $p_{mf}(t) = p^*(-t)$ is matched to the pulse $p$.
(b) Since $||s||^2$ does not depend on $\theta$, we need to maximize $\mathrm{Re} y, s\rangle$ over $\theta$. Setting $\sum_k b^*[k]z[k] = Re^{j\phi}$, we have

$$\mathrm{Re}\langle y, s\rangle = \mathrm{Re}\left(Re^{j\phi}e^{-j\theta}\right) = R\cos(\phi - \theta)$$

which is maximized by

$$\hat{\theta}_{ML} = \phi = \arg\left(\sum_k b^*[k]z[k]\right)$$

(c) We have

$$J_k(\theta) = \mathrm{Re}\left(b^*[k]z[k]e^{-j\theta}\right) = b^*[k]z[k]e^{-j\theta} + b[k]z^*[k]e^{j\theta}$$

ignoring additive and multiplicative constants independent of $\theta$. Differentiating with respect to $\theta$, we obtain

$$\begin{aligned}\frac{\partial J_k(\theta)}{\partial \theta} &= -jb^*[k]z[k]e^{-j\theta} + jb[k]z^*[k]e^{j\theta} \\ &= \mathrm{Re}\left(-jb^*[k]z[k]e^{-j\theta}\right) = \mathrm{Im}\left(b^*[k]z[k]e^{-j\theta}\right)\end{aligned}$$

upto scalar multiple.
(d) We now wish to maximize $L(y|\theta, \mathbf{b})$ over $\mathbf{b}$ for known $\theta$. Assuming that $p$ is square root Nyquist (i.e., that $p * p_{mf}$ is Nyquist at rate $\frac{1}{T}$), we can show that $||s||^2 = ||e^{j\theta}\sum_k b[k]p(t-kT)||^2$ does not depend on $\mathbf{b}$. We must therefore maximize

$$\mathrm{Re}\langle y, s\rangle = \mathrm{Re}\left(\sum_k b^*[k]e^{-j\theta}z[k]\right)$$

This can be maximized term by term as follows: for BPSK, $\hat{b}[k] = +1$ if

$$\mathrm{Re}(e^{-j\theta}z[k]) > \mathrm{Re}(-e^{-j\theta}z[k])$$

and $\hat{b}[k] = -1$ otherwise. This corresponds to $\hat{b}[k] = \mathrm{sign}\left(\mathrm{Re}(e^{-j\theta}z[k])\right)$. Now, for decision-directed adaptation, we have, specializing (c) for real-valued symbols, that (ignoring multiplicative constants)

$$\frac{\partial J_k(\theta)}{\partial \theta} = \hat{b}[k]\mathrm{Im}\left(z[k]e^{-j\theta}\right) = \mathrm{sign}\left(\mathrm{Re}(e^{-j\theta}z[k])\right)\mathrm{Im}\left(z[k]e^{-j\theta}\right)$$

Plugging in the BPSK decision rule, the desired ascent-based tracking algorithm immediately follows.
(e) Assuming that $p$ is square root Nyquist, the term $||s||^2$ in the likelihood ratio does not depend on $\mathbf{b}$, and can be dropped. We therefore need to average $\exp\left(\frac{1}{\sigma^2}\mathrm{Re} y, s\rangle\right)$ over $\mathbf{b}$ for obtaining the likelihood function $L(y|\theta)$ for NDA estimation. For the $k$th bit, this yields

$$\mathbb{E}_{b[k]}\left[\exp\left(\mathrm{Re}(b^*[k]z[k]e^{-j\theta})\right)\right] = \cosh\left(\frac{\mathrm{Re}(z[k]e^{-j\theta})}{\sigma^2}\right)$$

so that

$$\log L(y|\theta) = \sum_k \log\cosh\left(\frac{\mathrm{Re}(z[k]e^{-j\theta})}{\sigma^2}\right)$$

(f) Differentiating the log likelihood function using the chain rule, we have

$$\frac{\partial \log L(y|\theta)}{\partial \theta} = \sum_k \frac{\sinh\left(\frac{\mathrm{Re}(z[k]e^{-j\theta})}{\sigma^2}\right)}{\cosh\left(\frac{\mathrm{Re}(z[k]e^{-j\theta})}{\sigma^2}\right)} \frac{\mathrm{Im}(z[k]e^{-j\theta})}{\sigma^2}$$

since

$$\frac{\partial \mathrm{Re}(z[k]e^{-j\theta})}{\partial\theta} = \frac{1}{2}\frac{\partial}{\partial\theta}\left(z[k]e^{-j\theta} + z^*[k]e^{j\theta}\right)$$

$$= \frac{1}{2}\left(-jz[k]e^{-j\theta} + jz^*[k]e^{j\theta}\right) = \mathrm{Im}(z[k]e^{-j\theta})$$

Setting the derivative equal to zero, we obtain that the ML estimate is the solution to

$$\sum_k \tanh\left(\frac{\mathrm{Re}(z[k]e^{-j\hat{\theta}_{ML}})}{\sigma^2}\right)\mathrm{Im}(z[k]e^{-j\hat{\theta}_{ML}}) = 0$$

(g) We now take only the derivative of the $k$th term in order to obtain a tracking algorithm. The work done in (f) shows that

$$\frac{\partial J_k(\theta)}{\partial\theta} = \frac{\partial \log L(z[k]|\theta)}{\partial\theta} = a\tanh\left(\frac{\mathrm{Re}(z[k]e^{-j\hat{\theta}_{ML}})}{\sigma^2}\right)\mathrm{Im}(z[k]e^{-j\hat{\theta}_{ML}})$$

where $a > 0$ is a constant. The NDA ascent algorithm is therefore given by

$$\hat{\theta}[k+1] = \theta[k] + b\frac{\partial J_k(\theta)}{\partial\theta}|_{\theta=\theta[k]} = \theta[k] + \alpha\ \tanh\left(\frac{\mathrm{Re}(z[k]e^{-j\theta[k]})}{\sigma^2}\right)\ \mathrm{Im}(z[k]e^{-j\theta[k]})$$

where $b, \alpha > 0$ are constants. The argument of tanh in the preceding equation gets large at high SNR ($\sigma^2 \to 0$. Since $\tanh(x) \approx \mathrm{sign}(x)$ for $|x|$ large, we see that the NDA tracking algorithm reduces to the decision-directed algorithm in (g) at high SNR.

At low SNR, the argument of tanh is small. Since $\tanh(x) \approx x$ as $|x| \to 0$, we obtain the tracking rule

$$\hat{\theta}[k+1] = \theta[k] + \beta\ \mathrm{Re}(z[k]e^{-j\theta[k]})\mathrm{Im}(z[k]e^{-j\theta[k]}) \qquad \textbf{low SNR approximation}$$

where the new constant $\beta$ is related to $\alpha$ as $\beta = \alpha/\sigma^2$. This is the classic Costas loop.

**Problem 4.4:** The samples are modeled as

$$y[k] = b[k]e^{j(\Gamma k+\theta)} + N[k]\ , \quad k = 1, ..., K$$

(a) Ignoring additive constants, the log likelihood ratio is given by

$$\log L(\mathbf{y}|\Gamma,\theta) = \frac{1}{\sigma^2}\sum_{k=1}^{K}\left(\mathrm{Re}(y[k]b^*[k]e^{-j(\Gamma k+\theta)}) - |b[k]e^{j(\Gamma k+\theta)}|^2/2\right) = \frac{1}{\sigma^2}\sum_{k=1}^{K}\left(\mathrm{Re}(y[k]b^*[k]e^{-j(\Gamma k+\theta)}) - |b[k]|^2/2\right)$$

The second term can be dropped, since $\{b[k]\}$ are known. Ignoring multiplicative constants, the function to be maximized is given by

$$J(\Gamma,\theta) = \sum_{k=1}^{K}\mathrm{Re}(y[k]b^*[k]e^{-j(\Gamma k+\theta)}) = \mathrm{Re}\left(e^{-j\theta}\sum_{k=1}^{K}y[k]b^*[k]e^{-j\Gamma k}\right)$$

(b) Let $Z(\Gamma) = \sum_{k=1}^{K} y[k]b^*[k]e^{-j\Gamma k}$, it is easy to see that

$$J(\Gamma,\theta) = |Z(\Gamma)|\cos\left(\arg(Z(\Gamma)) - \theta\right)$$

which is maximized by $\theta = \arg(Z(\Gamma))$. Substituting into the cost function, we have (reusing the notation for the cost function):

$$J(\Gamma) = \max_\theta J(\Gamma,\theta) = |Z(\Gamma)| = |\sum_{k=1}^{K} y[k]b^*[k]e^{-j\Gamma k}|$$

(c) The structure of the cost function is that of maximizing the magnitude of a DFT coefficient. Specifically, if we wish to estimate $\Gamma$ up to a resolution of $\frac{1}{N}$, we can take an $N$-point ($N \geq K$) DFT of the sequence $\{y[k]b^*[k]\}$ padded with $N-K$ zeros. The $n$th DFT coefficient is given by

$$Z[n] = \sum_{k=1}^{K} y[k]b^*[k]e^{-j2\pi nk/N} \ , \quad n = 0, 1, ..., N-1$$

If the maximum of $\{|Z[n]|\}$ occurs at $n_{max}$, we obtain the estimate $\hat{\Gamma} = \frac{2\pi n_{max}}{N}$. Note that $\hat{\Gamma} \in [0, 2\pi)$. If the frequency offset $\Gamma = 2\pi\Delta fT$ is large enough to induce more than an offset of $2\pi$ per symbol interval, then we can only estimate it modulo $2\pi$. This is a fundamental limitation of our observation model, since we are sampling at the symbol rate.

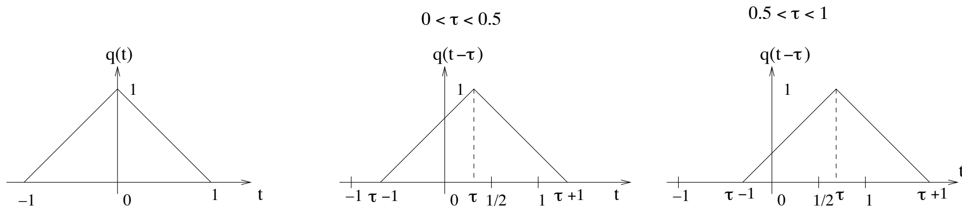



Figure 1: The convolution of $p(t)$ with its matched filter, and its delayed versions (Problem 4.5).

**Problem 4.5:** The convolution $q(t) = (p * p_{MF})(t) = (1-|t|)I_{[-1,1]}(t)$ is shown in Figure 1. When the signal is delayed by $\tau$, the output of the matched filter is given by

$$z(t) = q(t-\tau) + w(t)$$

where $w(t) = (n * p_{MF})(t)$ is zero mean WSS Gaussian noise with autocorrelation function $R_w(\tau) = \sigma^2 q(\tau)$. The random process $z(t)$ (for any fixed $\tau$) is therefore a Gaussian random process with mean function $m_z(t) = q(t-\tau)$ and autocovariance function $C_z(\tau) = R_w(\tau) = \sigma^2 q(\tau)$. The vector of samples $\mathbf{z}$ is therefore Gaussian with mean $\mathbf{m}(\tau) = (q(-\tau), q(\frac{1}{2}-\tau), q(1-\tau))^T$ and covariance matrix

$$\mathbf{C} = \sigma^2 \begin{pmatrix} 1 & \frac{1}{2} & 0 \\ \frac{1}{2} & 1 & \frac{1}{2} \\ 0 & \frac{1}{2} & 1 \end{pmatrix}$$

so that

$$p(\mathbf{z}|\tau) = \frac{1}{(2\pi)^{3/2}|\mathbf{C}|^{1/2}} \exp\left(-(\mathbf{z} - \mathbf{m}(\tau))^T \mathbf{C}^{-1} (\mathbf{z} - \mathbf{m}(\tau))\right)$$

From Figure 1, we see that the mean vector is nonzero only if $\tau \in (0, 1)$. For $0 \leq \tau \leq \frac{1}{2}$, we have $\mathbf{m} = (1-\tau, \frac{1}{2}+\tau, \tau)^T$. For $\frac{1}{2} \leq \tau \leq 1$, we have $\mathbf{m} = (1-\tau, \frac{3}{2}-\tau, \tau)^T$.
(c) The ML estimate is given by $\hat{\tau}_{ML} = \arg\max_\tau p(\mathbf{z}|\tau)$, or $\hat{\tau}_{ML} = \arg\min_\tau J(\tau)$, where

$$J(\tau) = (\mathbf{z} - \mathbf{m}(\tau))^T \mathbf{C}^{-1} (\mathbf{z} - \mathbf{m}(\tau))$$

Note that

$$\mathbf{C}^{-1} = \sigma^{-2} \begin{pmatrix} \frac{3}{2} & -1 & \frac{1}{2} \\ -1 & 2 & -1 \\ \frac{1}{2} & -1 & \frac{3}{2} \end{pmatrix}$$

We have $\mathbf{z} = (0.7, 0.8, -0.1)^T$. For $0 \leq \tau \leq \frac{1}{2}$, we have $\mathbf{z} - \mathbf{m}(\tau) = (\tau - 0.3, 0.3 - \tau, -0.1 - \tau)^T$, which yields

$$J(\tau) = (4\tau^2 - 2.4\tau + 0.6)/\sigma^2$$

which attains its minimum at $\tau = 0.3$. For $\frac{1}{2} \leq \tau \leq 1$, we have $\mathbf{z} - \mathbf{m}(\tau) = (\tau - 0.3, \tau - 0.7, -0.1 - \tau)^T$, which yields

$$J(\tau) = (4\tau^2 + 0.95\tau + 0.735)/\sigma^2$$

This is a quadratic function which attains its minimum for $\tau < 0$. Thus, in the prescribed range $[\frac{1}{2}, 1]$, the minimum value is at one of the extreme points, specifically at $\tau = \frac{1}{2}$. It can be checked that the value of the cost function at $\tau = \frac{1}{2}$ is bigger than that at $\tau = 0.3$. We therefore obtain that

$$\hat{\tau}_{ML} = 0.3$$

**Problem 4.6:** (a) From Example 4.3.2, we know that the ML delay estimate corresponds to picking the peak of the matched filter

$$s_{MF}(t) = s^*(-t) = \sum_k b^*[k] p^*(-t - kT) = \sum_k b^*[k] p_{MF}(t + kT)$$

The output of the matched filter can therefore be written as

$$(y * s_{MF})(t) = \sum_k b^*[k] (y * p_{MF})(t + kT) = \sum_k b^*[k] z(t + kT)$$

where $z(t) = (y * p_{MF})(t)$ is the output of the pulse matched filter.
(b) If we only have access to $\{z(mT/2), m \text{ integer}\}$, then we can compute $(y * s_{MF})(t)$ for values of $t$ which are integer multiples of $\frac{T}{2}$. We could, of course, restrict our search for $\hat{\tau}$ to these values. However, a more accurate, although computationally more intensive, approach is to model the distribution of these sampled outputs, conditioned on $\tau$, and to then find an ML estimate for $\tau$. Various approximations can be devised based on this approach.

**Problem 4.7:** (a) Under $H_0$, $Y \sim CN(0, 2\sigma^2)$. Since $h \sim CN(0, 1)$, under $H_1$, $Y \sim CN(0, A^2 + 2\sigma^2)$. The ML rule is therefore given by

$$\frac{1}{\pi(A^2 + 2\sigma^2)} e^{-\frac{|y|^2}{A^2 + 2\sigma^2}} \underset{H_0}{\overset{H_1}{\gtrless}} \frac{1}{\pi(2\sigma^2)} e^{-\frac{|y|^2}{2\sigma^2}}$$

Taking log and ignoring additive constants, we obtain

$$-\frac{|y|^2}{A^2 + 2\sigma^2} - \log(A^2 + 2\sigma^2) \underset{H_0}{\overset{H_1}{\gtrless}} -\frac{|y|^2}{2\sigma^2} - \log(2\sigma^2)$$

which simplifies to

$$|y|^2 \underset{H_0}{\overset{H_1}{\gtrless}} 2\sigma^2 (1 + \frac{2\sigma^2}{A^2}) \log(\frac{A^2}{2\sigma^2} + 1)$$

b) Under $H_0$, $|Y|^2$ has an exponential distribution with mean $2\sigma^2$. The conditional error probability

$$P_{e|0} = P[|Y|^2 > \gamma | H_0] = e^{-\frac{\gamma}{2\sigma^2}}$$

where $\gamma = 2\sigma^2 (1 + \frac{2\sigma^2}{A^2}) \log(\frac{A^2}{2\sigma^2} + 1)$. This simplifies to

$$P_{e|0} = \left( \frac{A^2}{2\sigma^2} + 1 \right)^{-\left(1 + \frac{2\sigma^2}{A^2}\right)}$$

(c) For coherent detection, we have the ML rule

$$\mathrm{Re}(y(hA)^*) - |hA|^2/2 \underset{H_0}{\overset{H_1}{\gtrless}} 0$$

(The value of $\sigma^2$ is not needed.) For the given values, the left-hand side simplifies as

$$\mathrm{Re}((1+j)(-j)(3/2)) - (3/2)^2/2 = 3/2 - 9/8 > 0$$

so that the ML decision is to say $H_1$.

**Problem 4.8:** The solution steps are clearly outlined in the problem statement.

**Problem 4.9:** The solution steps are clearly outlined in the problem statement. We only provide the following answer to a question asked in part (c). As $m \to \infty$, it is easy to see that $R_1 \approx m + V_1$, so that $U = R_1 - m \approx V_1$. Thus, we expect $U \to V_1 \sim N(0,1)$ as $m \to \infty$. The convergence is both almost surely and in the mean square sense, but we will not worry about such technical details.

**Problem 4.10:** (a) Conditioned on $H_i$, $A$ and $\theta$, the likelihood function is given by

$$L(y|H_i, A, \theta) = \exp\left(\frac{1}{\sigma^2}\left[\mathrm{Re}\langle y, As_i e^{j\theta}\rangle - ||As_i e^{j\theta}||^2/2\right]\right) = \exp\left(\frac{1}{\sigma^2}\left[\mathrm{Re}\langle y, As_i e^{j\theta}\rangle - A^2 E_b/2\right]\right)$$

Let $Z_i = \langle y, s_i\rangle = |Z_i|e^{j\phi_i}$, $i = 0, 1$. Then

$$L(y|H_i, A, \theta) = \exp\left(\frac{1}{\sigma^2}\left[A|Z_i|\cos(\phi_i - \theta) - A^2 E_b/2\right]\right)$$

Removing the conditioning on $\theta$, we obtain

$$L(y|H_i, A) = e^{-\frac{A^2 E_b}{2\sigma^2}} I_0\left(\frac{A|Z_i|}{\sigma^2}\right)$$

where $I_0(x) = \frac{1}{2\pi}\int_0^{2\pi} e^{x\cos\alpha} d\alpha$ is increasing in $|x|$. We have used the result here that, modulo $2\pi$, $\phi_i - \theta$ is uniform over $[0, 2\pi]$, since $\theta$ is uniform over $[0, 2\pi]$. The likelihood function is monotone increasing in $|Z_i|$ for any given $A$, so that the MPE rule (which is the ML rule for equal priors) is

$$|Z_1| \underset{H_0}{\overset{H_1}{\gtrless}} |Z_0|$$

(b) The error probability for equal energy, orthogonal signaling is $\frac{1}{2}e^{-\frac{E_s}{2N_0}}$, so that

$$P_{e|A=a} = \frac{1}{2}e^{-\frac{a^2 E_b}{2N_0}}$$

where $E_b$ denotes the *average* energy per bit, averaged over $A$. Removing the conditioning on $A$, we have

$$\begin{aligned} P_e &= \int P_{e|A=a} p(a)\, da = \int_0^\infty \tfrac{1}{2} e^{-\frac{a^2 E_b}{2N_0}} 2a e^{-a^2}\, da = \int_0^\infty a e^{-a^2(1+\frac{E_b}{2N_0})}\, da \\ &= \left.\frac{e^{-a^2(1+\frac{E_b}{2N_0})}}{-2(1+\frac{E_b}{2N_0})}\right|_0^\infty = \frac{1}{2(1+\frac{E_b}{2N_0})} = \frac{1}{\frac{E_b}{N_0}+2} \end{aligned}$$

(c) Since $P_e = P_{e|0}$, we condition on $H_0$ throughout. We have $y = hs_0 + n$, so that

$$Z_0 = \langle y, s_0\rangle = hE_b + \langle n, s_0\rangle\ , \quad Z_1 = \langle y, s_1\rangle = \langle n, s_0\rangle$$

where we have used $\langle s_0, s_1 \rangle = 0$. The random variables $\langle n, s_0 \rangle$ and $\langle n, s_1 \rangle$ are jointly complex Gaussian with

$$\text{cov}(\langle n, s_0 \rangle, \langle n, s_1 \rangle) = 2\sigma^2 \langle s_1, s_0 \rangle = 0$$

Thus, they are independent. Furthermore,

$$\text{cov}(\langle n, s_0 \rangle, \langle n, s_0 \rangle) = 2\sigma^2 \langle s_0, s_0 \rangle = 2\sigma^2 E_b$$

Similarly, $\text{cov}(\langle n, s_1 \rangle, \langle n, s_1 \rangle) = 2\sigma^2 E_b$. Thus, $\langle n, s_0 \rangle$ and $\langle n, s_1 \rangle$ are i.i.d. $CN(0, 2\sigma^2 E_b)$. Since $h$ and $n$ are independent zero mean complex Gaussian, $Z_0 = hE_b + \langle n, s_0 \rangle$ is zero mean complex Gaussian with $\text{cov}(Z_0, Z_0) = \mathbb{E}[|h|^2]E_b^2 + 2\sigma^2 E_b = E_b^2 + 2\sigma^2 E_b$. It is also independent of $Z_1 = \langle n, s_1 \rangle$. Thus, $Z_0 \sim CN(0, E_b^2 + 2\sigma^2 E_b)$ and $Z_1 \sim CN(0, 2\sigma^2 E_b)$ are independent. This implies that $Y_0 = |Z_0|^2$ and $Y_1 = |Z_1|^2$ are independent exponential random variables with means $E_b^2 + 2\sigma^2 E_b = \frac{1}{\mu_0}$ and $2\sigma^2 E_b = \frac{1}{\mu_1}$, respectively. The (conditional) error probability can now be computed as

$$P_{e|0} = P[Y_1 > Y_0 | H_0] = \int P[Y_1 > y_0 | Y_0 = y_0, H_0] p(y_0) dy_0 = \int P[Y_1 > y_0 | H_0] p(y_0) dy_0$$

where we have used the conditional independent of $Y_0$ and $Y_1$. Substituting the distributions, we have, as before, that

$$\begin{aligned} P_{e|0} &= \int_0^\infty e^{-\mu_1 y_0} \mu_0 e^{-\mu_0 y_0} dy_0 = \frac{\mu_0}{\mu_0 + \mu_1} = \frac{\frac{1}{E_b^2 + 2\sigma^2 E_b}}{\frac{1}{E_b^2 + 2\sigma^2 E_b} + \frac{1}{2\sigma^2 E_b}} \\ &= \frac{1}{\frac{E_b}{2\sigma^2} + 2} = \frac{1}{\frac{E_b}{N_0} + 2} \end{aligned}$$

(d) The plot is omitted. The error probability with Rayleigh fading decays inversely as the SNR. This is much slower than the exponential decay of the error probability over the AWGN channel, which is given by $\frac{1}{2} e^{-\frac{E_b}{2N_0}}$. Methods for addressing the severe performance penalty due to Rayleigh fading are discussed in Chapter 8.

**Problem 4.11:** If 1 is sent, then $Y \sim CN(0, 4)$: both $h$ and $N$ are i.i.d. complex Gaussian with

$$\mathbb{E}[|h + N|^2] = \mathbb{E}[|h|^2] + \mathbb{E}[|N|^2] + \mathbb{E}[h^* N] + \mathbb{E}[h N^*] = 3 + 1 + 0 + 0 = 4$$

If 0 is sent, then $Y = N \sim CN(0, 1)$. Thus, we have

$$p(y|1) = \frac{1}{4\pi} e^{-|y|^2/4}, \quad p(y|0) = \frac{1}{\pi} e^{-|y|^2}$$

Using Bayes' rule,

$$\pi(1|y) = P[1 \text{ sent} | Y = y] = \frac{\pi_1 p(y|1)}{\pi_1 p(y|1) + \pi_0 p(y|0)}$$

For $y = 1 - 2j$ (so that $|y|^2 = 5$)and $\pi_1 = 1/3$, we obtain upon substitution that

$$\pi(1|y) = \frac{\frac{1}{3} e^{-5/4}/(4\pi)}{\frac{1}{3} e^{-5/4}/(4\pi) + \frac{2}{3} e^{-5}/\pi} = \frac{1}{1 + 8e^{-15/4}} = 0.842$$

**Problem 4.12:** (a) The error probability conditioned on $b = -1$ and $h = h_0$ is $P[Y > 0 | b = -1, h = h_0] = Q(h_0)$, since $Y \sim N(-h_0, 1)$. The error probability conditioned on $b = -1$ is therefore given by

$$\begin{aligned} P[Y > 0 | b = -1] &= P[h = 1] P[Y > 0 | b = -1, h = 1] + P[h = 2] P[Y > 0 | b = -1, h = 2] \\ &= \tfrac{1}{4} Q(1) + \tfrac{3}{4} Q(2) = P_e \end{aligned}$$

where we have invoked symmetry to infer that the probability of error $P_e$ equals the conditional error probability under either hypothesis. Note that $E_b = \mathbb{E}[h^2] = \frac{1}{4} 1^2 + \frac{3}{4} 2^2 = 3.25$ and that $\sigma^2 = N_0/2 = 1$, so that $E_b/N_0 = 3.25/2 = 1.625$. We know that the argument of the $Q$ function must scale as $a\sqrt{E_b/N_0}$.

Setting $a\sqrt{E_b/N_0} = a\sqrt{1.625} = 1$, we obtain $a = 1/\sqrt{1.625} = 0.784$. We can therefore write the error probability as

$$P_e = \frac{1}{4}Q\left(0.784\sqrt{E_b/N_0}\right) + \frac{3}{4}Q\left(1.569\sqrt{E_b/N_0}\right)$$

(b) **True.** To find the MPE rule, let us compute the likelihood ratio $L(y) = \frac{p(y|b=+1)}{p(y|b=-1)}$. We have

$$p(y|+1) = P[h=1]p(y|b=+1,h=1)+P[h=2]p(y|b=+1,h=2) = \frac{1}{4}\frac{1}{\sqrt{2\pi}}e^{-(y-1)^2/2}+\frac{3}{4}\frac{1}{\sqrt{2\pi}}e^{-(y-2)^2/2}$$

Similarly,

$$p(y|-1) = P[h=1]p(y|b=-1,h=1)+P[h=2]p(y|b=-1,h=2) = \frac{1}{4}\frac{1}{\sqrt{2\pi}}e^{-(y+1)^2/2}+\frac{3}{4}\frac{1}{\sqrt{2\pi}}e^{-(y+2)^2/2}$$

Upon simplification, we obtain that the likelihood ratio is given by

$$L(y) = \frac{p(y|b=+1)}{p(y|b=-1)} = \frac{e^{y-\frac{1}{2}} + e^{2y-2}}{e^{-y-\frac{1}{2}} + e^{-2y-2}}$$

Clearly, $L(0) = 1$, and $L(y)$ increases as $y$ increases (numerator increases and denominator decreases). Thus, $L(y) > 1$ for $y > 0$ and $L(y) < 1$ for $y < 0$. Thus, $\hat{b}_{MPE} = \text{sign}(y)$.
(c) Conditioned on $b$, $h_1$ and $h_2$, we have $Y_1 + Y_2 \sim N((h_1+h_2)b, 2)$. Given $b = -1$, the conditional error probability, conditioned further on $h_1, h_2$, is $Q(\frac{h_1+h_2}{\sqrt{2}})$. For removing the conditioning on $h_1 + h_2$, note that $P[h_1+h_2 = 4] = (3/4)^2 = 9/16$, $P[h_1+h_2 = 3] = 2(3/4)(1/4) = 3/8$, and $P[h_1+h_2 = 2] = (1/4)^2 = 1/16$. We therefore obtain the error probability, which equals the error probability conditioned on $b = -1$ by symmetry, to be

$$P_e = \frac{9}{16}Q(4/\sqrt{2}) + \frac{3}{8}Q(3/\sqrt{2}) + \frac{1}{16}Q(1/\sqrt{2})$$

(d) For the MPE rule, let us compute the likelihood ratio again. Noting that $y_1$ and $y_2$ are conditionally independent given $b$, we can simply use the results in (b) to infer that

$$L(y_1,y_2) = L(y_1)L(y_2) = \frac{e^{y_1-\frac{1}{2}} + e^{2y_1-2}}{e^{-y_1-\frac{1}{2}} + e^{-2y_1-2}}\frac{e^{y_2-\frac{1}{2}} + e^{2y_2-2}}{e^{-y_2-\frac{1}{2}} + e^{-2y_2-2}}$$

This is certainly not a monotone increasing function of $y_1 + y_2$, hence the decision rule in (c) is not MPE.

**Problem 4.13:** (a) For known $h_i$, the ML rule is given by

$$L(y_1, y_2|H_1, h_1, h_2) \underset{H_0}{\overset{H_1}{\gtrless}} L(y_1, y_2|H_0, h_1, h_2)$$

Conditioned on $H_i$, $y_1$ and $y_2$ are independent, since $n_1$ and $n_2$ are independent, so that

$$L(y_1, y_2|H_i, h_1, h_2) = L(y_1|H_i, h_1, h_2)L(y_2|H_i, h_1, h_2)$$

$$= \exp\left(\tfrac{1}{\sigma^2}\left[\text{Re}\langle y_1, h_1 s_i\rangle - ||h_1 s_i||^2/2\right]\right)\exp\left(\tfrac{1}{\sigma^2}\left[\text{Re}\langle y_1, h_2 s_i\rangle - ||h_2 s_i||^2/2\right]\right)$$

Taking log and throwing away constants irrelevant to the comparison (including the energy term, since $||s_1||^2 = ||s_0||^2$, we have

$$\text{Re}\left(h_1^*\langle y_1, s_1\rangle + h_2^*\langle y_2, s_1\rangle\right) \underset{H_0}{\overset{H_1}{\gtrless}} \text{Re}\left(h_1^*\langle y_1, s_0\rangle + h_2^*\langle y_2, s_0\rangle\right)$$

which can be rewritten as

$$\mathrm{Re}\left(h_1^*\langle y_1, s_1 - s_0\rangle + h_2^*\langle y_2, s_1 - s_0\rangle\right) \begin{array}{c} H_1 \\ > \\ < \\ H_0 \end{array} 0$$

(b) Our signal model is

$$H_i : y = \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} + \begin{pmatrix} h_1 s_i \\ h_2 s_i \end{pmatrix} + \begin{pmatrix} n_1 \\ n_2 \end{pmatrix}$$

is an example of standard coherent communication, with error probability of ML reception given by

$$P_{e|h_1,h_2} = Q\left(\frac{d}{2\sigma}\right)$$

where

$$d^2 = |h_1|^2||s_1 - s_0||^2 + |h_2|^2||s_1 - s_0||^2 = (|h_1|^2 + |h_2|^2)||s_1 - s_0||^2$$

Note that the average received signal energy under $H_i$ is $\mathbb{E}[|h_1|^2 + |h_2|^2]||s_i||^2 = ||s_i||^2 = E_b$ for equal energy signaling. For orthogonal signaling, $||s_1 - s_0||^2 = ||s_1||^2 + ||s_0||^2 = 2E_b$. However, $d^2$ is random, because $h_1, h_2$ are random, and

$$\frac{d}{2\sigma} = \frac{\sqrt{|h_1|^2 + |h_2|^2}\sqrt{2E_b}}{2\sqrt{\frac{N_0}{2}}} = \sqrt{|h_1|^2 + |h_2|^2}\sqrt{\frac{E_b}{N_0}}$$

Proceeding as in Problem 3.4, we can show that $R = \sqrt{|h_1|^2 + |h_2|^2}$ is a Rayleigh random variable with density

$$p(r) = 2re^{-r^2} I_{r \geq 0}$$

The probability of error conditioned on $R = r$ is $Q(rA)$, where $A = \sqrt{\frac{E_b}{N_0}}$. We can now remove the conditioning on $R$:

$$P_e = \int Q(rA)p(r)dr = \int_0^\infty Q(rA)2re^{-r^2}dr$$

Integrating by parts, we obtain

$$P_e = -Q(rA)e^{-r^2}\Big|_0^\infty + \int_0^\infty \left(-\frac{A}{\sqrt{2\pi}}e^{-r^2A^2/2}\right)e^{-r^2}dr = \frac{1}{2} - \frac{A}{\sqrt{2\pi}}\int_0^\infty e^{-r^2(1+A^2/2)}$$

The integrand in the last equation has the form of a $N(0, v^2)$ density with $\frac{1}{2v^2} = 1 + \frac{A^2}{2}$, so that $v^2 = \frac{1}{2+A^2}$. The integral

$$\int_0^\infty e^{-r^2(1+A^2/2)}dr = \int_0^\infty e^{-\frac{r^2}{2v^2}}dr = \frac{1}{2}\sqrt{2\pi v^2} = \sqrt{\frac{\pi}{2(2+A^2)}}$$

Substituting in the expression for the error probability we obtain

$$P_e = \frac{1}{2} - \frac{A}{2\sqrt{2+A^2}} = \frac{1}{2}\left(1 - (1 + \frac{2}{A^2})^{-\frac{1}{2}}\right)$$

That is,

$$P_e = \frac{1}{2}\left(1 - (1 + \frac{2N_0}{E_b})^{-\frac{1}{2}}\right)$$

This evaluates to $P_e = 0.015$ at $\frac{E_b}{N_0}$ of 15 dB.

(c) We know from (a) that $L(y_1, y_2|H_i, h_1, h_2)$ depends on the received signal only through the complex inner products $Z_{ji} = \langle y_j, s_i \rangle$, $j = 1, 2$, $i = 0, 1$. We can therefore obtain the optimal noncoherent decision rule starting from these decision statistics. Conditioned on $H_1$, we have $y_1 = h_1 s_1 + n_1$, $y_2 = h_2 s_1 + n_2$, so that

$$\begin{aligned} Z_{11} &= \langle y_1, s_1 \rangle = h_1 ||s_1||^2 + \langle n_1, s_1 \rangle, Z_{21} = \langle y_2, s_1 \rangle = h_2 ||s_1||^2 + \langle n_2, s_1 \rangle \\ Z_{10} &= \langle y_1, s_0 \rangle = \langle n_1, s_0 \rangle, Z_{20} = \langle y_2, s_0 \rangle = \langle n_2, s_0 \rangle \end{aligned}$$

since $\langle s_1, s_0 \rangle = 0$. Since the channel gains and noises are complex Gaussian, the decision statistics are complex Gaussian (conditioned on $H_1$). By means of covariance computations, we can check that they are independent: this uses the independence of $n_1$, $n_2$, $h_1$ and $h_2$, and the fact that $\langle s_1, s_0 \rangle = 0$. As an example covariance computation, consider

$$\begin{aligned} \text{cov}(Z_{11}, Z_{11}) &= \text{cov}\left(h_1 E_b + \langle n_1, s_1 \rangle, h_1 E_b + \langle n_1, s_1 \rangle\right) \\ &= \text{cov}(h_1, h_1) E_b^2 + \text{cov}(\langle n_1, s_1 \rangle, \langle n_1, s_1 \rangle) + \text{cross terms which equal zero} \\ &= \frac{E_b^2}{2} + 2\sigma^2 ||s_1||^2 = \frac{E_b^2}{2} + 2\sigma^2 E_b \end{aligned}$$

Using similar computations, we arrive at the following hypothesis testing model:

$$\begin{aligned} H_1 :\ & Z_{11} \sim CN(0, 2v_{big}^2), Z_{21} \sim CN(0, 2v_{big}^2) \\ & Z_{10} \sim CN(0, 2v_{small}^2), Z_{20} \sim CN(0, 2v_{small}^2) \end{aligned}$$

$$\begin{aligned} H_0 :\ & Z_{11} \sim CN(0, 2v_{small}^2), Z_{21} \sim CN(0, 2v_{small}^2) \\ & Z_{10} \sim CN(0, 2v_{big}^2), Z_{20} \sim CN(0, 2v_{big}^2) \end{aligned}$$

where $2v_{big}^2 = \frac{E_b^2}{2} + 2\sigma^2 E_b$ and $2v_{small}^2 = 2\sigma^2 E_b$. Let $\mathbf{Z} = (Z_{11}, Z_{21}, Z_{10}, Z_{20})^T$, the ML rule is given by

$$\frac{p(\mathbf{z}|1)}{p(\mathbf{z}|0)} = \frac{p(z_{11}|1)p(z_{21}|1)p(z_{10}|1)p(z_{20}|1)}{p(z_{11}|0)p(z_{21}|0)p(z_{10}|0)p(z_{20}|0)} \underset{H_0}{\overset{H_1}{\gtrless}} 1$$

A typical marginal density in the preceding likelihood ratio is of the form $p(z|i) = \frac{1}{2\pi v^2} e^{-\frac{|z|^2}{2v^2}}$, where $v = v_{big}$ or $v = v_{small}$. Substituting and simplifying upon taking log, we obtain the following rule

$$|Z_{11}|^2 + |Z_{21}|^2 \underset{H_0}{\overset{H_1}{\gtrless}} |Z_{10}|^2 + |Z_{20}|^2$$

That is,

$$|\langle y_1, s_1 \rangle|^2 + |\langle y_2, s_1 \rangle|^2 \underset{H_0}{\overset{H_1}{\gtrless}} |\langle y_1, s_0 \rangle|^2 + |\langle y_2, s_0 \rangle|^2$$

which is nicely intuitive: compute the energies of the projection of the received signal along each signal for each diversity branch, add across branches for each signal, and then choose the largest.
(d) By symmetry, $P_e = P_{e|1}$. Condition on $H_1$. Then $Y_1 = |Z_{11}|^2 + |Z_{21}|^2$ is the sum of two i.i.d. exponential random variables, each with mean $2v_{big}^2 = \frac{1}{\mu_b}$; $Y_0 = |Z_{10}|^2 + |Z_{20}|^2$ is the sum of two i.i.d. exponential random variables, each with mean $2v_{small}^2 = \frac{1}{\mu_s}$; $Y_1$, $Y_0$ are independent.
**Note:** A random variable $Y$ which is a sum of two i.i.d. exponential random variables, each with mean $\frac{1}{\mu}$, has density $p(y) = \mu(\mu y)e^{-\mu y} I_{y \geq 0}$. We now compute the conditional error probability as follows:

$$P_{e|1} = P[Y_0 > Y_1 | H_1] = \int P[Y_0 > y_1 | Y_1 = y_1, H_1] p(y_1|H_1) dy_1 = \int P[Y_0 > y_1 | H_1] p(y_1|H_1) dy_1$$

using the conditional independence of $Y_1$ and $Y_0$. We have

$$P[Y_0 > y_1|H_1] = \int_{y_1}^{\infty} \mu_s y e^{-\mu_s y} \mu_s dy = \int_{\mu_s y_1} t e^{-t} dt$$

substituting $t = \mu_s y$. Integrating by parts, we obtain

$$P[Y_0 > y_1|H_1] = (\mu_s y_1 + 1) e^{-\mu_s y_1}$$

Substituting in the expression for error probability, we have

$$P_{e|1} = \int_0^{\infty} (\mu_s y_1 + 1) e^{-\mu_s y_1} \mu_b^2 y_1 e^{-\mu_b y_1} dy_1$$

Substituting $t = (\mu_s + \mu_b) y_1$, we get

$$P_{e|1} = \left( \frac{\mu_b}{\mu_s + \mu_b} \right)^2 \int_0^{\infty} \left( \frac{\mu_s}{\mu_s + \mu_b} t + 1 \right) t e^{-t} dt$$

Integrating and simplifying, we obtain

$$P_{e|1} = \left( \frac{1}{\frac{\mu_s}{\mu_b} + 1} \right)^2 \left( \frac{2 \frac{\mu_s}{\mu_b}}{\frac{\mu_s}{\mu_b} + 1} + 1 \right)$$

Noting that

$$\frac{\mu_s}{\mu_b} = \frac{2 v_{big}^2}{2 v_{small}^2} = \frac{E_b^2/2 + 2\sigma^2 E_b}{2\sigma^2 E_b} = 1 + \frac{E_b}{4\sigma^2} = 1 + \frac{E_b}{2N_0}$$

we obtain

$$P_e = P_{e|1} = \frac{1}{\left(1 + \frac{E_b}{2N_0}\right)^2} \left( 1 + \frac{2 + \frac{E_b}{N_0}}{1 + \frac{E_b}{N_0}} \right)$$

which decays as $\frac{1}{(E_b/N_0)^2}$ at high SNR.

# Solutions to Chapter 5 Problems

*Fundamentals of Digital Communication*

**Problem 5.1:**

**(a)** Given a bit rate of $2/T$, the symbol rate for QPSK signaling is $2/(T\log_2 4) = 1/T$. The output of the optimal received filter is given by

$$z_{opt}[n] = (y * p_{mf})(nT) = \int y(t)p(t-nT)\,dt$$

where $p(t-nT)$, the impulse response of the cascade of the transmit and channel filters, is given by

$$p(t-nT) = I_{[nT,nT+\frac{T}{2}]} - \frac{3}{2}I_{[nT+\frac{T}{2},nT+T]} + \frac{1}{2}I_{[nT+T,nT+\frac{3}{2}T]}$$

The receive filter output sample at time $kT_s + \tau$ is

$$z[k] = (y * g_{RX})(kT_s+\tau) = \int y(t)g_{RX}(kT_s+\tau-t)\,dt$$

Setting $\tau = 0$ and $T_s = \frac{T}{2}$, we get $g_{RX}(k\frac{T}{2}-t) = I_{[(k-1)\frac{T}{2},k\frac{T}{2}]}$. We can then rewrite $p(t-nT)$ in terms of $g_{RX}(t)$. For example, the first term in the expression for $p(t-nT)$ above is given by $I_{[nT,nT+\frac{T}{2}]} = g_{RX}((2n+1)\frac{T}{2}-t)$.

$$p(t-nT) = g_{RX}\left((2n+1)\frac{T}{2}-t\right) - \frac{3}{2}g_{RX}\left((2n+2)\frac{T}{2}-t\right) + \frac{1}{2}g_{RX}\left((2n+3)\frac{T}{2}-t\right)$$

Applying the above result, we see that $z_{opt}[k]$ can be generated from the samples $z[k]$ as follows

$$z_{opt}[k] = z[2n+1] - \frac{3}{2}z[2n+2] + \frac{1}{2}z[2n+3]$$

**(b)** The sampled autocorrelation sequence is given by

$$h[n] = \int p(t)p(t-nT)\,dt$$

The finite memory condition is satisfied with $L = 1$:

$$h[n] = 0, \quad |n| > 1$$

The number of required states, given QPSK signaling, is $M^L = 4^1 = 4$ states.

**Problem 5.2:**

**(a)** The bandwidth efficiency is given by $\eta_B = \log_2 8$ bits/symbol $= 3$ bits/symbol. From Chapter 2, the expression for minimum bandwidth is given by

$$B_{min} = \frac{6\times 10^6 \text{ bits/second}}{3 \text{ bits/symbol}} = 2 \text{ MHz}$$

**(b)** The impulse response of the cascade of the transmit and channel filters is given by

$$\begin{aligned} p(t) &= (g_{TX} * g_C)(t) \\ &= g_{TX}(t-0.5T) - \frac{1}{2}g_{TX}(t-1.5T) + \frac{1}{4}g_{TX}(t-2.5T) \end{aligned}$$

We can compute the sampled autocorrelation sequence of the pulse $p(t)$, given by $h[m] = \int p(t)p^*(t-mT)dt$, and find that $h[0] = 0.907$, $h[1] = -0.289$, $h[2] = 0.067$, $h[3] = 0.050$, and so on without $h[n]$ reaching zero. We can treat $h[m] = 0$ once it becomes sufficiently small. Alternatively, we can sample the input at $t = 0.5T + kT$. In this case, the sampled response to the symbol $b[0]$is given by $(..., 0, 1, -\frac{1}{2}, \frac{1}{4}, 0, ...)$ and the required memory is $L = 2$. The number of states in the trellis is $M^L = 8^2 = 64$.

**Problem 5.3:**
**(a)** We express the pairwise error probability as

$$\begin{aligned}
P_e &= \mathrm{Q}\left(\frac{||\mathrm{s}(\mathbf{b}-2\mathbf{e})-\mathrm{s}(\mathbf{b})||}{2\sigma}\right) \\
&= \mathrm{Q}\left(\frac{||\mathrm{s}(\mathbf{e}||}{\sigma}\right) \\
&= \mathrm{Q}\left(\sqrt{\frac{\sum_n \mathrm{h}[0]\mathrm{e}^2[\mathrm{n}] + 2\mathrm{e}[\mathrm{n}]\mathrm{h}[1]\mathrm{e}[\mathrm{n}-1]}{\sigma^2}}\right) \\
&= \mathrm{Q}\left(\sqrt{\frac{(\mathrm{h}[0]\mathrm{e}^2[1]) + (\mathrm{h}[0]\mathrm{e}^2[2] + 2\mathrm{e}[2]\mathrm{e}[1]\mathrm{h}[1])}{\sigma^2}}\right) \\
&= \mathrm{Q}\left(\sqrt{\frac{(2\mathrm{h}[0] + 2\mathrm{e}[2]\mathrm{e}[1]\mathrm{h}[1])}{\sigma^2}}\right)
\end{aligned}$$

The above expression is maximized when $e[1] = -e[2]$, resulting in

$$P_{e,max} = \mathrm{Q}\left(\sqrt{\frac{(2(\mathrm{h}[0]-\mathrm{h}[1])}{\sigma^2}}\right) = \mathrm{Q}\left(\sqrt{4\frac{\mathrm{E_b}}{\mathrm{N_0}}\left(1-\frac{\mathrm{h}[1]}{\mathrm{h}[0]}\right)}\right) = \mathrm{Q}\left(\sqrt{2.1\frac{\mathrm{E_b}}{\mathrm{N_0}}}\right)$$

**(b)** This problem is identical in form to the transfer function bound example provided in the text. We calculate $a_0 = \exp\left(-\frac{h[0]}{2\sigma^2}\right)$, $a_1 = \exp\left(-\frac{h[0]+2h[1]}{2\sigma^2}\right)$, and $a_2 = \exp\left(-\frac{h[0]-2h[1]}{2\sigma^2}\right)$. The bound is given by

$$P_e \leq \frac{\frac{1}{2}\exp\left(-\frac{h[0]}{2\sigma^2}\right)}{\left[1-\frac{1}{2}\left(\exp\left(-\frac{h[0]+2h[1]}{2\sigma^2}\right)+\exp\left(-\frac{h[0]-2h[1]}{2\sigma^2}\right)\right)\right]^2}$$

To express this in terms of $\frac{E_b}{N_0}$, we use the relations $N_0 = 2\sigma^2$ and $E_b = h[0]\mathbb{E}[|b[n]|^2] = h[0]$. The probability of error can then be written

$$\begin{aligned}
P_e &\leq \frac{\frac{1}{2}\exp\left(-\frac{E_b}{N_0}\right)}{\left[1-\frac{1}{2}\left(\exp\left(-\frac{E_b}{N_0}\left(1+\frac{2h[1]}{h[0]}\right)\right)+\exp\left(-\frac{E_b}{N_0}\left(1-\frac{2h[1]}{h[0]}\right)\right)\right)\right]^2} \\
&= \frac{\frac{1}{2}\exp\left(-\frac{E_b}{N_0}\right)}{\left[1-\frac{1}{2}\left(\exp\left(-\frac{E_b}{N_0}1.6\right)+\exp\left(-\frac{E_b}{N_0}0.4\right)\right)\right]^2}
\end{aligned}$$

The transfer function bound is plotted in Fig. 1.

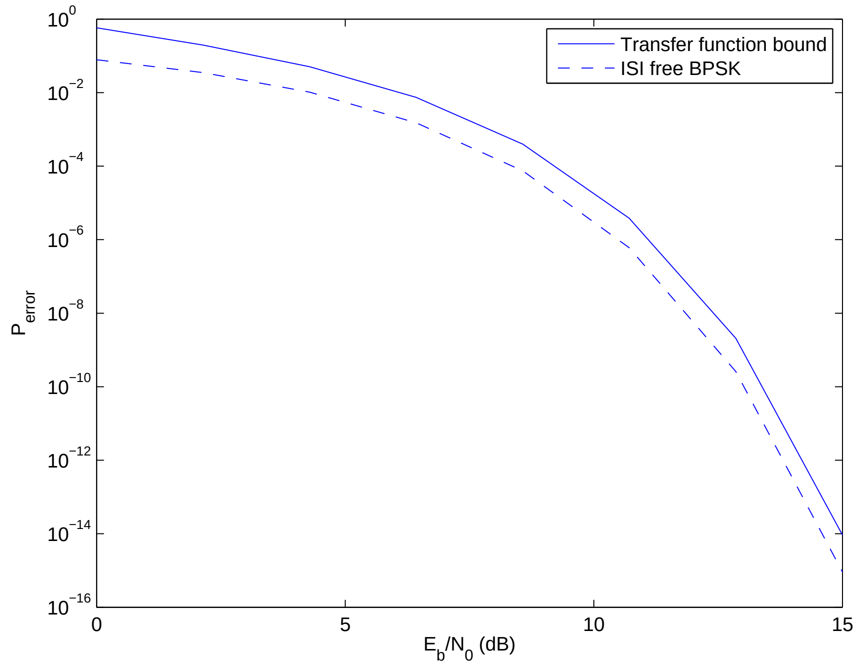



Figure 1: MLSE transfer function bound plotted vs $E_b/N_0$.

**Problem 5.4:**
**(a)** No, we know that $p(t)$ is not a Nyquist pulse because $P(f)$ does not satisfy the Nyquist condition. Specifically,

$$\frac{1}{T}\sum_{k=-\infty}^{\infty} P(f+\frac{k}{T}) \neq 1.$$

**(b)** We first derive an expression for the sampled output of the matched filter $p_{mf}(t) = p^*(-t) = p(-t)$. We will then recreate this expression using only the samples $\{r[l]\}$ at the output of the non-matched filter. The output of the matched filter is given by:

$$\begin{aligned} z_{mf}[n] &= (y * p_{mf})(nT) \\ &= \int y(t)p(t-nT)\,dt \end{aligned}$$

$P(f)$ can be written as

$$\begin{aligned} P(f) &= \cos(\pi f T)\mathrm{rect}(Tf) \\ &= \frac{1}{2}(e^{j\pi f T} + e^{-j\pi f T})\mathrm{rect}(Tf) \end{aligned}$$

Taking the inverse Fourier transform of $P(f)$, we obtain $p(t)$

$$p(t) = \frac{1}{2T}\mathrm{sinc}\left(\frac{t+\frac{T}{2}}{T}\right) + \frac{1}{2T}\mathrm{sinc}\left(\frac{t-\frac{T}{2}}{T}\right)$$

Therefore, the sampled output of the matched filter is given by

$$z_{mf}[n] = \frac{1}{2T}\int y(t)\mathrm{sinc}\left(\frac{t}{T} - \left(n-\frac{1}{2}\right)\right)\,dt + \frac{1}{2T}\int y(t)\mathrm{sinc}\left(\frac{t}{T} - \left(n+\frac{1}{2}\right)\right)\,dt$$

Instead of this, however, we have samples at the output of our receive filter $g_{RX}(t)$, given by

$$\begin{aligned} r[l] &= r(lT_s - \tau) \\ &= (y * g_{RX})(lT_s - \tau) \\ &= \frac{1}{T}\int y(t)\mathrm{sinc}\left(\frac{lT_s - \tau - t}{T}\right) dt \end{aligned}$$

where we have obtained $g_{RX}(t)$ by taking the inverse Fourier transform of $G_{RX}(f) = I_{[-\frac{1}{2T},\frac{1}{2T}]}$. Setting $T_s = T$ and $\tau = 1/2$, we obtain

$$r[l] = \frac{1}{T}\int y(t)\mathrm{sinc}\left(\frac{l - \frac{1}{2} - t}{T}\right) dt = \frac{1}{T}\int y(t)\mathrm{sinc}\left(\frac{t}{T} - (l - \frac{1}{2})\right) dt$$

where we used the fact that sinc$(t)$ is an even function to reverse its time index. By summing samples $r[l]/2$ and $r[l+1]/2$, we obtain

$$\begin{aligned} \frac{1}{2}(r[l] + r[l+1]) &= \frac{1}{T}\int y(t)\mathrm{sinc}\left(\frac{t}{T} - (l - \frac{1}{2})\right) dt + \frac{1}{T}\int y(t)\mathrm{sinc}\left(\frac{t}{T} - (l + 1 - \frac{1}{2})\right) dt \\ &= \frac{1}{2T}\int y(t)\mathrm{sinc}\left(\frac{t}{T} - (l - \frac{1}{2})\right) dt + \frac{1}{2T}\int y(t)\mathrm{sinc}\left(\frac{t}{T} - (l + \frac{1}{2})\right) dt \\ &= z_{mf}[l] \end{aligned}$$

We have obtained the matched filter output using only the samples $\{r[l]\}$. Note that it is possible to obtain $z_{mf}[l]$ when other values of $T_s$ and $\tau$ are chosen. For instance, we could instead take $T_s = T/2$ and $\tau = 0$ resulting in $z_{mf}[n] = \frac{1}{2}r[2n-1] + \frac{1}{2}r[2n+1]$.

**(c)** The *maximum weight* branch metric is given by

$$\lambda_n(b[n], s[n]) = b[n]z[n] - \frac{h[0]}{2}|b[n]|^2 - b[n]\sum_{m=n-L}^{n-1} b[m]h[n-m]$$

We first determine the sampled autocorrelation sequence

$$h[n] = \int (p * p_{mf})(mT) = \int p(t)p(t - nT)\, dt$$

$$\begin{aligned} h[0] &= \int p(t)p(t)\, dt \\ &= \int |P(f)|^2\, df \quad \text{(by Parseval's theorem)} \\ &= \int_{-\frac{1}{2T}}^{\frac{1}{2T}} \cos^2(\pi f T)\, df = \frac{1}{2T} \end{aligned}$$

Similarly,

$$\begin{aligned}
h[1] &= \int p(t)p(t-T)\,dt \\
&= \int p(t)p^*(t-T)\,dt \\
&= \int P(f)\left(P(f)e^{-j2\pi fT}\right)^* \; df \\
&= \int P^2(f)e^{j2\pi fT} \; df \\
&= \frac{1}{2}\int_{-\frac{1}{2T}}^{\frac{1}{2T}} (1+\cos(2\pi fT))e^{j2\pi fT} \; df \\
&= \frac{1}{2}\int_{-\frac{1}{2T}}^{\frac{1}{2T}} [e^{j2\pi fT} + \frac{1}{2}(e^{j4\pi fT}+1)] \; df \\
&= \frac{1}{4T}
\end{aligned}$$

For $n \geq 2$, $h[n] = 0$. Combining these results, we see that the (weight maximizing) branch metric from state $s[n]$ to state $s[n+1]$ is given by

$$\begin{aligned}
\lambda_n(b[n], s[n]) &= b[n]z_{mf}[n] - \frac{h[0]}{2}|b[n]|^2 - h[1]b[n]b[n-1] \\
&= \frac{1}{2}b[n](r[n]+r[n+1]) - \frac{1}{4T} - \frac{1}{4T}b[n]b[n-1]
\end{aligned}$$

The $\frac{1}{4T}$ term can be removed as it is a constant. If we assume that the Viterbi algorithm searches for a minimum weight path through the trellis, we can modify the above branch metric as follows

$$m_n(b[n], s[n]) = b[n]\left\{b[n-1] - 2T(r[n]+r[n+1])\right\}$$

Suppose now that we know that $b[0] = +1$. The resulting trellis is shown in Fig. 2.

**(d)** For $L = 1$

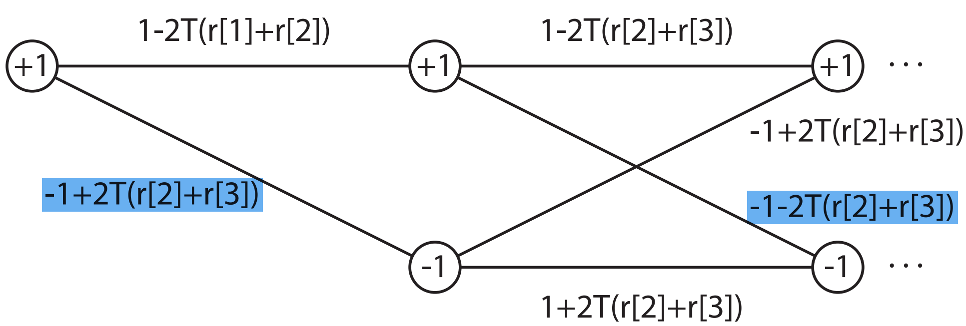



Figure 2: Example trellis for Problem 5.4.

$$||s(e)||^2_{min} = h[0] + (w-1)(h[0] - 2|h[1]|)$$

when $w = 1$, $\epsilon^2_{min} = h[0]$. Thus the asymptotic efficiency is given by

$$\eta = \frac{h[0]}{h[0]} = 1$$

**(e)** Consider a single sample, given by $y[k] = h[0]b[k] + h[1]b[k-1] + h[-1]b[k+1] + n[k]$, where $n[k] \sim N(0, \sigma^2)$. The asymptotic efficiency of one-shot detection based on this sample is given by:

$$\eta = \frac{h[0] - 2|h[1]|}{h[0]} = \frac{\frac{1}{2T} - \frac{2}{4T}}{\frac{1}{2T}} = 0$$

**(f)** Denote the $k$-th sample by $y_k = h_0 b_k + h_1 b_{k-1} + h_{-1} b_{k+1} + n_k$. To calculate the probability of error for one shot detection, we must calculate the variance of the noise component of the sum $z_{mf}[k] = \frac{1}{2}(r[k] + r[k+1])$, which we will denote by $v_z^2$. First we find $v_r^2$, variance of the noise component of $r[k]$, given by

$$v_r^2 = \sigma^2 \int |g_{RX}(t)|^2 \, dt = \sigma^2 \int |G_{RX}(f)|^2 \, df = \sigma^2 \frac{1}{T}$$

We then obtain

$$v_z^2 = \frac{1}{4}(v_r^2 + v_r^2) = \sigma^2 \frac{1}{2T}$$

Next, we condition on the interference bits to find the probability of error. Setting $b[k] = +1$, the probability of error for one-shot detection is

$$\begin{aligned}
P_{e,o.s.} &= P(y_k < 0) \\
&= P(y_k < 0 | b_{k+1} = -b_{k-1}) P(b_{k+1} = -b_{k-1}) \\
&\quad + P(y_k < 0 | b_{k+1} = b_{k-1} = 1) P(b_{k+1} = b_{k-1} = 1) \\
&\quad + P(y_k < 0 | b_{k+1} = b_{k-1} = -1) P(b_{k+1} = b_{k-1} = -1) \\
&= \frac{1}{2} Q\left(\frac{h_0}{v_z}\right) + \frac{1}{4} Q\left(\frac{h_0 + 2|h_1|}{v_z}\right) + \frac{1}{4} Q\left(\frac{h_0 - 2|h_1|}{v_z}\right)
\end{aligned}$$

Noting that $|h_1| = \frac{h_0}{2}$ and $E_b = h_0 = \frac{1}{2T}$, we can write

$$\begin{aligned}
P_{e,o.s.} &= \frac{1}{2} Q\left(\frac{h_0}{v_z}\right) + \frac{1}{4} Q\left(\frac{2h_0}{v_z}\right) + \frac{1}{8} \\
&= \frac{1}{2} Q\left(\frac{1/(2T)}{\sigma/\sqrt{2T}}\right) + \frac{1}{4} Q\left(\frac{1/T}{\sigma/\sqrt{2T}}\right) + \frac{1}{8} \\
&= \frac{1}{2} Q\left(\sqrt{\frac{E_b}{N_0}}\right) + \frac{1}{4} Q\left(\sqrt{\frac{2E_b}{N_0}}\right) + \frac{1}{8} \\
&= 0.1254
\end{aligned}$$

We next compute the transfer function bound on the error probability of the MLSE solution.

Observing that $N_0 = 2\sigma^2$, $E_b = h[0]\mathbb{E}[|b[n]|^2] = h[0]$ and $\frac{h[1]}{h[0]} = \frac{1/(4T)}{1/(2T)} = \frac{1}{2}$, we obtain

$$\begin{aligned}
P_e &\leq \frac{\frac{1}{2}a_0}{[1-\frac{1}{2}(a_1+a_2)]^2} \\
&= \frac{\frac{1}{2}\exp\left(-\frac{h[0]}{2\sigma^2}\right)}{\left[1-\frac{1}{2}\left(\exp\left(-\frac{h[0]+2h[1]}{2\sigma^2}\right)+\exp\left(-\frac{h[0]-2h[1]}{2\sigma^2}\right)\right)\right]^2} \\
&= \frac{\frac{1}{2}\exp\left(-\frac{E_b}{N_0}\right)}{\left[1-\frac{1}{2}\left(\exp\left(-\frac{E_b}{N_0}\left(1+\frac{2h[1]}{h[0]}\right)\right)+\exp\left(-\frac{E_b}{N_0}\left(1-\frac{2h[1]}{h[0]}\right)\right)\right)\right]^2} \\
&= \frac{\frac{1}{2}\exp\left(-\frac{E_b}{N_0}\right)}{\left[1-\frac{1}{2}\left(\exp\left(-2\frac{E_b}{N_0}\right)+1\right)\right]^2} \\
&= 9.08 \times 10^{-5}
\end{aligned}$$

The ISI-free error probability is given by

$$P_e = Q\left(\sqrt{\frac{2E_b}{N_0}}\right) = 3.872 \times 10^{-6}$$

**Problem 5.5:**

**(a)** From the discussion of proper complex Gaussian random processes provided in Chapter 4, we know that $N[k]$ is proper complex Gaussian since it is the result of a linear transformation on a proper complex Gaussian random process. The mean of $N[k]$ is given

$$\begin{aligned}
\mathbb{E}[N[k]] &= \mathbb{E}\left[\int n(t)g(kT_s - t)\,dt\right] \\
&= \int \mathbb{E}[n(t)]g(kT_s - t)\,dt \\
&= 0
\end{aligned}$$

The autocorrelation function is obtained as follows

$$\begin{aligned}
R_N[l] &= \mathbb{E}[N[k]N^*[k-l]] \\
&= \mathbb{E}\left[\int n(\tau)g(kT_s-\tau)\,d\tau \int n^*(u)g^*((k-l)T_s-u)\,du\right] \\
&= \int\int \mathbb{E}[n(\tau)n^*(u)]\,g(kT_s-\tau)g^*((k-l)T_s-u)\,d\tau\,du \\
&= \int\int 2\sigma^2\delta(\tau-u)g(kT_s-\tau)g^*((k-l)T_s-u)\,d\tau\,du \\
&= 2\sigma^2\int g(kT_s-u)g^*((k-l)T_s-u)\,du \\
&= 2\sigma^2\int g(t)g^*(t-lT_s)\,dt \\
&= 2\sigma^2 r_g[l]
\end{aligned}$$

**(b)** We first show that $\{r_g[l]\}$ is conjugate symmetric.

$$
\begin{aligned}
r_g^*[-l] &= \left(\int g(t)g^*(t+lT_s)\,dt\right)^* \\
&= \int g^*(t)g(t+lT_s)\,dt \\
&= \int g^*(u-lT_s)g(u)\,du \\
&= r_g[l]
\end{aligned}
$$

where the substitution $u = t + lT_s$ was used in the third equality. The conjugate symmetry of $R_N[l]$ follows directly: $R_N^*[-l] = 2\sigma^2 r_g^*[-l] = 2\sigma^2 r_g[l] = R_N[l]$.
**(c)** We show that $S_N(z) = S_N^*(1/z^*)$ as follows

$$
\begin{aligned}
S_N\left(\frac{1}{z^*}\right) &= \left(\sum_l R_N[l]\left(\frac{1}{z^*}\right)^{-l}\right)^* \\
&= \sum_l R_N^*[l]z^l \\
&= \sum_m R_N^*[-m]z^{-m} \\
&= \sum_m R_N[m]z^{-m} \\
&= S_N(z)
\end{aligned}
$$

where we used the substitution $l = -m$ in the third line and the conjugate symmetry of $R_N[l]$ in the fourth.
**(d)** If $z_0$ is a root of $S_N$, then $S_N(z_0) = 0$. From **(c)** it follows that $S_N(1/z_0^*) = (S_N(z_0))^* = 0$.
**(e)** From **(d)**, we know that if $S_N(z)$ has a finite number of roots inside the unit circle, given by $K$, it also has a $K$ roots outside the unit circle given by $\{1/a_k^*\}$. For each root $\{a_k\}$ falling inside the unit circle, we can extract a linear factor $(1 - a_k z^{-1})$, resulting in

$$
S_N(z) = G(z)\prod_{k=1}^{K}(1 - a_k z^{-1})
$$

where $G(z)$ is a polynomial of degree $K$. Similarly, we can extract the linear factors corresponding to the roots $\{1/a_k^*\}$ outside the unit circle to obtain

$$
S_N(z) = A\prod_{k=-1}^{K}(1 - a_k z^{-1})(1 - a_k^* z)
$$

where $A$ must be a constant because all roots have been factored.
**(f)** If we pass discrete-time WGN denoted by $W[k]$ through a causal filter with z-transform $H(z) = \sqrt{A}\prod_k(1 - a_k z^{-1})$, the PSD of the output $N[k]$ is given by

$$
\begin{aligned}
S_N(z) &= H(z)H^*((z^*)^-1)S_W(z) \\
&= A\prod_k(1 - a_k z^{-1})(1 - a_k^* z)
\end{aligned}
$$

which is the desired form for the PSD of our colored noise process.

**(g)** If we pass $N[k]$ through a causal filter with z-transform $H(z) = 1/\left(\sqrt{A}\prod_k(1-a_k^* z)\right)$, the PSD of the output $W[k]$ is given by

$$
\begin{aligned}
S_W(z) &= \frac{S_N(z)}{H(z)H^*((z^*)^-1)} \\
&= \frac{A\prod_k(1-a_k z^{-1})(1-a_k^* z)}{A\prod_k(1-a_k z^{-1})(1-a_k^* z)} \\
&= 1
\end{aligned}
$$

which represents WGN.

**Problem 5.6:**

**(a)** The signal contribution to $z[k]$ can be written

$$
\begin{aligned}
z_s[k] &= \int \sum_n b[n]p(t-nT)p_{MF}(kT-t)\,dt \\
&= \sum_n b[n]\int p(t-nT)p_{MF}(kT-t)\,dt \\
&= \sum_n b[n]\int p(\tau)p_{MF}((k-n)T-\tau)d\tau \\
&= \sum_n b[n](p * p_{MF})((k-n)T) \\
&= \sum_n b[n]h[k-n] = (b*h)[k]
\end{aligned}
$$

**(b)** A linear transformation on complex WGN results in a proper complex Gaussian random process as described in Chapter 4. Wide-sense stationarity is preserved as described in Appendix A. The mean is given by

$$
\mathbb{E}[z_n[k]] = \mathbb{E}\left[\int n(t)p_{MF}(kT-t)dt\right] = \int \mathbb{E}[n(t)]p_{MF}(kT-t)dt = 0
$$

and the covariance is given by

$$
\begin{aligned}
\mathbb{E}[z_n[l]z_n^*[l-k]] &= \mathbb{E}\left[\int n(t)p_{MF}(lT-t)dt \int n^*(u)p_{MF}^*((l-k)T-u)du\right] \\
&= \int\int \mathbb{E}[n(t)n^*(u)]p_{MF}(lT-t)p_{MF}^*((l-k)T-u)dt\;du \\
&= \int\int 2\sigma^2\delta(t-u)p_{MF}(lT-t)p_{MF}^*((l-k)T-u)dt\;du \\
&= 2\sigma^2\int p_{MF}(lT-t)p(t-(l-k)T)dt \\
&= 2\sigma^2\int p(\tau)p_{MF}(kT-\tau)dt \\
&= 2\sigma^2 h[k]
\end{aligned}
$$

where, in the second to last line, we used the substitution $\tau = t-(l-k)T$. The derivation for real valued signals is the same with the exception $\mathbb{E}[n(t)n(u)] = \sigma^2\delta(t-u)$ for real-valued noise processes.

**(c)** The signal component is given by

$$\begin{aligned} z_s[k] &= \sum_n b[n]h[k-n] \\ &= b[k-1]h[-1] + b[k]h[0] + b[k+1]h[-1] \\ &= \frac{3}{2}b[k] - \frac{1}{2}(b[k-1] + b[k+1]) \end{aligned}$$

The PSD of the noise component is given by

$$\begin{aligned} S_{z_n}(z) &= \sum_k C_{z_n}[k]z^{-k} \\ &= \sigma^2 \sum_k h[k]z^{-k} \\ &= \sigma^2(h[0] + h[1]z^{-1}h[-1]z) \\ &= \sigma^2 \left( \frac{3}{2} - \frac{1}{2}(z + z^{-1}) \right) \end{aligned}$$

**(d)** We wish to factorize $S_{z_n}(z)$ such that it takes the form $(a + bz)(a^* + b^* z^{-1})$. First find the roots of $S_{z_n}(z)$, using the quadratic equation to solve

$$-\frac{1}{2}z^2 + \frac{3}{2}z - \frac{1}{2} = 0$$

We obtain the root $x_0 = \frac{1}{2}(3 - \sqrt{5})$, chosen from the two available roots because $|x_0| < 1$. Solving for $A$ below

$$\sigma^2(-\frac{1}{2}z + \frac{3}{2} - \frac{1}{2}z^{-1}) = A(1 - x_0 z)(1 - x_0 z^{-1})$$

we find that $A = \sigma^2/(3 - \sqrt{5})$. To get $S_{z_n}(z)$ into its desired form, we write

$$\begin{aligned} A(1 - x_0 z)(1 - x_0 z^{-1}) &= (\sqrt{A} - \sqrt{A}x_0 z)(\sqrt{A} - \sqrt{A}x_0 z^{-1}) \\ &= (a + bz)(a + bz^{-1}) \end{aligned}$$

where $a = \frac{\sigma}{\sqrt{3-\sqrt{5}}}$ and $b = -\frac{\sigma}{2}\sqrt{3 - \sqrt{5}}$. The factor $(1 - x_0 z^{-1})$ is causal and specifies our filter coefficients. Specifically, we obtain

$$g[0] = \frac{\sigma}{\sqrt{3 - \sqrt{5}}}$$

$$g[1] = -\frac{\sigma}{2}\sqrt{3 - \sqrt{5}}$$

**(e)** A plot of the estimated MLSE performance and transfer function bound is shown in Fig. 3. The MLSE performance was estimated by generating signal and noise samples as described above and running them algorithm.

**Problem 5.7:**

**(a)** We wish to factorize $S_{z_n}(z)$ such that it takes the form $A(1 + az^{-1})(1 + az)$. First find the roots of $S_{z_n}(z)$, using the quadratic equation to solve

$$-\frac{1}{2}z^2 + \frac{3}{2}z - \frac{1}{2} = 0$$

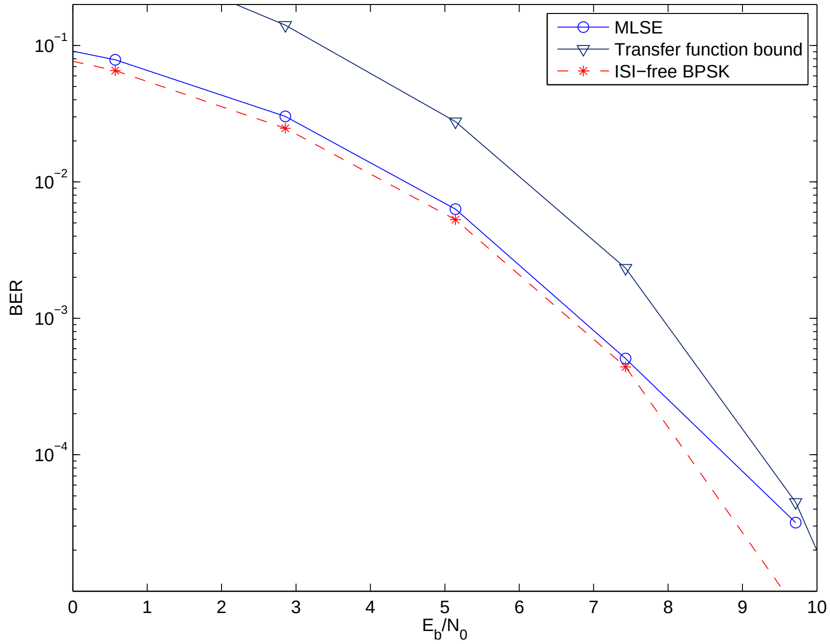



Figure 3: Simulated MLSE performance and transfer function bound as a function of $E_b/N_0(dB)$.

We find the root $a = \frac{1}{2}(3 - \sqrt{5}) < 1$. Solving for $A$ below

$$-\frac{1}{2}z + \frac{3}{2} - \frac{1}{2}z^{-1} = A(1 - az^{-1})(1 - az)$$

we find that $A = 1/(3 - \sqrt{5})$.

**(b)** When $\{z_n[k]\}$ is passed through filter $Q$, the output $\{w[k]\}$ has power spectral density (see Appendix A) given by

$$\begin{aligned} S_W(z) &= Q(Z)Q^*((z^*)^{-1})S_{z_n}(z) \\ &= \frac{A(1+az^{-1})(1+az)}{A(1+az^{-1})(1+az)} \\ &= 1 \end{aligned}$$

indicating that $\{w[k]\}$ is WGN.

**(c)** Observe that $z_s[k] = (b * h)[k]$ and the corresponding $z$-transform is given by $Z_s(z) = B(z)H(z)$, with $H(z) = (\frac{3}{2} - \frac{1}{2}(z + z^{-1}))$ by inspection. As shown in **(a)**, we can express $H(z)$ as $H(z) = A(1 + az^{-1})(1 + az)$. Passing $\{z_s[k]\}$ through $Q$ yields $\{y_s[k]\}$ with $z$-transform

$$Y_s(z) = B(z)H(z)Q(z) = B(z)\frac{A(1+az^{-1})(1+az)}{\sqrt{A}(1+az^{-1})} = B(z)\sqrt{A}(1+az)$$

In the discrete time domain, we obtain $z_s[k] = (b * h_1)[k]$ with $H_1(z) = \sqrt{A}(1 + az)$.

**(d)** From above, we have $y_s[k] = (b * h_1)[k] = \sqrt{A}(b[k] + ab[k-1])$. Summing this with the result in **(b)**, we obtain $y[k] = y_s[k] + w[k] = \sqrt{A}(b[k] + ab[k-1]) + w[k]$.

**Problem 5.8:**
**(a)** The sampled output of the receive filter is now given by

$$r[k] = (y * g_{RX})(k + 1/2) = \int_{k-1/2}^{k+1/2} y(t)dt$$

The sampled response to $b[0]$ is now $(..., 0, \frac{1}{2}, \frac{3}{4}, 0, -\frac{1}{2}, 0, ...)$. The matrix $\mathbf{U}$ is written as

$$\mathbf{U} = \begin{pmatrix} 0 & \frac{1}{2} & 0 \\ -\frac{1}{4} & \frac{3}{4} & 0 \\ 0 & 0 & \frac{1}{2} \\ 0 & -\frac{1}{4} & \frac{3}{4} \\ 0 & 0 & 0 \end{pmatrix}$$

The zero-forcing coefficients are given by

$$\mathbf{c}_{ZF} = \mathbf{U}(\mathbf{U}^H\mathbf{U})^{-1}\mathbf{e} = (1.8571, 0, 0.4286, -0.2857, 0)^T$$

The real part of the output of the ZF equalizer is given by

$$Z[n] = b[n] + N[n]$$

where $N[n] \sim N(0, v^2)$ and $v^2 = \sigma^2||\mathbf{c}_{ZF}||^2 = \frac{N_0}{2}||\mathbf{c}_{ZF}||^2$. The error probability is given by $Q(1/v)$. To find the noise enhancement, we must solve for $a$ in the following equation

$$P_e = Q\left(\sqrt{\frac{E_b}{||\mathbf{c}_{ZF}||^2\frac{N_0}{2}}\frac{1}{E_b}}\right) = Q\left(\sqrt{a\frac{E_b}{N_0}}\right)$$

This results in

$$a = \frac{2}{||\mathbf{c}_{ZF}||^2}\frac{1}{E_b} = \frac{4}{3||\mathbf{c}_{ZF}||^2} = 0.359$$

Finally, we calculate the noise enhancement as $10\log_{10}\frac{2}{0.359} = 7.4594$ dB. Contrast this with noise enhancement of 2.73 dB when the sampling times are not shifted by $\frac{1}{2}$.

**Problem 5.9:**
**(a)**

$$\begin{aligned} MSE &= \mathbb{E}[(\mathbf{c}^T\mathbf{r}_n - b_n)^2] \\ &= \mathbf{c}^T\mathbb{E}[\mathbf{r}_n\mathbf{r}_n^T]\mathbf{c} - 2\mathbf{c}^T\mathbb{E}[b_n\mathbf{r}_n] + \mathbb{E}[b_n^2] \\ &= \mathbf{c}^T\mathbf{R}\mathbf{c} - 2\mathbf{c}^T\mathbf{p} + 1 \end{aligned}$$

since $\mathbf{R}\mathbf{c}_{mmse}^T = \mathbf{p}$,

$$\begin{aligned} MMSE &= \mathbf{c}_{mmse}^T\mathbf{p} - 2\mathbf{c}_{mmse}^T\mathbf{p} \\ &= 1 - \mathbf{c}_{mmse}^T\mathbf{p} \\ &= 1 - \mathbf{p}^T\mathbf{R}^{-1}\mathbf{p} \end{aligned}$$

**(b)** The MSE for a linear correlator $\mathbf{c}$ is given by

$$\begin{aligned} J(\mathbf{c}) &= \mathbb{E}[|\langle\mathbf{c}, \mathbf{r}_n\rangle - b_n|^2] \\ &= |(\langle\mathbf{c}, \mathbf{u}_0\rangle - 1)|^2 + \sum_{j\neq 0}|\langle\mathbf{c}, \mathbf{u}_j\rangle|^2 + \mathbf{c}^T\mathbf{C}_w\mathbf{c} \\ &= |\langle\mathbf{c}, \mathbf{u}_0\rangle - 1|^2 + \mathbf{c}^T\mathbf{A}\mathbf{c} \\ SIR &= \frac{|\langle\mathbf{c}, \mathbf{u}_0\rangle|^2}{\mathbf{c}^T\mathbf{A}\mathbf{c}} \end{aligned}$$

where $\mathbf{A} = \sum_{j\neq 0} \mathbf{u}_0\mathbf{u}_0^T + \mathbf{C}_w$. Let $\alpha = \langle \mathbf{c}_{mmse}, \mathbf{u}_0 \rangle$, then $\mathbf{c}^T\mathbf{A}\mathbf{c} = \frac{\alpha}{SIR_{mmse}}$.
**(c)** Taking the derivative of the MSE, we obtain

$$\frac{d}{d\mathbf{c}}\left( |\langle \mathbf{c}, \mathbf{u}_0\rangle - 1|^2 + \sum_{j\neq 0} |\langle \mathbf{c}, \mathbf{u}_j\rangle|^2 + \mathbf{c}^T\mathbf{C}_w\mathbf{c} \right) = 2(\mathbf{c}^T\mathbf{u}_0 - 1)\mathbf{u}_0 + 2\sum_{j\neq 0} \mathbf{u}_j\mathbf{u}_j^T + 2\mathbf{C}_w\mathbf{c}$$

Setting equal to zero, we get $\mathbf{A}\mathbf{c}_{mmse} = (1 - \mathbf{c}_{mmse}^T\mathbf{u}_0)\mathbf{u}_0$, where $\mathbf{A}$ is defined above. It follows that $\mathbf{c}_{mmse}^T\mathbf{A}\mathbf{c}_{mmse} = \mathbf{c}_{mmse}^T\mathbf{u}_0(1 - \mathbf{c}_{mmse}^T\mathbf{u}_0)$. Using this result and noting that $MMSE = 1 - \mathbf{c}_{mmse}^T\sigma_b^2\mathbf{u}_0 = 1 - \mathbf{c}_{mmse}^T\mathbf{u}_0$, we obtain

$$SIR_{max} = \frac{(\mathbf{c}_{mmse}^T\mathbf{u}_0)^2}{\mathbf{c}_{mmse}^T\mathbf{u}_0(1 - \mathbf{c}_{mmse}^T\mathbf{u}_0)} = \frac{\mathbf{c}_{mmse}^T\mathbf{u}_0}{1 - \mathbf{c}_{mmse}^T\mathbf{u}_0} = \frac{1 - MMSE}{MMSE} = \frac{1}{MMSE} - 1$$

**(d)** The MSE for the zero-forcing solution is given by

$$MSE(\mathbf{c}_{ZF}) = \mathbf{c}_{ZF}^T\mathbf{C}_w\mathbf{c}_{ZF} = \sigma^2||\mathbf{c}_{ZF}||^2$$

As $\sigma^2 \to 0$, $MSE(\mathbf{c}_{ZF})$ goes to zero. Since $MSE(\mathbf{c}_{ZF}) \geq MSE(\mathbf{c}_{mmse})$, this implies

$$\lim_{\sigma^2\to 0} MSE(\mathbf{c}_{mmse}) = 0$$

The MSE is given by

$$MSE(\mathbf{c}) = |\langle \mathbf{c}, \mathbf{u}_0\rangle - 1|^2 + \sum_{j\neq 0} |\langle \mathbf{c}, \mathbf{u}_j\rangle|^2 + \sigma^2||\mathbf{c}||^2$$

Thus, for $\lim_{\sigma^2\to 0} MSE(\mathbf{c}_{mmse}) = 0$, we must have $\langle \mathbf{c}_{mmse}, \mathbf{u}_0\rangle \to 1$ and $\langle \mathbf{c}_{mmse}, \mathbf{u}_j\rangle \to 0$ for $j \neq 0$ as $\sigma_2 \to 0$.
**(e)** We write the cost function as

$$J = \mathbf{c}^T\mathbf{u}_0 + \lambda(\mathbf{c}^T\mathbf{R}\mathbf{c} - 1)$$

so that

$$\nabla_{\mathbf{c}} J = \mathbf{u}_0 + \lambda(2\mathbf{R}\mathbf{c}_p) = 0 \Rightarrow \mathbf{u}_0 = -\lambda 2\mathbf{R}\mathbf{c}_p$$

Solving for $\mathbf{c}_p$ we get

$$\mathbf{c}_p = -\frac{1}{2\lambda}R^{-1}\mathbf{u}_0 = kR^{-1}\mathbf{u} \propto R^{-1}\mathbf{p}$$

**Problem 5.10:**
**(a)** Matrix $\mathbf{U}$ is given by

$$\mathbf{U} = \begin{pmatrix}
\frac{1}{4} & \frac{1}{2} & -\frac{1+j}{2} & 0 & 0 \\
\frac{1+2j}{4} & 0 & \frac{1-j}{4} & 0 & 0 \\
\frac{3-j}{2} & \frac{-j}{4} & 1+2j & 0 & 0 \\
0 & \frac{1}{4} & \frac{1}{2} & -\frac{1+j}{2} & 0 \\
0 & \frac{1+2j}{4} & 0 & \frac{1-j}{4} & 0 \\
0 & \frac{3-j}{2} & \frac{-j}{4} & 1+2j & 0 \\
0 & 0 & \frac{1}{4} & \frac{1}{2} & -\frac{1+j}{2} \\
0 & 0 & \frac{1+2j}{4} & 0 & \frac{1-j}{4} \\
0 & 0 & \frac{3-j}{2} & \frac{-j}{4} & 1+2j
\end{pmatrix}$$

The correlator $\mathbf{c}_{ZF}$ is given by

$$\begin{aligned}\mathbf{c}_{ZF} &= \mathbf{U}(\mathbf{U}^H\mathbf{U})^{-1}\mathbf{e} \\ &= \begin{pmatrix} -0.154-0.192j \\ 0.234-0.189j \\ 0.087+0.114j \\ 0.183+0.034j \\ -0.049+0.049j \\ 0.034+0.008j \\ 0.161-0.072j \\ 0.116+0.204j \\ 0.092-0.013j \end{pmatrix}\end{aligned}$$

The output of the equalizer is given by

$$Z[n] = \mathbf{c}_{ZF}^H\mathbf{r}[n] = s[n] + N[n]$$

where $N[n] \sim CN(0, 2v^2)$ and $2v^2 = ||\mathbf{c}_{ZF}||^2 N_0$. For BPSK signaling, the error probability (as given in the text) is $Q(\frac{1}{v})$. To find the noise enhancement, we first solve for $a$ in the following equation

$$\frac{1}{v} = \sqrt{\frac{aE_b}{N_0}}$$

where $E_b = ||\mathbf{p}||^2 = 8.8125$ and $\mathbf{p}$ is the discrete time end-to-end impulse reponse. The above equation can be written

$$\sqrt{\frac{2}{N_0||\mathbf{c}_{ZF}||^2}} = \sqrt{\frac{a||p||^2}{N_0}}$$

and solving for $a$ yields $a = \frac{2}{||\mathbf{c}_{ZF}||^2||p||^2}$. The noise enhancement is given by

$$10\log_{10}(\frac{2}{a}) = 10\log_{10}(||\mathbf{c}_{ZF}||^2||p||^2) = 4.36 \text{ dB}$$

**(b)** For complex-valued equalizers, as in this case, the coefficients of the filters $\{G_i(z)\}$ are time-reversed, subsampled versions of $\mathbf{c}_{ZF}^*$. The parallel filters are

$$\begin{aligned}G_1(z) &= (0.161+0.072j) + (0.183-0.034j)z^{-1} + (-0.154+0.192j)z^{-2} \\ G_2(z) &= (0.116-0.204j) + (-0.049-0.049j)z^{-1} + (0.234+0.189j)z^{-2} \\ G_3(z) &= (0.092+0.013j) + (0.034-0.008j)z^{-1} + (0.087-0.114j)z^{-2}\end{aligned}$$

and $\sum_{i=1}^{3} H_i(z)G_i(z) = z^{-2}$.
**(c)** The symbol error probability for ISI-free 16-QAM is estimated as

$$P_{e,sym} \approx 3Q\left(\sqrt{\frac{4}{5}\frac{E_b}{N_0}}\right)$$

The received symbol energy is given by $E_s = ||\mathbf{p}||^2 = 8.8125$. At the output of the ZF equalizer, however, the energy contributed per symbol is equal to $\mathbb{E}[||s[n]||]^2$ (which we assume is unity for simplicity, although this assumption does not affect the final result). The energy per bit is $\mathbb{E}[||s[n]||]^2/\log_2 M = 1/4$, and the noise component at the output has variance given by

$2v^2 = N_0||\mathbf{c}_{ZF}||^2 = 0.3094N_0$. The estimated probability of symbol error at the output of the receiver is given by

$$P_{e,sym} \approx 3Q\left(\sqrt{\frac{4}{5}\frac{1}{||\mathbf{p}||^2||\mathbf{c}_{ZF}||^2}\frac{E_b}{N_0}}\right) = 3.475 \times 10^{-4}$$

The probability of bit error with Gray coding can be approximated by

$$P_{e,bit} = \frac{1}{log_2 M}P_{e,sym} = \frac{1}{4}P_{e,sym} = 8.69 \times 10^{-5}$$

**(d)** The noise enhancement in dB is plotted in Fig. 4.

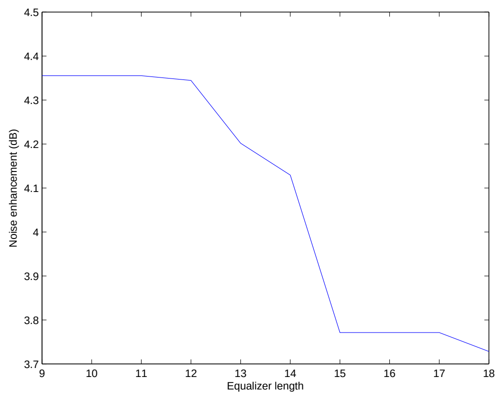



Figure 4: The noise enhancement as a function of ZF equalizer length.

**Problem 5.11:**
**(a)** The received symbol energy is given by $E_s = \sigma_b^2||\mathbf{p}||^2$, the energy per bit is given by $E_b = E_s/\log_2 M = E_s/4$, and $N_0 = \frac{E_b}{10^{15/10}} = 0.0697$. To find the MMSE equalizer coefficients, we compute

$$\mathbf{R} = \mathbf{U}\mathbf{U}^H + N_0\mathbf{I}$$

and

$$\mathbf{p} = \mathbf{U}(:,3)$$

where $\mathbf{U}$ is specified in the Problem 5.10 solution. The MMSE correlator is given by

$$\begin{aligned}\mathbf{c}_{mmse} &= \mathbf{R}^{-1}\mathbf{p} \\ &= \begin{pmatrix} -0.145 - 0.183j \\ 0.222 - 0.175j \\ 0.0875 + 0.120j \\ 0.171 + 0.030j \\ -0.042 + 0.043j \\ 0.033 + 0.005j \\ 0.152 - 0.068j \\ 0.110 + 0.192j \\ 0.092 - 0.014j \end{pmatrix}\end{aligned}$$

**(b)** To generate the histograms displayed in Fig. 5, we first generated 16-QAM symbols denoted by $\{s[n]\}$. From these symbols, noiseless received vector samples were generated as follows

$$\mathbf{r}_s[n] = s[n]\mathbf{u}_0 + \sum_{i \neq 0} s[n+i]\mathbf{u}_i$$

These samples were passed through the MMSE correlator to obtain $Z[n]$, the correlator output. The residual ISI (plus a neglible term $b[n](\mathbf{c}^H\mathbf{u}_0)$) can be obtained by taking

$$\mathrm{x}_{\mathrm{ISI}}[n] = Z[n] - s[n]$$

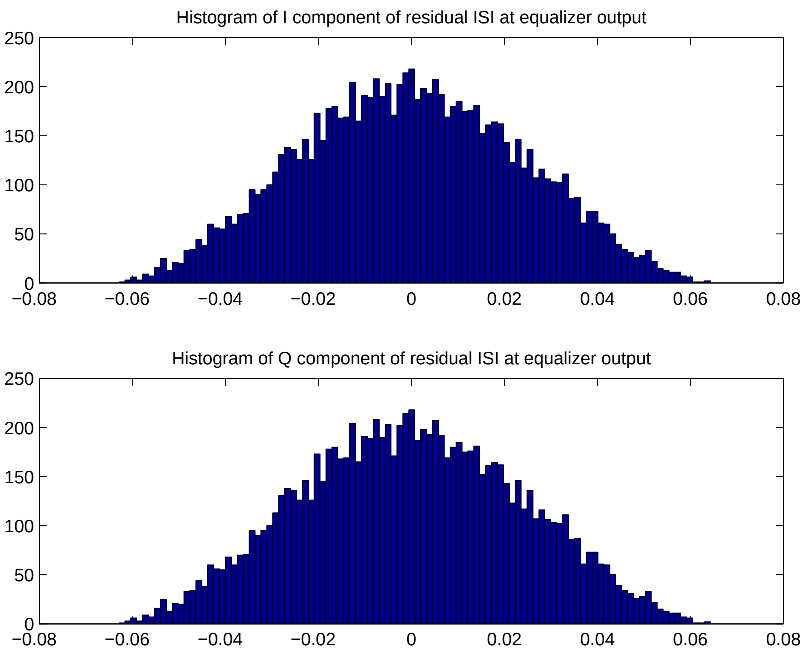



Figure 5: Histograms of the residual ISI at the output of the MMSE equalizer.

**(c)** We treat the residual ISI at time $n$ as a zero mean Gaussian random variable with variance $v_I^2$. $v_I^2$ can be calculated estimated according to the method provided in the book, $v_I^2 = \sum_{i \neq 0} A_i^2 = 6.87 \times 10^{-4}$, where $A_i = \langle \mathbf{c}, \mathbf{u}_i \rangle$, or it can be estimated directly from the samples generated in **(b)** to obtain $v_I^2 = 1.09 \times 10^(-3)$. The samples at the equalizer output are approximated as

$$Z[n] = A_0 s[n] + W_I[n] + W[n]$$

where $W_I[n] \sim N(0, v_I^2)$ is the Gaussian approximation of the ISI and $W[n] \sim N(0, 2v^2)$ is Gaussian noise with variance $2v^2 = N_0||\mathbf{c}||^2$. Note that the bit energy at the output is given by $E_b = E_s/4 = A_0^2/4$. The estimated error probability for 16-QAM can thus be written as

$$\begin{aligned} P_{e,sym} &\approx 3Q\left(\sqrt{\frac{4}{5}\frac{A_0^2}{4(v_I^2 + 2v^2)}}\right) \\ &= 2.89 \times 10^{-4} \end{aligned}$$

**(d)** The normalized inner product at $E_b/N_0 = 15dB$ is given by

$$\frac{\langle \mathbf{c}_{mmse}, \mathbf{c}_{ZF} \rangle}{||\mathbf{c}_{mmse}||^2 ||\mathbf{c}_{ZF}||^2} = 3.417$$

At 5 dB and 25 dB, the correlation is 4.625 and 3.252, respectively.

**Problem 5.12:**

**(a)** Two of the five symbols contributing to the observation interval are "past" symbols. Two feedback taps are required to cancel out their effect.

**(b)** To find the coefficients of the feedforward and feedback taps for a ZF-DFE, we first find $\mathbf{U}_f$ and $\mathbf{U}_p$, the matrices containing $\{\mathbf{u}_j, j \geq 0\}$ and $\{\mathbf{u}_j, j < 0\}$ respectively. These are given by

$$\mathbf{U}_f = \begin{pmatrix} -\frac{1+j}{2} & 0 & 0 \\ \frac{1-j}{4} & 0 & 0 \\ 1+2j & 0 & 0 \\ \frac{1}{2} & -\frac{1+j}{2} & 0 \\ 0 & \frac{1-j}{4} & 0 \\ \frac{-j}{4} & 1+2j & 0 \\ \frac{1}{4} & \frac{1}{2} & -\frac{1+j}{2} \\ \frac{1+2j}{4} & 0 & \frac{1-j}{4} \\ \frac{3-j}{2} & \frac{-j}{4} & 1+2j \end{pmatrix}$$

and

$$\mathbf{U}_p = \begin{pmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ -\frac{1+j}{2} & 0 \\ \frac{1-j}{4} & 0 \\ 1+2j & 0 \\ \frac{1}{2} & -\frac{1+j}{2} \\ 0 & \frac{1-j}{4} \\ \frac{-j}{4} & 1+2j \end{pmatrix}$$

The ZF-DFE equalizer is specified by

$$\begin{aligned} c_{FF} &= \mathbf{U}_f(\mathbf{U}_f^H \mathbf{U}_f)^{-1}\mathbf{e} \\ &= \begin{pmatrix} -0.072 - 0.072j \\ 0.036 - 0.036j \\ 0.144 + 0.288j \\ 0.067 - 0.006j \\ 0.003 - 0.003j \\ 0.009 - 0.012j \\ 0.086 - 0.037j \\ 0.055 + 0.094j \\ 0.046 - 0.006j \end{pmatrix} \end{aligned}$$

and

$$\begin{aligned} c_{FB} &= -\mathbf{U}_p^H \mathbf{c}_{FF} \\ &= \begin{pmatrix} -0.045 - 0.459j \\ 0.072 + 0.0176j \end{pmatrix} \end{aligned}$$

**(c)** The received symbol energy is given by $E_s = \sigma_b^2||\mathbf{p}||^2$, the energy per bit is given by $E_b = E_s/\log_2 M = E_s/4$, and $N_0 = \frac{E_b}{10^{15/10}} = 0.0697$ Using the $\mathbf{U}_f$ and $\mathbf{U}_p$ defined above, we calculate the MMSE-DFE coefficients as follows

$$\begin{aligned} c_{FF} &= (\mathbf{U}_f\mathbf{U}_f^H + N_0 I)^{-1}\mathbf{u}_0 \\ &= \begin{pmatrix} -0.071 - 0.071j \\ 0.036 - 0.036j \\ 0.142 + 0.284j \\ 0.066 - 0.006j \\ 0.003 - 0.003j \\ 0.009 - 0.012j \\ 0.084 - 0.036j \\ 0.054 + 0.093j \\ 0.048 - 0.007j \end{pmatrix} \end{aligned}$$

and

$$\begin{aligned} c_{FB} &= -\mathbf{U}_p^H \mathbf{c}_{FF} \\ &= \begin{pmatrix} -0.044 - 0.453j \\ 0.071 + 0.0178j \end{pmatrix} \end{aligned}$$

**(d)** Error probability calculations are identical except $c_{FF}$ has replaced $c$. Figures 6 and 7 show an approximately 3.25 dB improvement in both cases.

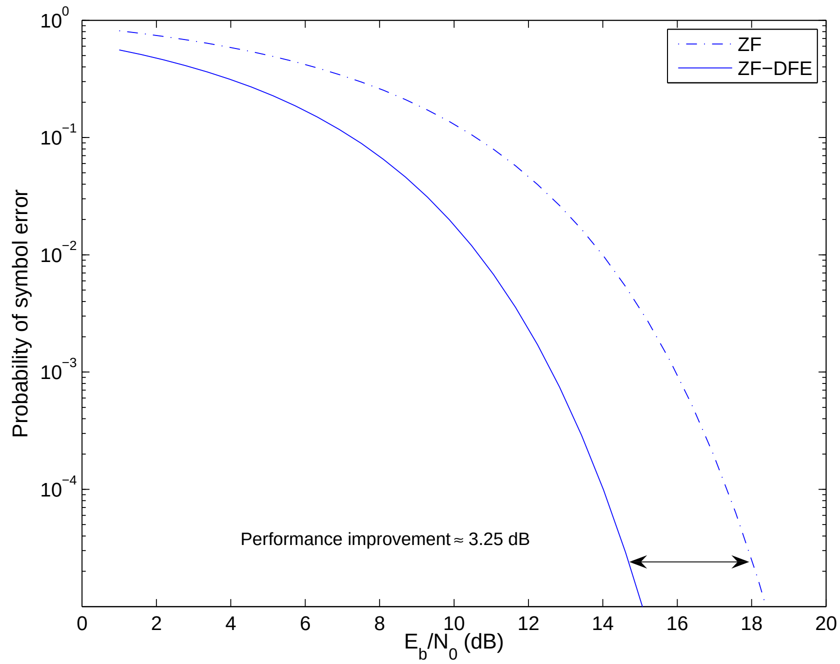



Figure 6: Performance improvement of ZF-DFE over ZF equalizer.

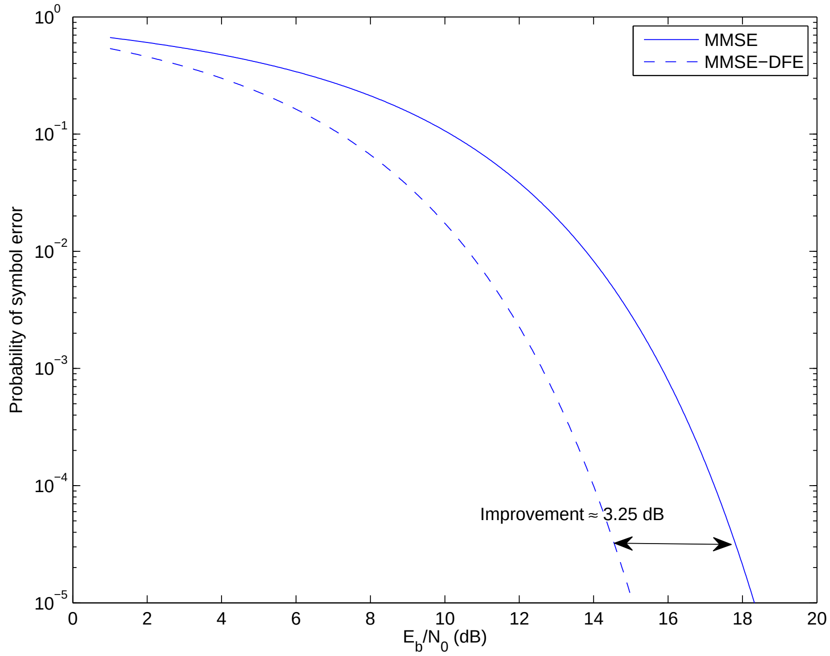



Figure 7: Performance improvement of MMSE-DFE over MMSE equalizer.

**Problem 5.13:**
**(a)** There are three parallel symbol-spaced subchannels. The impulse response of the first subchannel is $(\ldots, 0, -\frac{1+j}{2}, \frac{1}{2}, \frac{1}{4}, 0, \ldots)$, of the second is $(..., 0, \frac{1-j}{4}, 0, \frac{1+2j}{4}, 0, \ldots)$, and of the third is $(..., 0, 1+2j, -\frac{j}{4}, \frac{3-j}{2}, 0, \ldots)$. Denoting the impulse response of the $i$-th channel by $g_i[n]$, we can compute

$$h[m] = \sum g_i[n] g_i^*[n-m]$$

and find that $h[m] > 0$ for $|m| \leq 2$, thus we must keep track of the past $L = 2$ symbols. The required number of states is given by $M^L = 4^2 = 16$.
**(b)** The received samples $\{r_i[n]\}$ for the $i$th subchannel are given by

$$r_i[n] = \ s_{\mathbf{b},i}]n] + w[n] = \sum_{m=k-L}^{k+L} b[m] g_i[k-m] + w[n]$$

where $w[n]$ is Gaussian noise. The received samples from all three subchannels can be written in vector form as

$$\mathbf{r}[n] = \mathbf{s_b}[n] + \mathbf{w}[n] = \sum_{m=k-L}^{k+L} b[m] \mathbf{g}[k-m] + \mathbf{w}[n]$$

The ML sequence is given by

$$\arg\min_{\mathbf{b}} \sum_n ||\mathbf{r}[n] - \mathbf{s_b}[n]||^2 = \arg\max_{\mathbf{b}} \sum_n \langle \mathbf{r}[n], \mathbf{s_b}[n] \rangle - \frac{||\mathbf{s_b}[n]||^2}{2}$$

The first term on the right hand side can be expressed as

$$\langle \mathbf{r}[n], \mathbf{s_b}[n] \rangle = \sum_{k=n-L}^{n+L} b[k] \mathbf{g}^H[n-k] \mathbf{r}[n]$$

The second term can be written

$$\begin{aligned}
\frac{||\mathbf{s_b}[n]||^2}{2} &= \langle \sum_{m=n-L}^{n+L} b[m]g[n-m], \sum_{k=n-L}^{n+L} b[k]g[n-k]\rangle \\
&= \sum_{k=n-L}^{n+L}\sum_{m=n-L}^{n+L} b^*[k]b[m]\mathbf{g}^H[n-k]\mathbf{g}[n-m] \\
&= \sum_{k=n-L}^{n+L} |b[k]|^2||\mathbf{g}[n-k]||^2 + \sum_{k=n-L}^{n+L}\sum_{m<k} b^*[k]b[m]\mathbf{g}^H[n-k]\mathbf{g}[n-m] \\
&\quad + \sum_{k=n-L}^{n+L}\sum_{m>k} b^*[k]b[m]\mathbf{g}^H[n-k]\mathbf{g}[n-m] \\
&= \sum_{k=n-L}^{n+L} |b[k]|^2||\mathbf{g}[n-k]||^2 + \sum_{k=n-L}^{n+L}\sum_{m<k} 2\mathrm{Re}\{b^*[k]b[m]\mathbf{g}^H[n-k]\mathbf{g}[n-m]\}
\end{aligned}$$

where we switched the roles of $k$ and $m$ to obtain the expression in the last line. Combining these expressions, the additive branch metric $\lambda(b[n], s[n])$ can be expressed as

$$\begin{aligned}
\lambda_n &= \sum_{k=n-L}^{k=n+L} (b[k]\mathbf{g}^H[n-k]\mathbf{r}[n] - \frac{1}{2}|b[k]|^2||\mathbf{g}[n-k]||^2 \\
&\quad - \mathrm{Re}\{b^*[n]\sum_{m=n-L}^{k-1} b[m]\mathbf{g}^H[n-k]\mathbf{g}[n-m]\}) \\
&= \sum_{k=n-L}^{k=n+L} \mathbf{g}^H[n-k]\left(b[k]\mathbf{r}[n] - \frac{1}{2}|b[k]|^2\mathbf{g}[n-k] - \mathrm{Re}\{b^*[n]\sum_{m=n-L}^{k-1} b[m]\mathbf{g}[n-m]\}\right)
\end{aligned}$$

**Problem 5.14:**
**(a)** The estimated BER is plotted for the linear MMSE and MMSE-DFE in Fig. 8.
**(b)** ISI causes approximately 3 dB of degradation when using the MMSE equalizer, and 2 dB degradation when using the MMSE-DFE equalizer. The noise enhancement caused by the ZF equalizers is higher, but comparable.

**Problem 5.15: Computer simulation**

**Problem 5.16:**
**(a)** The probability is given by

$$\begin{aligned}
P(\tilde{b}_n \neq b_n) &= P(y[n] > 0 | b[n] = -1) \\
&= P(0.1b[n-1] - 0.05b[n-2] - 0.1b[n+1] - 0.05b[n+2] + w[n] > 1) \\
&= P(w[n] > 1 - .1b[n-1] + 0.05b[n-2] + 0.1b[n+1] + 0.05b[n+2])]) \\
&= \mathbb{E}\left[Q\left(\frac{1 - .1b[n-1] + 0.05b[n-2] + 0.1b[n+1] + 0.05b[n+2]}{\sigma}\right)\right] \\
&= Q\left(\frac{0.7}{\sigma}\right) \\
&\approx \exp\left(\frac{-0.7^2}{2\sigma^2}\right) \text{ when } \sigma^2 \to 0
\end{aligned}$$

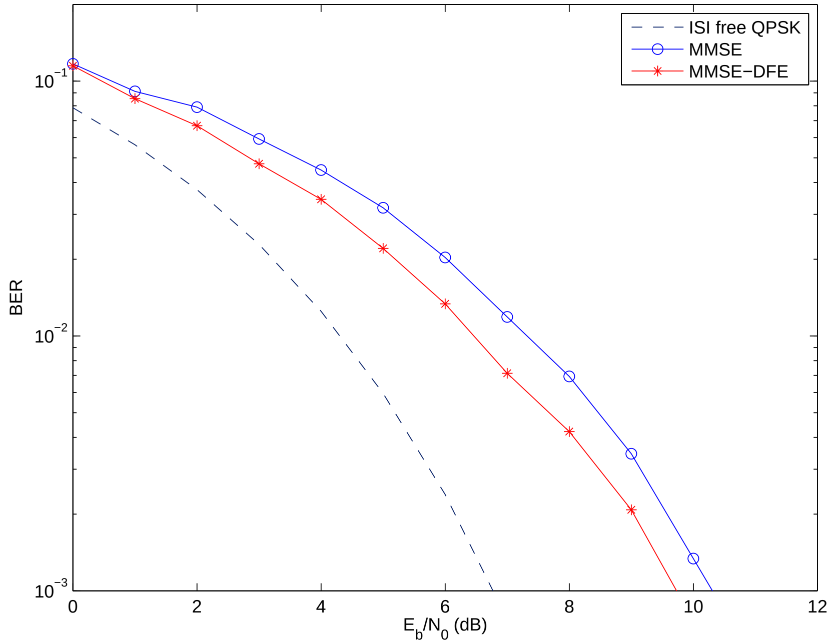



Figure 8: Estimated probability of error for MMSE and MMSE-DFE equalizers.

Therefore, $\lim_{\sigma^2 \to 0} \sigma^2 \log P(\tilde{b}_n \neq b_n) = -0.7^2/2 = -0.254$
**(b)** Let $y[n] = b[n] + I[n]$, where $I[n] \sim N(0, v_I^2)$. We can estimate $v_I^2$ as follows

$$I[n] = 0.1b[n-1] - 0.05b[n-2] - 0.1b[n+1] - 0.05b[n+2] + w[n]$$

$$v_I^2 = (0.1)^2 + (0.05)^2 + (0.1)^2 + (0.05)^2 + \sigma^2 = 0.185$$

The SIR is given by $SIR = \frac{1}{0.185}$ and the probability of error estimate is given by

$$P_e \approx Q(\sqrt{SIR}) = Q(\sqrt{\frac{1}{0.184}}) = 0.0099$$

**(c)** To find the exact error probability, we consider all possible bit combinations. We define $f(b[n-1], b[n-2], b[n+1], b[n+2]) = 0.1b[n-1] - 0.1b[n-1] - 0.05(b[n+2] + b[n-2]) + 1$.

Running through all possible bit combinations, we obtain

$$
\begin{aligned}
f(1,1,1,1) &= 1.1 \\
f(1,1,1,-1) &= 1 \\
f(1,1,-1,1) &= 0.9 \\
f(1,-1,1,1) &= 1 \\
f(-1,1,1,1) &= 1.3 \\
f(1,1,-1,-1) &= 0.8 \\
f(1,-1,1,-1) &= 0.9 \\
f(-1,1,1,-1) &= 1.2 \\
f(1,-1,-1,1) &= 0.8 \\
f(-1,1,-1,1) &= 1.1 \\
f(-1,-1,1,1) &= 1.2 \\
f(-1,-1,-1,1) &= 1 \\
f(-1,-1,1,-1) &= 1.1 \\
f(-1,1,-1,-1) &= 1 \\
f(1,-1,-1,-1) &= 0.7 \\
f(-1,-1,-1,-1) &= 0.9
\end{aligned}
$$

These values are then used to calculate the exact probability of error, where we have conditioned on the various bit combinations.

$$
\begin{aligned}
P_e &= \frac{1}{16}\left(Q(\frac{1.3}{\sigma}) + 2Q(\frac{1.2}{\sigma}) + 3Q(\frac{1.1}{\sigma}) + 4Q(\frac{1}{\sigma}) + 3Q(\frac{0.9}{\sigma}) + 2Q(\frac{0.8}{\sigma}) + Q(\frac{0.7}{\sigma}) +\right) \\
&\approx 0.01
\end{aligned}
$$

**Problem 5.17:**
**(d)** and **(e)** The BER for the MMSE-DFE and MLSE is plotted in Fig. **??**.

**Problem 5.18:**
**(a)**

$$\mathbf{A}^{-1}(\mathbf{A} + \mathbf{x}\mathbf{x}^H)\mathbf{z} = \mathbf{A}^{-1}\mathbf{y}$$

$$(\mathbf{I} + \mathbf{A}^{-1}\mathbf{x}\mathbf{x}^H)\mathbf{z} = \mathbf{A}^{-1}\mathbf{y}$$

$$\mathbf{z} + \mathbf{A}^{-1}\mathbf{x}\mathbf{x}^H\mathbf{z} = \mathbf{A}^{-1}\mathbf{y}$$

$$\mathbf{z} = \mathbf{A}^{-1}\mathbf{y} - \mathbf{A}^{-1}\mathbf{x}\mathbf{x}^H\mathbf{z}$$

**(b)**

$$\mathbf{x}^H\mathbf{z} = \mathbf{x}^H\mathbf{A}^{-1}\mathbf{y} - \mathbf{x}^H\mathbf{A}^{-1}\mathbf{x}\mathbf{x}^H\mathbf{z}$$

$$(1 + \mathbf{x}^H\mathbf{A}^{-1}\mathbf{x})\mathbf{x}^H\mathbf{z} = \mathbf{x}^H\mathbf{A}^{-1}\mathbf{y}$$

$$\mathbf{x}^H\mathbf{z} = \frac{\mathbf{x}^H\mathbf{A}^{-1}\mathbf{y}}{1 + \mathbf{x}^H\mathbf{A}^{-1}\mathbf{x}}$$

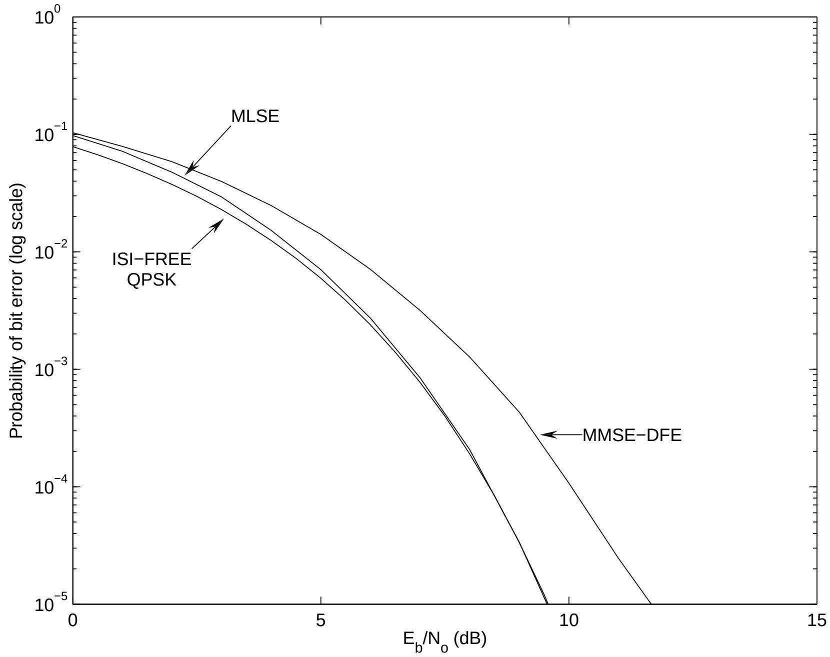



Figure 9: Simulated MLSE and MMSE-DFE performance with ISI-free QPSK BER plotted for reference.

**(c)**

$$
\begin{aligned}
\mathbf{z} &= \mathbf{A}^{-1}\mathbf{y} - \mathbf{A}^{-1}\mathbf{x}\frac{\mathbf{x}^H\mathbf{A}^{-1}\mathbf{y}}{1+\mathbf{x}^H\mathbf{A}^{-1}\mathbf{x}} \\
&= \left(\mathbf{A}^{-1} - \frac{\mathbf{A}^{-1}\mathbf{x}\mathbf{x}^H\mathbf{A}^{-1}}{1+\mathbf{x}^H\mathbf{A}^{-1}\mathbf{x}}\right)\mathbf{y} \\
&= \left(\mathbf{A}^{-1} - \frac{\tilde{\mathbf{x}}\tilde{\mathbf{x}}^H}{1+\mathbf{x}^H\tilde{\mathbf{x}}}\right)\mathbf{y}
\end{aligned}
$$

# Solutions to Chapter 6 Problems

*Fundamentals of Digital Communication*

## Problem 6.1

The total bandwidth available ($B$) is given to be 1.5 GHz. Since this scheme utilizes 50% excess bandwidth, we have

$$B_{min} = \frac{B}{1+a} = 1\text{GHz} \tag{1}$$

We know that the symbol rate $R_s = B_{min} = 1$ GHz and the required bit rate is 1 Gbps. Therefore, the spectral efficiency is 1 bit/channel use. From the Shannon Capacity Formula for AWGN channels, we have

$$\left(\frac{E_b}{N_0}\right)_{min} = \frac{2^1 - 1}{1} = 1$$

Also, since spectral efficiency is 1, we have $SNR_{reqd} = \frac{E_b}{N_0} = 1$. We now calculate the noise power at the receiver. We have

$$P_n = kT_0B10^{\frac{F}{10}} = (1.38 \times 10^{-23}) \times (290) \times (1.5 \times 10^9) \times 3 \text{ W} \approx 1.8 \times 10^{-11} \text{ W}$$

For the required $SNR$ at the receiver, we must have $P_{rx} = 1.8 \times 10^{-11}$ W. We also know that

$$P_{tx} = P_{rx} \frac{1}{G_{tx}G_{rx}} \frac{16\pi^2 R^2}{\lambda^2}$$

where $P_{tx}$ is the transmit power, $G_{tx}$ and $G_{rx}$ are the transmit and receive antenna gains, $R$ is the range and $\lambda$ is the wavelength of the carrier used. We have $G_{tx} = 10, G_{rx} = 4$ and $r = 1000$m. Further, we are given that the carrier frequency $f_c = 28$GHz. Therefore, we have

$$\lambda = \frac{3 \times 10^8 \text{m/s}}{f_c} = \frac{3}{280} = 1.07 \times 10^{-2}\text{m}$$

Substituting the values, we get

$$\begin{aligned} P_{tx} &= 1.8 \times 10^{-11} \times \frac{1}{40} \times \frac{16\pi^2 10^6}{1.07^2 \times 10^{-4}}\text{W} \\ &= 0.62\text{W} \end{aligned}$$

## Problem 6.2

### Part a

Let $\mathcal{X}$ be the alphabet of $X$ with $|\mathcal{X}| = M$. We are also given that,

$$p(x) = \frac{1}{M} \qquad \forall\ x \in \mathcal{X}$$

The entropy $H(X)$ is given by,

$$\begin{aligned}
H(X) &= \sum_{x\epsilon\mathcal{X}} p(x)\log_2\frac{1}{p(x)} \\
&= \sum_{x\epsilon\mathcal{X}} \frac{1}{M}\log_2 M \\
&= M \times \frac{1}{M}\log_2 M = \log_2 M
\end{aligned}$$

### Part b

$$\begin{aligned}
D(p||q) &= \sum_{x\epsilon\mathcal{X}} p(x)\log_2\frac{p(x)}{q(x)} \\
&= \sum_{x\epsilon\mathcal{X}} p(x)\log_2 q(x) - p(x)\log_2\frac{1}{p(x)} = \left(\sum_{x\epsilon\mathcal{X}} p(x)\log_2 M\right) - H_p(X) \\
&= \log_2 M - H_p(X)
\end{aligned}$$

### Part c

From the non-negativity of divergence, we have

$$D(p||q) = \log_2 M - H_p(X) \geq 0 \Rightarrow H_p(X) \leq \log_2 M$$

Therefore, the uniform distribution maximizes the entropy of a discrete random variable.

## Problem 6.3

### Part a

$$\begin{aligned}
D(p||q) &= \int p(x)\log_2\frac{p(x)}{q(x)}dx \\
&= \int p(x)\log_2\left(\sqrt{2\pi v^2}e^{x^2/2v^2}\right)dx - \int p(x)\log_2\frac{1}{p(x)}dx \\
&= \int p(x)\frac{1}{2}\log_2\left(2\pi v^2\right)dx + \log_2 e\int p(x)\frac{x^2}{2v^2}dx - h(X) \\
&= \frac{1}{2}\log_2\left(2\pi v^2\right) + \frac{1}{2}\log_2 e - h(X) \\
&= \frac{1}{2}\log_2\left(2\pi e v^2\right) - h(X) = h(N(0,v^2)) - h(X)
\end{aligned}$$

## Part b

Once again, from the non-negativity of divergence, we have

$$D(p||q) = h(N(0,v^2)) - h(X) \geq 0 \Rightarrow h(X) \leq h(N(0,v^2))$$

Therefore, the Gaussian distribution maximizes the differential entropy for a given variance.

## Problem 6.4

### Part a

Let $f(\mathbf{x})$ denote the pdf of $\mathbf{X}$ and it is given by

$$f(\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n|\mathbf{C}|}}\exp\left(-\frac{(\mathbf{x}-\mathbf{m})^T\mathbf{C}^{-1}(\mathbf{x}-\mathbf{m})}{2}\right)$$

Let $h_e(\mathbf{X})$ denote the differential entropy of $\mathbf{X}$ measured in nats. We will convert it to bits finally.

$$h_e(\mathbf{X}) = \int f(\mathbf{x})\ln\frac{1}{f(\mathbf{x})}d\mathbf{x}$$

Setting $\mathbf{y} = \mathbf{x} - \mathbf{m}$ in the above integral, we get

$$\begin{aligned}
h_e(\mathbf{X}) &= \frac{1}{\sqrt{(2\pi)^n|\mathbf{C}|}}\int \exp\left(-\frac{\mathbf{y}^T\mathbf{C}^{-1}\mathbf{y}}{2}\right)\left[\frac{1}{2}\left[\ln(2\pi)^n|\mathbf{C}|\right] + \frac{1}{2}\mathbf{y}^T\mathbf{C}^{-1}\mathbf{y}\right]d\mathbf{y} \\
&= \frac{1}{2}\left[\ln(2\pi)^n|\mathbf{C}|\right] + \frac{1}{\sqrt{(2\pi)^n|\mathbf{C}|}}\int \frac{\mathbf{y}^T\mathbf{C}^{-1}\mathbf{y}}{2}\exp\left(-\frac{\mathbf{y}^T\mathbf{C}^{-1}\mathbf{y}}{2}\right)d\mathbf{y}
\end{aligned}$$

We know that $\mathbf{C}$ will have an eigenvalue decomposition of the form $\mathbf{C} = \mathbf{U}\Sigma\mathbf{U}^T$ where $\mathbf{U}$ is an orthonormal matrix and $\mathbf{C}^{-1} = \mathbf{U}\Sigma^{-1}\mathbf{U}^T$. This decomposition also has the property that $|\mathbf{C}| = |\Sigma|$. We now make the substitution $\mathbf{w} = \Sigma^{\frac{-1}{2}}\mathbf{U}^T\mathbf{y} \Rightarrow \mathbf{y} = \mathbf{U}\Sigma^{\frac{1}{2}}\mathbf{w}$ in the above integral. The Jacobian of the transformation is given by $J = abs(|\mathbf{U}||\Sigma^{\frac{1}{2}}|) = 1 \times \sqrt{|\Sigma|} = \sqrt{|\Sigma|}$. Note that $|\Sigma^{\frac{1}{2}}| = \sqrt{|\Sigma|}$ since $\Sigma$ is a diagonal matrix. The above integral simplifies to,

$$h_e(\mathbf{X}) = \frac{1}{2}\left[\ln(2\pi)^n|\mathbf{C}|\right] + \frac{1}{\sqrt{(2\pi)^n|\mathbf{C}|}} \times \sqrt{|\Sigma|}\int \frac{\mathbf{w}^T\mathbf{w}}{2}\exp\left(-\frac{\mathbf{w}^T\mathbf{w}}{2}\right)d\mathbf{w}$$

Since $|\mathbf{C}| = |\Sigma|$, we have,

$$\begin{aligned}
&\frac{1}{\sqrt{(2\pi)^n|\mathbf{C}|}} \times \sqrt{|\Sigma|}\int \frac{\mathbf{w}^T\mathbf{w}}{2}\exp\left(-\frac{\mathbf{w}^T\mathbf{w}}{2}\right)d\mathbf{w} \\
=&\frac{1}{\sqrt{(2\pi)^n}}\int \frac{\mathbf{w}^T\mathbf{w}}{2}\exp\left(-\frac{\mathbf{w}^T\mathbf{w}}{2}\right)d\mathbf{w} \\
=&\frac{1}{2}E[W_1^2 + W_2^2 + \ldots + W_n^2]
\end{aligned}$$

where $W_i \sim$ i.i.d$N(0,1) \Rightarrow E[W_i^2] = 1\ \forall i$. Therefore,

$$\frac{1}{\sqrt{(2\pi)^n|\mathbf{C}|}} \times \sqrt{|\Sigma|}\int \frac{\mathbf{w}^T\mathbf{w}}{2}\exp\left(-\frac{\mathbf{w}^T\mathbf{w}}{2}\right)d\mathbf{w} = \frac{n}{2}$$

Finally, we have,

$$h_e(\mathbf{X}) = \frac{1}{2}\left[\ln(2\pi)^n|\mathbf{C}|\right] + \frac{n}{2} = \frac{1}{2}\ln\left[(2\pi e)^n|\mathbf{C}|\right]$$

Converting from nats to bits, we get

$$h(\mathbf{X}) = \frac{1}{2}\log_2\left[(2\pi e)^n|\mathbf{C}|\right]$$

**Part b**

Firstly, let $\mathbf{X} = \mathbf{X}_c + j\mathbf{X}_s$ where,

1. $E[\mathbf{X}_c] = \mathbf{m}_c, E[\mathbf{X}_s] = \mathbf{m}_s$
2. $E[(\mathbf{X}_c - \mathbf{m}_c)(\mathbf{X}_c - \mathbf{m}_c)^T] = E[(\mathbf{X}_s - \mathbf{m}_s)(\mathbf{X}_s - \mathbf{m}_s)^T] = \mathbf{C}_0$
3. $E[(\mathbf{X}_c - \mathbf{m}_c)(\mathbf{X}_s - \mathbf{m}_s)^T] = -E[(\mathbf{X}_s - \mathbf{m}_s)(\mathbf{X}_c - \mathbf{m}_c)^T] = \mathbf{C}_1$
4. $\mathbf{C} = E[(\mathbf{X} - \mathbf{m})(\mathbf{X} - \mathbf{m})^H] = 2(\mathbf{C}_0 - j\mathbf{C}_1)$

Let us now define $\mathbf{Y} = \begin{pmatrix} \mathbf{X}_c \\ \mathbf{X}_s \end{pmatrix}$. We have $h(\mathbf{X}) = h(\mathbf{X}_c, \mathbf{X}_s) = h(\mathbf{Y})$. Now $\mathbf{Y} \in \mathbb{R}^{2n}$ is Gaussian with

1. $E[\mathbf{Y}] = \mathbf{m}_y = \begin{pmatrix} \mathbf{m}_c \\ \mathbf{m}_s \end{pmatrix}$
2. $\mathbf{C}_y = E[(\mathbf{Y} - \mathbf{m}_y)(\mathbf{Y} - \mathbf{m}_y)^T] = \begin{pmatrix} \mathbf{C}_0 & \mathbf{C}_1 \\ -\mathbf{C}_1 & \mathbf{C}_0 \end{pmatrix}$

From part (a), we have $h(\mathbf{Y}) = \frac{1}{2}\log_2\big((2\pi e)^{2n}|\mathbf{C}_y|\big)$. Now, all we have to do is to relate $|\mathbf{C}_y|$ to $|\mathbf{C}|$. We proceed in the following manner

1. First, we show that $|2(\mathbf{C}_0 - j\mathbf{C}_1)| = |2(\mathbf{C}_0 + j\mathbf{C}_1)|$. Let $\mathbf{A}^T$ denote the transpose (*not* conjugate transpose, even if $\mathbf{A}$ is complex) of $\mathbf{A}$. We make the following observation,

$$
\begin{aligned}
(\mathbf{C}_0 - j\mathbf{C}_1)^T &= (\mathbf{C}_0^T - j\mathbf{C}_1^T) \\
&= \mathbf{C}_0 + j\mathbf{C}_1 && \because \mathbf{C}_0 = \mathbf{C}_0^T \text{and} \mathbf{C}_1 = -\mathbf{C}_1^T
\end{aligned}
$$

   Determinant is invariant under transposition (even if $\mathbf{A}$ is complex) i.e. $|\mathbf{A}^T| = |\mathbf{A}|$. We use this to conclude that $|2(\mathbf{C}_0 - j\mathbf{C}_1)| = |2(\mathbf{C}_0 + j\mathbf{C}_1)|$.

2. Using the formula for the determinant of a matrix in terms of its blocks , we have $|\mathbf{C}_y| = |\mathbf{C}_0^2 + \mathbf{C}_1\mathbf{C}_0^{-1}\mathbf{C}_1\mathbf{C}_0|$.

3. Let us evaluate $|\mathbf{C}|^2$. In the following steps, we will repeatedly use the fact that $|\mathbf{AB}| = |\mathbf{BA}| = |\mathbf{A}||\mathbf{B}|$. We have,

$$
\begin{aligned}
|\mathbf{C}|^2 &= |\mathbf{C}||\mathbf{C}| = |2(\mathbf{C}_0 - j\mathbf{C}_1)| \times |2(\mathbf{C}_0 + j\mathbf{C}_1)| && \text{From part a} \\
&= |2(\mathbf{I} - j\mathbf{C}_1\mathbf{C}_0^{-1})\mathbf{C}_0| \times |2(\mathbf{I} + j\mathbf{C}_1\mathbf{C}_0^{-1})\mathbf{C}_0| \\
&= |\big(2(\mathbf{I} - j\mathbf{C}_1\mathbf{C}_0^{-1})\mathbf{C}_0\big)\big(2(\mathbf{I} + j\mathbf{C}_1\mathbf{C}_0^{-1})\mathbf{C}_0\big)| && \because |\mathbf{A}||\mathbf{B}| = |\mathbf{AB}| \\
&= |[2(\mathbf{I} - j\mathbf{C}_1\mathbf{C}_0^{-1})] \times [2(\mathbf{I} + j\mathbf{C}_1\mathbf{C}_0^{-1})] \times \mathbf{C}_0 \times \mathbf{C}_0| && \because |\mathbf{ABCD}| = |\mathbf{ACBD}| \\
&= |4(\mathbf{I} + \mathbf{C}_1\mathbf{C}_0^{-1}\mathbf{C}_1\mathbf{C}_0^{-1}) \times \mathbf{C}_0^2| \\
&= 4^n|\mathbf{C}_0^2 + \mathbf{C}_1\mathbf{C}_0^{-1}\mathbf{C}_1\mathbf{C}_0| \\
&= 2^{2n}|\mathbf{C}_y| && \text{From part b}
\end{aligned}
$$

Therefore, $|\mathbf{C}_y| = \frac{1}{2^{2n}}|\mathbf{C}|^2$. Substituting back, we get

$$\begin{aligned} h(\mathbf{X}) = h(\mathbf{Y}) &= \frac{1}{2}\log_2\big((2\pi e)^{2n}|\mathbf{C}_y|\big) \\ &= \frac{1}{2}\log_2\big((2\pi e)^{2n}\frac{|\mathbf{C}|^2}{2^{2n}}\big) \\ &= \frac{1}{2}\log_2\big(\{(\pi e)^n|\mathbf{C}|\}^2\big) \\ &= \log_2\big((\pi e)^n|\mathbf{C}|\big) \end{aligned}$$

# Problem 6.5

## Part a

Let us define $Y_1 = aX, Y_2 = X + b$ where $a, b$ are some constants. We note that both $Y_1$ and $Y_2$ are invertible transformations of a discrete random variable. Therefore, $H(X|Y_1) = H(X|Y_2) = H(Y_1|X) = H(Y_2|X) = 0$. Let $Y$ denote any of the random variables $Y_1, Y_2$. Then,

$$H(X,Y) = H(X) + \underbrace{H(Y|X)}_{0} = H(Y) + \underbrace{H(X|Y)}_{0}$$

Therefore, $H(X) = H(Y)$ and we conclude that scaling or translating a discrete random variable doesn't change its entropy.

## Part b

Let us now define $Y$ to be $X+b$. Then $f_Y(y) = f_X(y-b)$. Let $\mathcal{S}_x$ and $\mathcal{S}_y$ denote the non-zero support of the pdf's of $X$ and $Y$ respectively. Then, $\mathcal{S}_y = \mathcal{S}_x + b$ in the sense that $x \,\epsilon\, \mathcal{S}_x \Leftrightarrow x + b \,\epsilon\, \mathcal{S}_y$. We have,

$$\begin{aligned} h(Y) &= \int_{\mathcal{S}_y} f_Y(y)\log_2\frac{1}{f_Y(y)}dy \\ &= \int_{\mathcal{S}_y} f_X(y-b)\log_2\frac{1}{f_X(y-b)}dy \end{aligned}$$

Setting $x = y - b$ in the above integral, we get

$$\begin{aligned} h(Y) &= \int_{\mathcal{S}_y - b} f_X(x)\log_2\frac{1}{f_X(x)}dx \\ &= \int_{\mathcal{S}_x} f_X(x)\log_2\frac{1}{f_X(x)}dx = h(X) \end{aligned}$$

Therefore, the differential entropy of a continuous random variable is invariant to translation.

Define the random variable $Z$ to be $aX$. We have, $f_Z(z) = \frac{1}{|a|}f_X(\frac{z}{a})$. Let $\mathcal{S}_x$ and $\mathcal{S}_z$ denote the non-zero support of the pdf's of $X$ and $Z$ respectively. Then, $\mathcal{S}_z = a\mathcal{S}_x$ in the sense that $x \,\epsilon\, \mathcal{S}_x \Leftrightarrow ax \,\epsilon\, \mathcal{S}_z$. Therefore,

$$h(Z) = \int_{\mathcal{S}_z} f_Z(z)\log_2\frac{1}{f_Z(z)}dz = \int_{\mathcal{S}_z} \frac{1}{|a|}f_X(\frac{z}{a})\log_2\frac{|a|}{f_X(\frac{z}{a})}dz$$

Setting $x = \frac{z}{a}$ in the above integral, we get,

$$
\begin{aligned}
h(Z) &= \int_{\frac{\mathcal{S}_z}{a}} f_X(x)\log_2\frac{|a|}{f_X(x)}dx = \int_{\mathcal{S}_x} f_X(x)\log_2\frac{1}{f_X(x)}dx + \int_{\mathcal{S}_x} f_X(x)\log_2|a|dx \\
&= h(X) + \log_2|a|
\end{aligned}
$$

Therefore, the differential entropy of a continuous random variable varies with scaling. We also note that by choosing $a$ sufficiently low, differential entropy can be made negative.

## Problem 6.6

From the symmetry of the channel, we guess that the optimal input distribution is $P(X=0) = P(X=1) = \frac{1}{2}$. All we need to do is to check that this guess satisfies the Kuhn-Tucker conditions. For the guessed input distribution, we have $P(Y=0) = P(Y=1) = \frac{1-q}{2}$ and $P(Y=e) = q$.

$$
\begin{aligned}
I(X=0;Y) &= \sum_y p(y|0)\log_2\frac{p(y|0)}{p(y)} \\
&= (1-q)\log_2 2 + q\log_2\frac{q}{q} + 0 = 1-q \\
I(X=1;Y) &= \sum_y p(y|1)\log_2\frac{p(y|1)}{p(y)} \\
&= 0 + q\log_2\frac{q}{q} + (1-q)\log_2 2 = 1-q
\end{aligned}
$$

Therefore, the Kuhn-Tucker conditions are satisfied and we have $C_E = I(X=0;Y) = I(X=1;Y) = 1-q$.

## Problem 6.7

From the symmetry of the channel, we guess that the optimal input distribution is $P(X=0) = P(X=1) = \frac{1}{2}$. With this input distribution, we have $P(Y=0) = P(Y=1) = \frac{1-q}{2}$ and $P(Y=e) = q$. Now, we check that the Kuhn-Tucker conditions are satisfied for our guess and read off the capacity.

$$
\begin{aligned}
I(X=0;Y) &= \sum_y p(y|0)\log_2\frac{p(y|0)}{p(y)} \\
&= (1-p-q)\log_2\frac{2(1-p-q)}{1-q} + q\log_2\frac{q}{q} + p\log_2\frac{2p}{1-q} \\
&= (1-p-q)\log_2\frac{2(1-p-q)}{1-q} + p\log_2\frac{2p}{1-q} \\
I(X=1;Y) &= \sum_y p(y|1)\log_2\frac{p(y|1)}{p(y)} \\
&= p\log_2\frac{2p}{1-q} + q\log_2\frac{q}{q} + (1-p-q)\log_2\frac{2(1-p-q)}{1-q} \\
&= p\log_2\frac{2p}{1-q} + (1-p-q)\log_2\frac{2(1-p-q)}{1-q}
\end{aligned}
$$

Therefore, the Kuhn-Tucker conditions are satisfied and the capacity of the Binary Errors and Erasure Channel is $C_{BEC} = p\log_2 \frac{2p}{1-q} + (1-p-q)\log_2 \frac{2(1-p-q)}{1-q}$

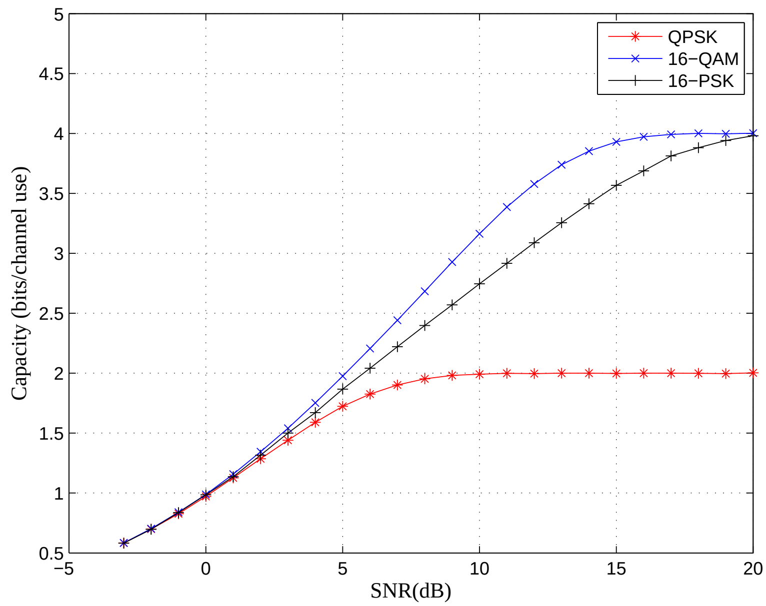
## Problem 6.8

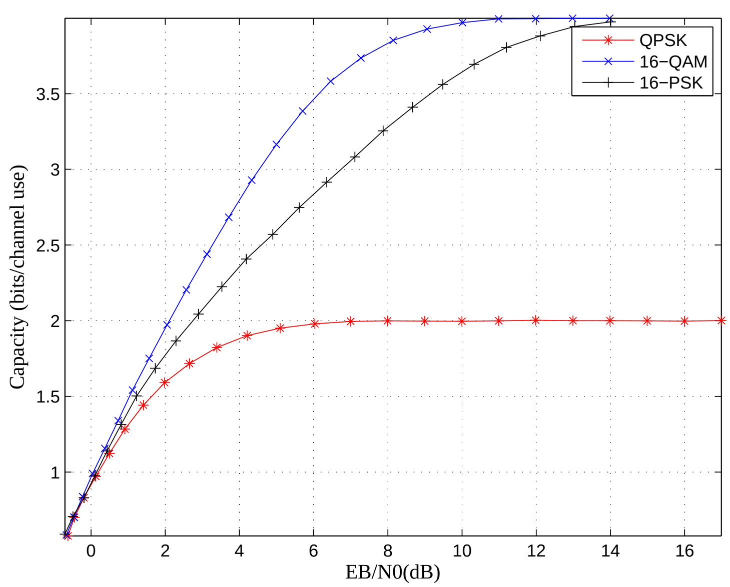
```
% This program computes the capacity for the QPSK,16-QAM and 16-PSK
% Constellations and we plot the resulting capacity against SNR

clear
clc

% ----------- CAPACITY FOR QPSK ------------------------------------ %

% The channel for QPSK can be described as follows:

% Y_i = sqrt(SNR)*X_i + Z_i
% Y_q = sqrt(SNR)*X_q + Z_q

% Here Z_q, Z_i ~ N(0,1) and X_i,X_q are drawn from i.i.d. from +/- 1
% The subscripts i and q refer to in-phase and quadrature components
% Since the in-phase and quadrature channels are orthogonal, the capacity
% is just twice that of any one of them. So we just calculate the capacity
% of one of these channels and multiply by 2

% I(X;Y) = h(Y) - h(Y|X)

h_yx = 0.5*log2(2*pi*exp(1));

% For calculating h(Y) we generate the bits i.i.d with p = 1/2 and compute
% the capacity numerically using the law of large numbers

snr_db = (-3:20)';
snr = 10.^(snr_db/10);

num_bits = 1e6;
capacity_qpsk = zeros(length(snr),1);

for ctr = 1:length(snr)
    bits = round(rand(num_bits,1));    % Generating the bits with equal probability
    bpsk_symbols = 1 - 2*bits;         % Mapping bits to symbols
    A = sqrt(snr(ctr));                % Defining A to be the "amplitude"
    noise = randn(num_bits,1);
    rec_symbols = A*bpsk_symbols + noise;

    % Now we find - E[log2(p(Y)] = h(Y)
    prob_y = (0.5/sqrt(2*pi))*(exp(-0.5*(rec_symbols - A).^2) + exp(-0.5*(rec_symbols + A).^
    h_y = -sum(log2(prob_y))/num_bits;

    % Finding the capacity
    capacity_qpsk(ctr) = 2*(h_y - h_yx);
end
```

```
% Plotting the capacity as a function of SNR
plot(snr_db,capacity_qpsk,'-*r')
hold on
ylim([0.5 5])

% ------------------------------------------------------------------------- %


% ---------------- CAPACITY FOR 16 - QAM ------------------------------ %

% Firstly, we notice that the 16-QAM is just 4-PAM's on orthogonal channels
% Therefore, the capacity of 16-QAM is just twice that of 4-PAM

% Let us say that the constellation points are at +/- d, +/- 3d and the
% noise has unit variance. Then we have 0.25*20*d^2 = SNR and hence d =
% sqrt(SNR/5). Hence, our channel model is Y = sqrt(SNR/20)*X + Z
% where Z ~ N(0,1) and X is drawn with probability 1/4 each from
% {+/-1,+/-3}

capacity_qam = zeros(length(snr),1);
num_symbols = 1e6;

for ctr = 1:length(snr)

    d = sqrt(snr(ctr)/5);

    % Generating the symbols
    % We do this as follows:
    % 1. We generate a sequence (seq_13) of  1's and 3's with equal probability
    % 2. We generate a sequence (seq_1_neg1) of +1 and -1 with equal probability
    % 3. We multiply the sequences and scale by d

    seq_13 = 2*round(rand(num_symbols,1)) + 1;
    seq_1_neg1 = 1 - 2*round(rand(num_symbols,1));
    pam_symbols = seq_13.*seq_1_neg1.*d;
    noise = randn(num_symbols,1);
    rec_symbols = pam_symbols + noise;

    % Now we calculate h(Y) as - E[log2(p(y)]
    prob_y = (0.25/sqrt(2*pi))*(exp(-0.5*(rec_symbols - d).^2) + ...
    exp(-0.5*(rec_symbols + d).^2) + exp(-0.5*(rec_symbols + 3*d).^2) + ...
    exp(-0.5*(rec_symbols - 3*d).^2));
    h_y = -sum(log2(prob_y))/num_symbols;

    capacity_qam(ctr) = 2*(h_y-h_yx);
end

plot(snr_db,capacity_qam,'-xb')

% ------------------------------------------------------------------------- %

% ------------------- CAPACITY FOR 16-PSK ------------------------------ %
```

```
% The model for 16-PSK is exactly the same as the derivation in the
% textbook with M = 16

% Now h(Y|X) = log2(pi*e)
h_yx = log2(pi*exp(1));

num_symbols = 1e5;
prob_y = zeros(num_symbols,1);
capacity_psk = zeros(length(snr),1);

for ctr = 1:length(snr)

    A = sqrt(snr(ctr));
    levels = round(16*rand(num_symbols,1));
    psk_symbols = A*exp(j*(2*pi/16)*levels);
    noise = (1/sqrt(2))*(randn(num_symbols,1) + j*randn(num_symbols,1));
    rec_symbols = psk_symbols + noise;

    % We form a vector consisting of the constellation points
    constellation_points = A*exp(j*(2*pi/16)*(0:15));
    constellation_points = constellation_points(:);

    rec_symbols_repeated = repmat(rec_symbols.',16,1);
    psk_symbols_repeated = repmat(constellation_points,1,num_symbols);
    diff_matrix = (rec_symbols_repeated - psk_symbols_repeated);
    prob_y = sum(exp(-(abs(diff_matrix)).^2),1)/(pi*16);

    h_y = -sum(log2(prob_y))/num_symbols;
    capacity_psk(ctr) = (h_y - h_yx);


end

plot(snr_db,capacity_psk,'-+k')
grid on
xlabel('SNR(dB)','FontName','Times','FontSize',12)
ylabel('Capacity (bits/channel use)','FontName','Times','FontSize',12)
legend('QPSK','16-QAM','16-PSK')

% ---------------- Getting the plots in terms of Eb/N0 -------------------

% For each constellation, we need to figure out the correspondence between
% Eb/N0 and SNR
% We have SNR = r(SNR) * (Eb/N0) => Eb/N0 = (SNR)/(r(SNR))

ebno_qpsk_db = 10*log10(snr./capacity_qpsk);
ebno_qam_db = 10*log10(snr./capacity_qam);
ebno_psk_db = 10*log10(snr./capacity_psk);

figure

plot(ebno_qpsk_db,capacity_qpsk,'-*r')
hold on
```

```
plot(ebno_qam_db,capacity_qam,'-xb')
plot(ebno_psk_db,capacity_psk,'-+k')
grid on
xlabel('EB/N0(dB)','FontName','Times','FontSize',12)
ylabel('Capacity (bits/channel use)','FontName','Times','FontSize',12)
axis tight
legend('QPSK','16-QAM','16-PSK')

% ----------------------------------------------------------------------------
```



## Problem 6.9

The probability of error for noncoherent FSK is given by

$$P_e = \frac{1}{2}\exp\left(-\frac{E_s}{2N_0}\right)$$

In our case, we have $\frac{E_b}{N_0} = 5$ dB with a rate-$\frac{1}{2}$ code, which means that

$$\frac{E_s}{N_0} = \frac{1}{2}\frac{E_b}{N_0} = \frac{\sqrt{10}}{2} = 1.5811$$

Therefore, the probability of error $P_e$ turns out to be 0.2268. The capacity of a Binary Symmetric Channel with a crossover probability of $p$ is $1 - H_B(p)$. The capacity of the "channel" that uses FSK modulation + AWGN + hard decisions upto decoder is

$$C_{channel} = 1 - H_B(0.2268) = 0.2276$$

Taking a BER of $10^{-5}$ to be virtually zero, we see that the Shannon limit for the capacity is 0.2276 bits/channel use while the designer claims to operate at 0.5 bits/channel use. Therefore, we shouldn't believe her claim.

## Problem 6.10

### Part a

To compute the capacity of this channel, we calculate the mutual information $I(X;Y) = h(Y) - h(Y|X)$. Of these, the term $h(Y|X)$ is easy to compute. We know that $h(Y|X=0) = h(Y|X=1) = \frac{1}{2}\log_2(2\pi e)$

$$\begin{aligned} h(Y|X) &= P(X=0)h(Y|X=0) + P(X=1)h(Y|X=1) \\ &= \left(\frac{1}{2}+\frac{1}{2}\right)\frac{1}{2}\log_2(2\pi e) = \frac{1}{2}\log_2(2\pi e) \end{aligned}$$

We also know that

$$\begin{aligned} f_Y(y) &= \frac{1}{2}\left[f_Y(y|X=0) + f_Y(y|X=1)\right] \\ &= \frac{1}{2}\frac{1}{\sqrt{2\pi}}\left[\exp\left(-\frac{(y-A)^2}{2}\right) + \exp\left(-\frac{(y+A)^2}{2}\right)\right] \end{aligned}$$

To calculate $h(Y)$, we generate $N$ samples of $Y$ ($y_k$, $k = 1, 2 \dots N$) according to the model, $Y = A(-1)^X + N(0,1)$ and then approximate $h(Y)$ as

$$\hat{h}(Y) \approx \frac{1}{N}\sum_{k=1}^{N} f_Y(y_k)\log_2\frac{1}{f_Y(y_k)}$$

The capacity can now be calculated as $C(SNR) = h(Y) - \frac{1}{2}\log_2(2\pi e)$. The curve is plotted along with Part (b) to make a comparison.

### Part b

A BSC is completely determined when we specify the crossover probability $\beta$. In our case, we have,

$$\begin{aligned}\beta &= P(Y<0|X=0) = P(Z<-A|X=0) = P(Z<-A)\\ &= \int_{-\infty}^{-A} \frac{1}{\sqrt{2\pi}} \exp\left(\frac{-t^2}{2}\right) dt = Q(A)\\ &= Q(\sqrt{SNR})\end{aligned}$$

Therefore, the capacity of the induced BSC is $C_{BSC} = 1 - H_B(Q(\sqrt{SNR}))$ where $H_B(p)$ is the binary entropy function.

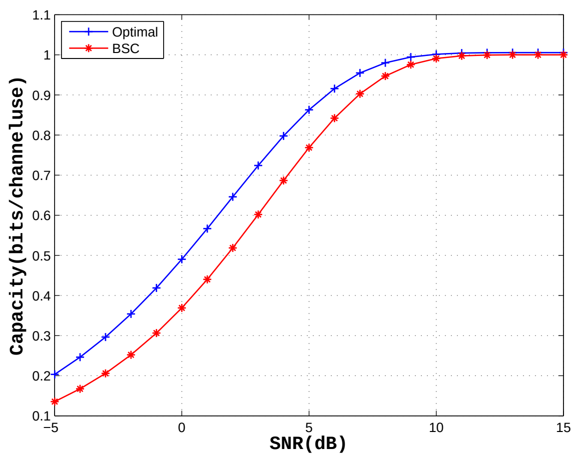



Figure 1: Comparing the capacity of the channel against the suboptimal use of the channel as a BSC

### Part c

From the figure, we see that the capacity of channel with hard decisions is 0.25 at - 2 dB (roughly) while the capacity of the channel with soft decisions is 0.25 at -4 dB (roughly). Therefore, the degradation caused by making hard decisions is about 2 dB.

### Part d

Since we are using a real-valued constellation, the spectral efficiency $r = 2C_A = 2 \times \frac{1}{4} = \frac{1}{2}$. We have $\text{SNR} = r\frac{E_b}{N_0}$. Therefore, $\frac{E_b}{N_0} = \frac{\text{SNR}}{r} = 2 \times \text{SNR}$. Therefore, $\left(\frac{E_b}{N_0}\right)_{hard} = -2 + 3\text{dB} = 1\text{dB}$ and $\left(\frac{E_b}{N_0}\right)_{soft} = -4 + 3\text{dB} = -1\text{dB}$

### Part e

Let $p$ denote the probability of making an error and $q$ denote the probability of an erasure. Then, we have the following relations,

$$\begin{aligned} p &= P(Y > \alpha | X = 1) = P(Y < -\alpha | X = 0) \\ &= P(-A + Z > \alpha) = P(Z > \alpha + A) \\ &= Q(\alpha + A) = Q(\alpha + \sqrt{SNR}) \end{aligned}$$

$$\begin{aligned} q &= P(|Y| < \alpha | X = 1) = P(|Y| < \alpha | X = 0) \\ &= P(|A + Z| < \alpha) = P(-\alpha < A + Z < \alpha) \\ &= Q(-\alpha - A) - Q(\alpha - A) \\ &= 1 - Q(\alpha + \sqrt{SNR}) - Q(\alpha - \sqrt{SNR}) \end{aligned}$$

### Part f

From Problem 6.7, the capacity of the errors and erasures channel is given by

$$C = p\log_2 \frac{2p}{1-q} + (1 - p - q)\log_2 \frac{2(1-p-q)}{1-q}$$

Substituting for $p$ and $q$ from Part e, we obtain the following plots.

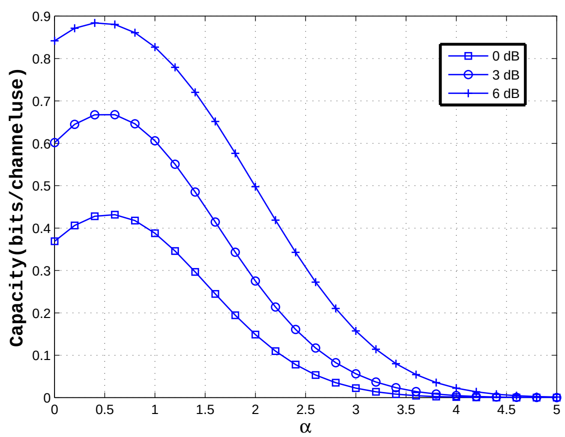



Figure 2: Plots of capacity against $\alpha$ for various values of SNR

### Part g

$\alpha = 0.5$ works well over the range 0-6 dB and is, in fact, the best value of $\alpha$ for 0 dB, 3 dB and 6 dB.

```
% Problem 6.10 - BPSK with errors and erasures

clear
```

```
clc
clf

% We generate n_data samples of X belonging to the set {0,1} with equal
% probability
n_data = 100000;
X = round(rand(n_data,1));

% We also generate n_data samples of noise
Z = randn(n_data,1);
% We define the range of SNR we are interested in
snr_db = -3:15;
snr = 10.^(snr_db/10);

% We now generate the received samples using the model:
% Y = sqrt(SNR)*(-1)^X + N(0,1)

% We store the capacity of the channel in cap_chan
cap_chan = zeros(length(snr),1);
for ctr = 1:length(snr)
    A = sqrt(snr(ctr));
    Y = A*(-1).^X + Z;
    % Let py_0 and py_1 denote the probability of receiving Y given that
    % 0 and 1 have transmitted
    py_0 = (exp(-((Y-A).^2)/2))/sqrt(2*pi);
    py_1 = (exp(-((Y+A).^2)/2))/sqrt(2*pi);
    % Let py denote the unconditional probability of receiving Y
    py = 0.5*(py_0 + py_1);
    % Let hy denote the differential entropy h(Y)
    hy = sum(log2(1./py))/n_data;
    cap_chan(ctr) = hy - 0.5*log2(2*pi*exp(1));
end

plot(snr_db,cap_chan,'b-+','LineWidth',1)
xlabel('SNR(dB)','FontName','Garamond','FontSize',14,'FontWeight','Bold')
ylabel('Capacity(bits/channeluse)','FontName','Garamond','FontSize',14, ...
'FontWeight','Bold')

% We now find the capacity of the Binary Symmetric Channel
% Let p_crossover denote the crossover probability
p_crossover = qfunc(sqrt(snr));
% Let cap_bsc denote the capacity of the BSC
cap_bsc = 1 + p_crossover.*log2(p_crossover) + (1 - p_crossover).*log2(1-p_crossover);

hold on
plot(snr_db,cap_bsc,'r-*','LineWidth',1)
legend('Optimal','BSC','Location','Northwest')
grid on


% We now move onto computing the capacity for the binary errors & erasures channel
% We compute the capacity as a function of alpha for a snr of 3dB
```

```
clear
figure

% Setting the required SNR to be 3 dB
snr_db = 3;
snr = 10.^(snr_db/10);
A = sqrt(snr);

% Getting the range of alpha in which we are interested
% cap_beec_3db stores the capacity for various values of alpha at 3 dB
alpha = 0:0.1:5;
cap_beec_3db = zeros(1,length(alpha));

for ctr = 1:length(alpha)
    % Let p denote the prob of error,q denote the prob of erasure
    p = qfunc(alpha(ctr)+A);
    q = qfunc(-alpha(ctr)-A) - qfunc(alpha(ctr)-A);
    cap_beec_3db(ctr) = p*log2((2*p)/(1-q)) + (1-p-q)*log2((2*(1-p-q))/(1-q));
end

plot(alpha,cap_beec_3db,'b-*','LineWidth',1)
xlabel('\alpha','FontName','Garamond','FontSize',14,'FontWeight','Bold')
ylabel('Capacity(bits/channeluse)','FontName','Garamond','FontSize',14, ...
'FontWeight','Bold')

grid on

% We now try to see if there is a value of alpha that works well for all
% values of SNR

clear
figure

snr_db = [0 3 6];
snr = 10.^(snr_db/10);
A = sqrt(snr);

% We now store the range of alpha's we are interested in
% cap_beec_across_snr stores the capacity of the channel across alpha,snr

alpha = 0:0.2:5;
cap_beec_across_snr = zeros(1,length(alpha));

for ctr_outer = 1:length(A)
    for ctr_inner = 1:length(alpha)
        p = qfunc(alpha(ctr_inner)+A(ctr_outer));
        q = qfunc(-alpha(ctr_inner)-A(ctr_outer)) - qfunc(alpha(ctr_inner)-A(ctr_outer));
        cap_beec_across_snr(ctr_inner) = p*log2((2*p)/(1-q)) + ...
    (1-p-q)*log2((2*(1-p-q))/(1-q));
    end
    plot(alpha,cap_beec_across_snr,'LineWidth',1)
    hold on
end
```

```
xlabel('\alpha','FontName','Garamond','FontSize',14,'FontWeight','Bold')
ylabel('Capacity(bits/channeluse)','FontName','Garamond','FontSize',14,'FontWeight','Bol
legend('0dB','3dB','6dB')
grid on
```

## Problem 6.11

### Part a

Firstly, we note that 16-QAM is just 4-PAM's along orthogonal carriers and therefore,

$$C_{16-QAM} = 2 \times C_{4-PAM}$$

where both systems are operating at the same value of $\frac{E_b}{N_0}$. Assume that the 4-PAM signal space is as shown in the figure. As a preliminary step, we need to figure out the distance between

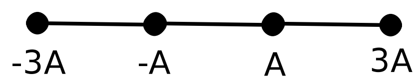



Figure 3: Representation of the 4-PAM signal space. Note that the distance between adjacent signal points is $2A$

the constellation points in terms of SNR. To do this, we need to assume the fact that all signal points are equiprobable. Let us assume that the distance between adjacent constellation points is $2A$. We have

$$E_s = \frac{1}{4}[2 \times A^2 + 2 \times 9A^2] = 5A^2$$

We assume that the noise variance (per dimension) $\sigma^2 = \frac{N_0}{2} = 1$. Therefore,

$$\text{SNR} = \frac{E_s}{\sigma^2} = 5A^2$$

$$\Rightarrow A = \sqrt{\frac{SNR}{5}}$$

Let us assume that the bits corresponding to each symbol are indexed as $(b_1, b_2)$. Let us characterize the channels seen by the bits separately and then add up the capacities. Strictly speaking, this might not be true. However, our primary purpose is to come up with a **model** that is a good approximation to the true scenario.

**Channel seen by $b_2$**
From the symmetry of the positioning of $b_2$, we have

$$\begin{aligned} P(error|b_2 = 1, +A \text{ tx}) &= P(error|b_2 = 1, -A \text{ tx}) \\ &= Q(A) + Q(3A) \end{aligned}$$

Therefore, by the law of total probability, we have

$$\begin{aligned} P(error|b_2 = 1) &= 2 \times \frac{\frac{1}{4}}{\frac{1}{4} + \frac{1}{4}}(Q(A) + Q(3A)) \\ &= Q(A) + Q(3A) \\ &\triangleq \eta_1 \end{aligned}$$

By a similar argument, we have,

$$\begin{aligned} P(error|b_2 = 0) &= 2 \times \frac{\frac{1}{4}}{\frac{1}{4} + \frac{1}{4}} (Q(A) - Q(5A)) \\ &= Q(A) - Q(5A) \\ &\triangleq \eta_0 \end{aligned}$$

Therefore, the channel seen by $b_2$ is a binary asymmetric channel. Instead of optimizing over the input distribution, let us just compute the capacity of the channel with equiprobable input. Note that this is a pretty good approximation in most cases and also what is most likely to be done in a practical scenario. Let us define $\mu_0 = \frac{1}{2}(\eta_1 + 1 - \eta_0)$. In this case, the capacity of the channel seen by bit $b_2$ is given by,

$$C_2 = H_B(\mu_0) - \frac{1}{2}\big(H_B(\eta_0) + H_B(\eta_1)\big)$$

**Channel seen by bit $b_1$**

For bit $b_1$, we first observe that $P(error|b_1 = 0) = P(error|b_1 = 1)$. Therefore it is enough to evaluate either one of these quantities. By the law of total probability,

$$\begin{aligned} P(error|b_1 = 0) &= P(error|b_1 = 0, -3A \text{ tx }) P(-3A \text{ tx } |b_1 = 0) + P(error|b_1 = 0, -A \text{ tx }) P(-A \text{ tx } |b_1 = 0) \\ &= Q(3A) \times \frac{\frac{1}{4}}{\frac{1}{2}} + Q(A) \times \frac{\frac{1}{4}}{\frac{1}{2}} \\ &= \frac{Q(A)}{2} + \frac{Q(3A)}{2} \end{aligned}$$

Therefore, the capacity of the channel seen by $b_1$ is $C_1 = 1 - H_B\left(\frac{Q(A)}{2} + \frac{Q(3A)}{2}\right)$ where $H_B(x)$ is the binary entropy function.

Therefore, the overall capacity of the 4-PAM is given by

$$\begin{aligned} C_{4-PAM} &= C_1 + C_2 \\ &= 1 - H_B\left(\frac{Q(A)}{2} + \frac{Q(3A)}{2}\right) + H_B(\mu_0) - \frac{1}{2}\big(H_B(\eta_0) + H_B(\eta_1)\big) \end{aligned}$$

The capacity of the 16-QAM system is given by $C_{16-QAM} = 2 \times C_{4-PAM}$. We now convert from SNR to $\frac{E_b}{N_0}$ and plot the results. We have also plotted the results for 16-QAM with soft decisions for comparison.

**Part b**

The total bandwidth available ($B$) is given to be 150 MHz. Since this scheme utilizes 50% excess bandwidth, we have

$$B_{min} = \frac{B}{1+a} = 100 \text{ MHz} \tag{2}$$

We know that the symbol rate $R_s = B_{min} = 100$ MHz and the required bit rate is 100 Mbps. Therefore, the spectral efficiency is 1 bit/s/Hz. We see that 16-QAM with hard decoding attains this spectral efficiency at $\frac{E_b}{N_0} = 2.48$ dB.

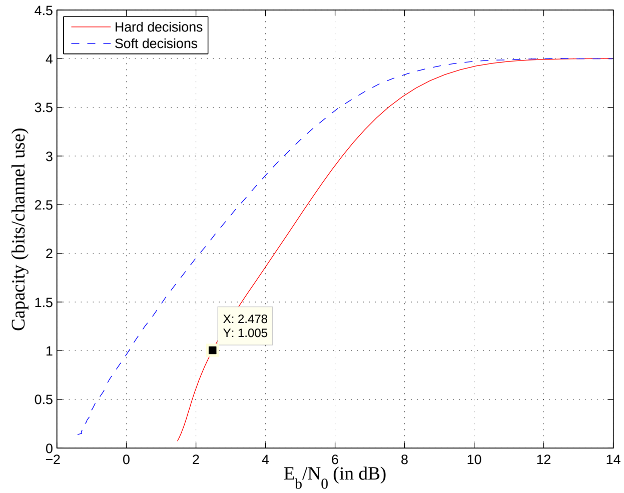



Figure 4: Comparison of 16-QAM systems with hard decision and soft decision decoding

### Part c

Once again, we note that QPSK is just BPSK on orthogonal channels and

$$C_{QPSK} = 2 \times C_{BPSK}$$

where both systems are operating at the same value of $\frac{E_b}{N_0}$. Therefore, for $C_{QPSK} = 1$, we need $C_{BPSK} = \frac{1}{2}$. From the results of Problem 10, this happens at an SNR of 2 dB. Therefore, $C_{QPSK} = 1$ at an SNR of 2 dB. This implies that

$$\frac{E_b}{N_0} = \frac{\text{SNR}}{r} = \frac{\text{SNR}}{1} = 2 \text{ dB}$$

Therefore, the minimum required value of $\frac{E_b}{N_0} = 2$ dB.

```
% This program evaluates the capacity of 16-QAM system

clear
clc
clf

% Range of interest of Eb/N0
snrdb = -10:0.5:20;
snrdb = snrdb(:);
snr = 10.^(snrdb/10);

% Defining the distance between points : 2d is the distance between
% adjacent constellation points
d = sqrt(snr/5);

% First we find the capacity of channel seen by bit 1
crossover_1 = 0.5*(qfunc(3*d) + qfunc(d));
cap_1 = 1 - binaryentropy(crossover_1);
```

```
% Next we find the capacity of the channel seen by bit 2
eta1 = qfunc(d) + qfunc(3*d);           % P(error | bit 2 = 1)
eta0 = qfunc(d) - qfunc(5*d);           % P(error | bit 1 = 0)
mu0 = 0.5*(eta1 + 1 - eta0);            % P(receiving a zero)
cap_2 = binaryentropy(mu0) - 0.5*(binaryentropy(eta0) + binaryentropy(eta1));

% Capacity of the system C = 2*(Cap1 + cap2)
capacity = 2*(cap_1 + cap_2);

% Calculating Eb/N0
ebno = snr./capacity;
ebnodb = 10*log10(ebno);

plot(ebnodb,capacity,'-r')
grid on
xlabel('E_b/N_0 (in dB)','FontName','Times','FontSize',14)
ylabel('Capacity (bits/channel use)','FontName','Times','FontSize',14)

% ----------------------- Capacity of QAM with soft decisions ---------
clear d
capacity_qam_soft = zeros(length(snr),1);
num_symbols = 1e6;

for ctr = 1:length(snr)

    d = sqrt(snr(ctr)/5);

    % Generating the symbols
    % We do this as follows:
    % 1. We generate a sequence (seq_13) of  1's and 3's with equal probability
    % 2. We generate a sequence (seq_1_neg1) of +1 and -1 with equal probability
    % 3. We multiply the sequences and scale by d

    seq_13 = 2*round(rand(num_symbols,1)) + 1;
    seq_1_neg1 = 1 - 2*round(rand(num_symbols,1));
    pam_symbols = seq_13.*seq_1_neg1.*d;
    noise = randn(num_symbols,1);
    rec_symbols = pam_symbols + noise;

    % Now we calculate h(Y) as - E[log2(p(y)]
    prob_y = (0.25/sqrt(2*pi))*(exp(-0.5*(rec_symbols - d).^2) + exp(-0.5*(rec_symbols + d).
    h_y = -sum(log2(prob_y))/num_symbols;
    h_yx = 0.5*log2(2*pi*exp(1));

    capacity_qam_soft(ctr) = 2*(h_y-h_yx);
end
ebnodb_soft = 10*log10(snr./capacity_qam_soft);

hold on
plot(ebnodb_soft,capacity_qam_soft,'--b')
xlabel('E_b/N_0 (in dB)','FontName','Times','FontSize',14)
ylabel('Capacity (bits/channel use)','FontName','Times','FontSize',14)
grid on
```

```
legend('Hard decisions','Soft decisions','Location','NorthWest')

% ---------------- Binary Entropy function -------------------------
function entropy = binaryentropy(in)

entropy = -in.*log2(in) - (1-in).*log2(1-in);
```

## Problem 6.12

### Part a

We have

$$\frac{N_1}{|h_1|^2} = \frac{1}{2}, \frac{N_2}{|h_2|^2} = \frac{2}{9}$$

Since $\frac{N_2}{|h_2|^2} < \frac{N_1}{|h_1|^2}$, we would use the second channel at "low" SNR.

### Part b

We would start using both channels for values of input power $P$ which satisfy,

$$\begin{aligned} P &> \frac{N_1}{|h_1|^2} - \frac{N_2}{|h_2|^2} \\ &= \frac{1}{2} - \frac{2}{9} = \frac{5}{18} \end{aligned}$$

### Part c

Let us first find the capacity with equal distribution of input power across the channels. Let the total available power be $P$. Then, the capacity is given by

$$\begin{aligned} C_{equal} &= \log_2\left(1 + \frac{2(\frac{P}{2})}{1}\right) + \log_2\left(1 + \frac{9(\frac{P}{2})}{2}\right) \\ &= \log_2\left(1 + P\right) + \log_2\left(1 + \frac{9P}{4}\right) \end{aligned}$$

Now we move on to finding the capacity with waterfilling. As we have seen, if the input power $P$ is less than $\frac{5}{18}$, we would use only channel 2. In this scenario, the capacity is

$$C_{waterfilling} = \log_2\left(1 + \frac{9P}{2}\right) \qquad\qquad P < \frac{5}{18}$$

When the input power is greater than $\frac{5}{18}$, we will be using both the channels. Let $P_1$ and $P_2$ be the powers allocated to channels 1 and 2 respectively. If $P$ is the total input power, we have

$$\begin{aligned} P_1 + P_2 &= P \\ P_2 - P_1 &= \frac{5}{18} \end{aligned}$$

Solving, we get $P_1 = \frac{P}{2} - \frac{5}{36}$ and $P_2 = \frac{P}{2} + \frac{5}{36}$. Therefore, the waterfilling capacity in this case is,

$$\begin{aligned} C_{waterfilling} &= \log_2\left(1 + \frac{2(\frac{P}{2} - \frac{5}{36})}{1}\right) + \log_2\left(1 + \frac{9(\frac{P}{2} + \frac{5}{36})}{2}\right) \\ &= \log_2\left(1 + P - \frac{5}{18}\right) + \log_2\left(1 + \frac{9P}{4} + \frac{5}{8}\right) \qquad P > \frac{5}{18} \end{aligned}$$

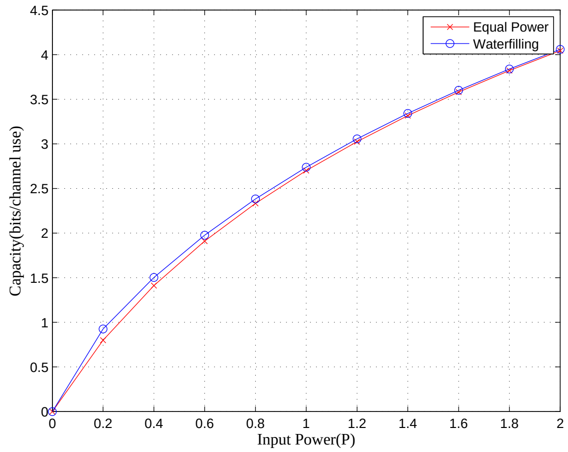



Figure 5: Comparing equal power and waterfilling schemes. We observe that except at low SNR, equal power allocation is very close to the waterfilling capacity

## Problem 6.13

### Part a

As a preliminary step, let us form the function $B(f) = \frac{S_n(f)}{|H(f)|^2}$. The function is described by,

$$\begin{aligned} B(f) &= 1 + \frac{f}{50} & 0 \le f < 40 \\ &= 0.45 + \frac{5}{1000}(f - 40) & 40 \le f < 100 \end{aligned}$$

Next, we have $N = \int S_n(f)df = \frac{1}{2} \times (1 + 3) \times 100 = 200$. Since the SNR is given to be 10 dB, the signal power $P$ is given by $P = 2000$. By the Kuhn-Tucker conditions, the optimal input signal PSD has to have the form,

$$S_s(f) = \begin{cases} a - B(f) & B(f) \le a \\ 0 & B(f) \ge a \end{cases}$$

Since the SNR is pretty high (10 dB), let us assume that the entire channel is used (which is equivalent to assuming $a > 1.8$). We will now solve for $a$ using the power constraint and then verify our guess. Assuming the water level comes upto a height $a$, the total signal power is nothing but the area of two trapezoids.

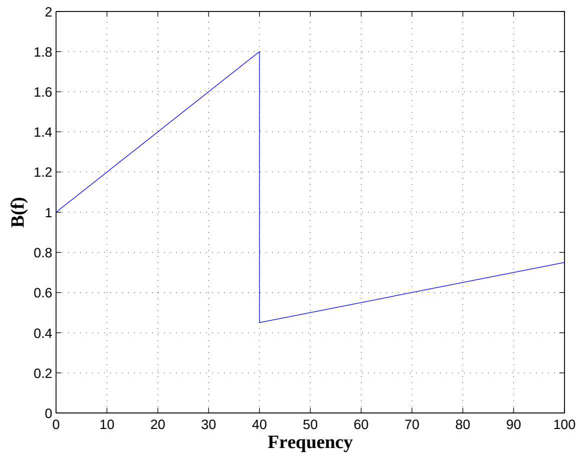



Figure 6: A plot of $B(f) = \frac{S_n(f)}{|H(f)|^2}$ versus frequency

$$
\begin{aligned}
P_{sig} = 2000 &= \frac{1}{2} \times 40 \times [(a-1) + (a-1.8)] + \frac{1}{2} \times 60 \times [(a-0.45) + (a-0.75)] \\
\Rightarrow a &= 20.92
\end{aligned}
$$

Since $a > 1.8$, our guess was right and the entire channel is used. Now, we have,

$$
S_s(f) = \begin{cases} 19.92 - \frac{f}{50} & 0 \le f < 40 \\ 20.47 - \frac{5}{1000}(f-40) & 40 \le f \le 100 \end{cases}
$$

The capacity is given by $C_{10dB} = \int_0^{100} \log_2\left(1 + \frac{S_s(f)}{B(f)}\right) df$ and we evaluate this integral numerically using MATLAB. We get $C_{10dB} = 465.21$ bits/sec.

### Part b

Let us once again guess that the entire channel is used. As discussed earlier, we would solve for $a$ (as defined in Part a) and check whether it is greater than 1.8. Since SNR is 0 dB, we have $P_{sig} = 200$. Just as before, we solve,

$$
\begin{aligned}
P_{sig} = 200 &= \frac{1}{2} \times 40 \times [(a-1) + (a-1.8)] + \frac{1}{2} \times 60 \times [(a-0.45) + (a-0.75)] \\
\Rightarrow a &= 2.92
\end{aligned}
$$

Since $a > 1.8$, our guess turned out to be correct. In this case, the optimal input PSD is given by,

$$
S_s(f) = \begin{cases} 1.92 - \frac{f}{50} & 0 \le f < 40 \\ 2.47 - \frac{5}{1000}(f-40) & 40 \le f \le 100 \end{cases}
$$

The capacity is given by $C_{0dB} = \int_0^{100} \log_2\left(1 + \frac{S_s(f)}{B(f)}\right) df$ and once again, we evaluate this integral numerically using MATLAB. We get $C_{0dB} = 181.12$ bits/sec.

### Part c

For the case when $P_{sig} = 2000$, and the power is distributed uniformly over the available band, we have,

$$S_s(f) = 20 \quad (0 \le f \le 100)$$

The capacity is given by $C_{10dB,unif} = \int_0^{100} \log_2\left(1 + \frac{S_s(f)}{B(f)}\right) df$ which turns out to be 465.177 bits/s.

When the SNR is 0 dB and $P_{sig} = 200$, we have $S_s(f) = 2 \quad 0 \le f \le 100$ and the capacity evaluates to 179.676 bits/s

## Problem 6.14

### Part a

We are given that,

$$h(t) = 2\delta(t-1) - \frac{j}{2}\delta(t-2) + (1+j)\delta(t-3.5)$$

This implies that,

$$H(f) = 2e^{-j2\pi f} - \frac{j}{2}e^{-j4\pi f} + (1+j)e^{-j7\pi f}$$

Let us denote the signal power by $P_s$ and since the PSD of the signal is flat, we have $S_s(f) = \frac{P_s}{W}, \quad \frac{-W}{2} \le f \le \frac{W}{2}$. Let the power spectral density of the noise $S_n(f) = \frac{N_0}{2}$. The noise power is then given by $P_n = N_0 W$. We then have the following identity,

$$\frac{S_s(f)}{S_n(f)} = \frac{P_s}{N_0 W} = \text{SNR} = 10$$

Therefore, the capacity is given by

$$\begin{aligned} C_W &= \int_{-\frac{W}{2}}^{\frac{W}{2}} \log_2\left(1 + \frac{S_s(f)}{S_n(f)}|H(f)|^2\right) df \\ &= \int_{-\frac{W}{2}}^{\frac{W}{2}} \log_2\left(1 + 10\,|H(f)|^2\right) df \end{aligned}$$

We evaluate the integral using MATLAB and plot the capacity as a function of $W$. We observe that $\frac{C_W}{W}$ is almost a constant and hence $C_W$ scales linearly with $W$.

## Part c

For this part, we assume that the PSD of the noise is flat and is given by $S_n(f) = N_0 = 1, \quad |f| < 5$. Since an SNR of 10 dB is reasonably high, a safe guess would be that the entire channel will be used. Let us make this assumption and we shall verify this later. Let $S_s(f)$ denote the PSD of the signal. Then, we have,

$$\begin{aligned} S_s(f) &= a - \frac{S_n(f)}{|H(f)|^2} \\ &= a - \frac{1}{|H(f)|^2} \end{aligned}$$

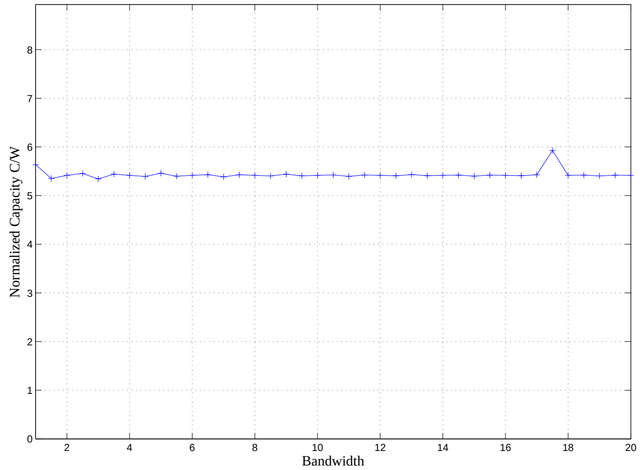



Figure 7: Plot of normalized capacity $\frac{C_W(SNR)}{W}$ versus $W$ for $SNR$ of 10 dB. We observe that the normalized capacity is almost a constant

where $a$ is a constant whose value will be decided by the power constraint. Since the SNR is 10 dB and the noise power is $N_0W = 10$, the signal power $P_{sig} = 100$. Therefore, we have,

$$\begin{aligned} P_{sig} = 100 &= \int_{\frac{-W}{2}}^{\frac{W}{2}} S_s(f)df \\ &= 10a - \int_{\frac{-W}{2}}^{\frac{W}{2}} \frac{1}{|H(f)|^2}df \\ \Rightarrow a &= 10 + \frac{1}{10}\int_{\frac{-W}{2}}^{\frac{W}{2}} \frac{1}{|H(f)|^2}df \end{aligned}$$

We evaluate the integral using MATLAB and we get $a = 11.09$. We also numerically confirm that $S_s(f) \geq 0$ for this particular value of $a$. Therefore, we have,

$$S_s(f) = 11.09 - \frac{1}{|H(f)|^2}$$

Once again,we use MATLAB to compute the capacity given by the formula,

$$C(S_s) = \int_{\frac{-W}{2}}^{\frac{W}{2}} \log_2\left(1 + \frac{|H(f)|^2 S_s(f)}{S_n(f)}\right)df$$

The capacity with waterfilling turns out to be 54.713 bits/s (assuming $W$ is in Hz). The capacity with white input is 54.152 bits/s and the **improvement is 0.5615 bits/s**.

```
% Problem 6.14 - Evaluating the capacity of a multipath channel

clear
clc
clf
```

```
% W represents the range of frequencies over which we evaluate the capacity
% Capacity is stored in the vector capacity_white_input
W = [1:0.5:20];
capacity_white_input = zeros(1,length(W));

% Cf represents the function to be integrated i.e Cf = 1 + 10*|H(f)|^2

for ctr = 1:length(W)
    Cf = @(f) log2(1 + 10*(abs(2*exp(-j*2*pi*f) - j/2*exp(-j*4*pi*f) + ...
    (1+j)*exp(-j*7*pi*f))).^2);
    capacity_white_input(ctr) = quad(Cf,-W(ctr)/2,W(ctr)/2);
end
capacity_white_normalized = capacity_white_input./W;

plot(W,capacity_white_normalized,'b-+')
ylim([0 max(capacity_white_normalized+3)]);
xlim([1 20])
xlabel('Bandwidth','FontName','Garamond','FontSize',14,'FontWeight','Bold')
ylabel('Normalized Capacity C/W','FontName','Garamond','FontSize',14, ...
'FontWeight','Bold')
grid on


% Setting N0 to be 1, when W = 10, we get a to be 10 + (1/10) \int
% 1/|H(f)|^2 df

Cf = @(f) 1./(abs(2*exp(-j*2*pi*f) - j/2*exp(-j*4*pi*f) + ...
         (1+j)*exp(-j*7*pi*f)).^2);
a = 10 + 0.1*quad(Cf,-5,5);

% Checking if all values of a - Sn(f)/|H(f)|^2 are greater than zero
f = [-5:0.1:5];
all(a-1./(abs(2*exp(-j*2*pi*f) - j/2*exp(-j*4*pi*f) + (1+j)*...
    exp(-j*7*pi*f)).^2) > 0)

% Since the answer turns out to be one, we use the entire channel
% The capacity when we use waterfilling is stored in cap_waterfilling
Cf = @(f) log2(a*(abs(2*exp(-j*2*pi*f) - j/2*exp(-j*4*pi*f) + (1+j)*...
    exp(-j*7*pi*f)).^2));
cap_waterfilling = quad(Cf,-5,5);
% We recompute the capacity with white input and store it in cap_white_w10
Cf = @(f) log2(1+10*(abs(2*exp(-j*2*pi*f) - j/2*exp(-j*4*pi*f) + (1+j)*...
    exp(-j*7*pi*f)).^2));
cap_white_w10 = quad(Cf,-5,5);
improvement = cap_waterfilling - cap_white_w10
```

## Problem 6.15

In this problem, we investigate the performance of the Blahut-Arimoto algorithm for the Binary Symmetric Channel with a crossover probability of 0.1. We start off with an initial guess of $P(X = 1) = 1 - P(X = 0) = 0.3$. The recursive equations are found in the textbook. The results obtained are as follows:

| Iteration Number | $P(X=0)$ | $P(X=1)$ |
|---|---|---|
| 0 | 0.7 | 0.3 |
| 1 | 0.5785 | 0.4215 |
| 2 | 0.5286 | 0.4714 |
| 3 | 0.5103 | 0.4897 |
| 4 | 0.5037 | 0.4963 |
| 5 | 0.5013 | 0.4987 |

We see that the input distribution is converging towards the optimal distribution of $P(X=0) = P(X=1) = \frac{1}{2}$

```
% Problem 6.15 - Blahut Arimoto Iterations for the BSC

clear
clc

% Denote the crossover probability by p_crossover
p_crossover = 0.1;

% Denote the "guessed" probability of a 1 by p_1 and the probability of a 0 by p_0
p_1 = 0.3;
p_0 = 1 - p_1;

% We shall denote the prior probabilities q(x|y) by q_xy
% Formulae for the q(x|y) are in the text
% Denote the number of iterations by niter
niter = 5;
% We store the values of p_1 as it varies in the vector named
% p1_across_iterations
p1_across_iterations = zeros(1,niter);

for iter = 1:niter

    q_10 = (p_1*p_crossover)/(p_1*p_crossover + p_0*(1-p_crossover));
    q_00 = 1 - q_10;
    q_01 = (p_0*p_crossover)/(p_0*p_crossover + p_1*(1-p_crossover));
    q_11 = 1 - q_01;

    % We now calculate the values of p_1 and p_0 for the next iteration
    p1_numerator = q_10^p_crossover * q_11^(1-p_crossover);
    p1_denominator = p1_numerator + q_01^p_crossover*q_00^(1-p_crossover);
    p_1 = p1_numerator / p1_denominator;
    p_0 = 1 - p_1;

    % We store the value of p_1 after this iteration
    p1_across_iterations(iter) = p_1;
end

p1_across_iterations
```

## Problem 16

### Part a

Let us set $P[X = \pm d] = p$ and $P[X = \pm 3d] = \frac{1}{2} - p$. Then, the average signal power is

$$\begin{aligned} P_{sig} &= 2 \times (pd^2 + (\frac{1}{2} - p)9d^2) \\ &= 2d^2 \times (\frac{9}{2} - 8p) \end{aligned}$$

We are given that $P_{noise} = 1$ and at an SNR of 3 dB, we have $\frac{P_{sig}}{P_{noise}} = 2$. From this, we get,

$$\begin{aligned} 2 &= \frac{2d^2 \times (\frac{9}{2} - 8p)}{1} \\ \Rightarrow d^2 &= \frac{1}{\frac{9}{2} - 8p} \end{aligned}$$

In general, for an SNR value of $\eta$, we will have

$$d^2 = \frac{\eta}{9 - 16p}$$

### Part b

For this part, we first derive the recursive equations for the probabilities of the various constellation points. Let $x_k$ denote a constellation point and assume we have completed $i$ iterations of the Blahut-Arimoto algorithm. Let $p_i(x_k)$ denote the probability assigned to point $x_k$ after the $i^{th}$ iteration. Then, we have,

$$p_{i+1}(x_k) = \nu \exp\Bigg( \int p(y|x_k) \log q(x_k|y) dy \Bigg)$$

where $\nu$ is a normalizing constant and

$$q(x_k|y) = \frac{p_i(x_k)p(y|x_k)}{p(y)}$$

From this, we have,

$$\begin{aligned} \log p_{i+1}(x_k) &= \log \nu + \int p(y|x_k) \log \frac{p_i(x_k)p(y|x_k)}{p(y)} dy \\ &= \log \nu + \log p_i(x_k) + \int p(y|x_k) \log p(y|x_k) dy - \int p(y|x_k) \log p(y) dy \\ &= \log p_i(x_k) - E[\log p(Y)|X_k] + \underbrace{\log \nu + \frac{1}{2}\log_2(2\pi e)}_{\log \nu'} \\ &= \log p_i(x_k) - E[\log p(Y)|X_k] + \log \nu' \end{aligned}$$

where $\nu'$ may be determined from the constraint $\sum_k p_{i+1}(x_k) = 1$. We use Monte Carlo simulation to calculate $E[\log p(Y)|X_k]$.

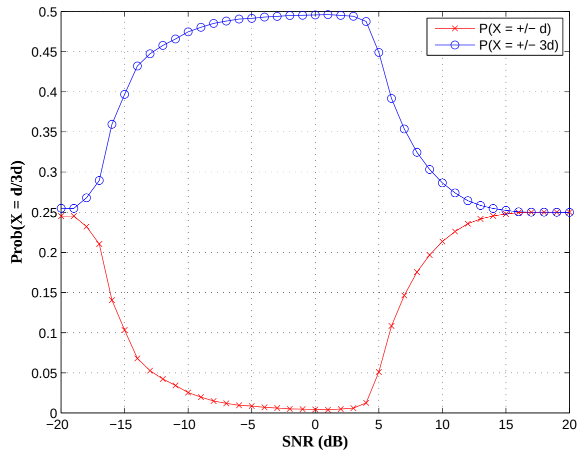



Figure 8: $P(X = \pm d)$ and $P(X = \pm 3d)$ vs SNR(dB)

### Part c

We see that at extremely high/low SNR, the uniform distribution is optimal whereas at "moderate" SNR, we try to push the points as far apart and virtually use it as a 2-PAM system.

```
{% This program optimizes the 4-PAM constellation using the Blahut-Arimoto
% algorithm

clear
clc

rand('twister',sum(100*clock));
randn('state',sum(100*clock));

% We first define the model
% Y = X + Z
% Z ~ N(0,1)
% X comes from {+/-d, +/- 3d}
% So if SNR is fixed to be h dB, d = sqrt(10^(h/10)/(9 - 16p))
% where p = P(X = +/- d), 0.5 - p = P(X = +/- 3d)



% We now define the parameters of the problem
snrdb = -20:1:20;                  % Range of interest of SNR in dB
snr = 10.^(snrdb/10);              % SNR in "actual" value
epsilon = 0.001;                   % Used in terminating criterion
N = 100000;                         % Number of points used to compute
                                   % expectation
const = 1/sqrt(2*pi);              % Scaling factor for Gaussian RV

% We initialize some vectors we shall use later
p_d = zeros(length(snr),1);        % Prob. of the points at +/- d
p_3d = zeros(length(snr),1);       % Prob. of the points at +/- 3d
```

```
for ctr = 1:length(snr)

    p1 = 0.25;                    % Prob of +/- d in current iter
    p3 = 0.25;                    % prob. of +/- 3d in current iter

    p1_prev = 0;                  % Prob. of +/-d in previous iter
    p3_prev = 0;                  % Prob. of +/-3d in previous iter



    while(norm(p1-p1_prev) > epsilon)         % Terminating criterion

        d = sqrt(snr(ctr)/(9 - 16*p1));     % Calculating d

        for ctr_inner = 1:2
            y = (2*ctr_inner-1)*d + randn(N,1);
            py_d = const*exp(-(y-d).^2/2);       % Calculating P(Y| +d)
            py_3d = const*exp(-(y-3*d).^2/2);    % Calculating P(Y| +3d)
            py_negd = const*exp(-(y+d).^2/2);    % Calculating P(Y| -d)
            py_neg3d = const*exp(-(y+3*d).^2/2); % Calculating P(Y| -3d)
            py = p1*(py_d + py_negd) + p3*(py_3d + py_neg3d);

            if (ctr_inner == 1)
                scaling_p1 = exp(-sum(log(py))/N);
            else
                scaling_p3 = exp(-sum(log(py))/N);
            end
        end

            p1_prev = p1;                   % Storing values for comparing
            p3_prev = p3;

            p1 = p1*scaling_p1;
            p3 = p3*scaling_p3;

            temp = (p1 + p3);            % Scaling values so that they
            p1 = 0.5*p1/temp;                   % add to 0.5
            p3 = 0.5*p3/temp;
    end

    p_d(ctr) = p1;
    p_3d(ctr) = p3;


end

plot(snrdb,p_d,'-xr')
hold on
plot(snrdb,p_3d,'-ob')
grid on
xlabel('SNR (dB)','FontName','Times','FontSize',12,'FontWeight','Bold')
ylabel('Prob(X = d/3d)','FontName','Times','FontSize',12,'FontWeight',...
       'Bold')
```

```
legend('P(X = +/- d)','P(X = +/- 3d)')
```

# Solutions to Chapter 7 Problems

*Fundamentals of Digital Communication*

## Problem 7.1

### Part a

The transfer function for the running example is

$$\begin{aligned} T(I,X) &= \frac{IX^5}{1-2IX} \\ &= IX^5\left(1+2IX+4I^2X^2+8I^3X^3+\ldots\right) \end{aligned}$$

The coefficient of $X^7$ is $4I^3 \Rightarrow$ the minimum number of differing input bits must be 3 to produce codewords that have a Hamming distance of 7.

### Part b

From the transfer function, we see that the coefficient of $I^4$ is $8X^8 \Rightarrow$ the maximum output weight that can be generated, by an input codeword of weight 4, is 8.

### Part c

The pairwise error probability is $q = Q\left(\sqrt{\frac{2RE_b}{N_0}d_H(\mathbf{c_1},\mathbf{c_2})}\right)$ where $R=\frac{1}{2}, \frac{E_b}{N_0} = 7dB \approx 5, d_H(\mathbf{c_1},\mathbf{c_2}) = 7$. Substituting these values, we get

$$\begin{aligned} P_{pairwise} &\approx Q\left(\sqrt{35}\right) \\ &= 1.58\times 10^{-9} \end{aligned}$$

### Part d

We modify the transfer function as follows:

**1.** $k$ ones on a branch $\Rightarrow$ factor of $W^k$ on the state transition

**2.** Factor of $L$ on every branch

Let $a$ refer to the state 10, $b$ refer to the state 11, $c$ refer to the state 01 and $d$ to the terminal 00 state. Therefore we get the following equations,

$$\begin{aligned} T_a &= LW^2 + LT_c \\ T_b &= LW(T_a+T_b) \\ &= \frac{LW}{1-LW}T_a \\ T_c &= LW(T_a+T_b) \\ &= \frac{LW}{1-LW}T_a \end{aligned}$$

Using the last equation in the first, we get the following,

$$\begin{aligned} T_c &= \frac{L^2W^3}{1 - LW - L^2W} \\ T_d &= LW^2T_c \\ &= \frac{L^3W^5}{1 - LW - L^2W} \end{aligned}$$

We have

$$T(W, L) = T_d = L^3W^5 \sum_{k=0}^{\infty} W^k(L + L^2)^k$$

We need the coefficient of $W^{100}$ which turns out to be

$$L^{98} \times (1 + L)^{95} = L^{98} \sum_{m=0}^{95} \binom{95}{m} L^m$$

We are actually looking for a term like $L^{190}$ and we find that there are $\binom{95}{92} = \binom{95}{3}$ events that satisfy the required criteria.

## Problem 7.2

Let the contents of the delay elements at time $k-1$ be $(y_{k-1}, y_{k-2})$. Then the state at time $k-1$ is $(s_1', s_2') = (y_{k-1}, y_{k-2})$. Let the input bit at time $k$ be denoted by $u_k$. Then the output bits at time $k$ ,which we denote by $(u_k, v_k)$ are given by,

$$\begin{aligned} u_k &= u_k \\ v_k &= y_k + y_{k-2} \\ &= (u_k + y_{k-1} + y_{k-2}) + y_{k-2} = u_k + y_{k-1} \\ &= u_k + s_1' \end{aligned}$$

Therefore $(u_k, v_k) = (u_k, u_k + s_1')$. Let the state at time $k$ be given by $(s_1, s_2)$. Then, we have,

$$\begin{aligned} (s_1, s_2) &= (y_k, y_{k-1}) = (u_k + y_{k-1} + y_{k-2}, y_{k-1}) \\ &= (u_k + s_1' + s_2', s_1') \end{aligned}$$

From figure 6, the following equations follow immediately:

$$\begin{aligned} T_a &= IX^2 + T_c \\ T_b &= XT_a + IXT_b \\ \Rightarrow T_b &= \frac{X}{1 - IX} T_a \\ T_c &= XT_b + IXT_a \\ &= \left( \frac{X^2}{1 - IX} + IX \right) T_a \\ \Rightarrow T_a &= \frac{1 - IX}{IX - I^2X^2 + X^2} T_c \end{aligned}$$

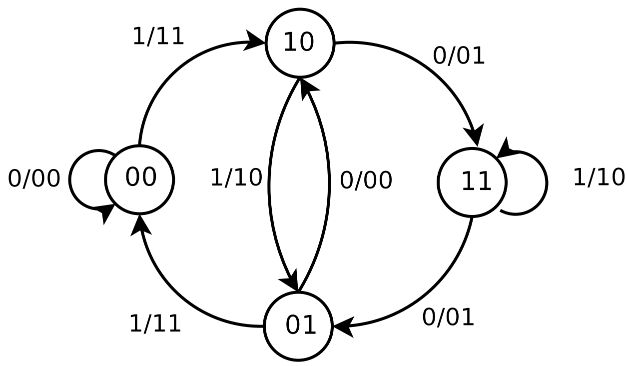


Figure 1: State diagram for [7,5] rate $\frac{1}{2}$ convolutional code with recursive systematic encoding

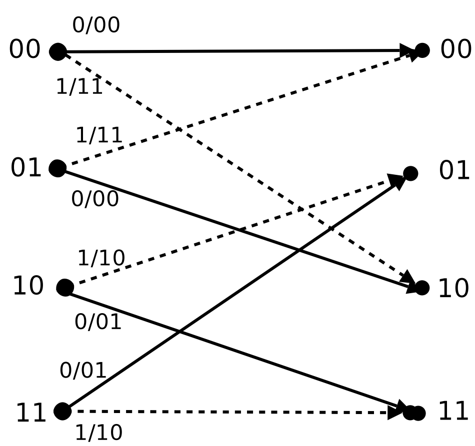


Figure 2: Trellis section for [7,5] rate $\frac{1}{2}$ convolutional code with recursive systematic encoding

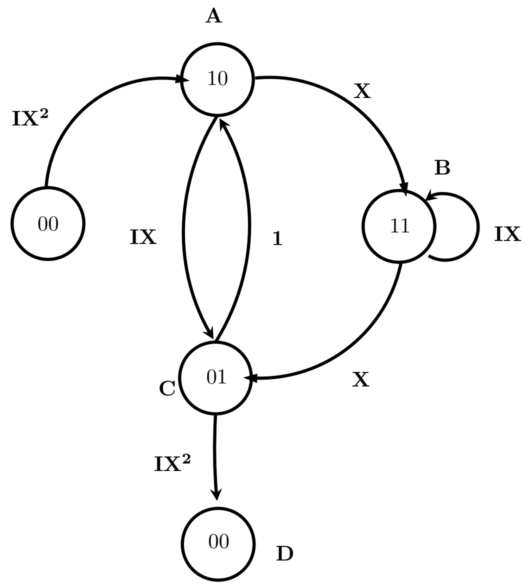



Figure 3: Modified state diagram for [7,5] rate $\frac{1}{2}$ convolutional code with recursive systematic encoding

From the first equation, we have,

$$T_c\left(\frac{1-IX}{IX-I^2X^2+X^2}-1\right)=IX^2$$
$$\Rightarrow T_c=\frac{IX^2(IX-I^2X^2+X^2)}{1-2IX-X^2+I^2X^2}$$

Using the fact that $T(I,X)=T_d=IX^2T_c$, we have,

$$T(I,X)=\frac{I^2X^4(X^2-I^2X^2+IX)}{1-2IX-X^2+I^2X^2}$$

## Problem 7.3

### Part a

Let the contents of the shift register at time $k-1$ be $(s_1', s_2')$. Let the input bit at time $k$ be $u_k$ and the output bits at time $k$ be $(w_k, z_k)$. Then, we have,

$$(w_k, z_k) = (u_k + s_1' + s_2', u_k + s_1')$$

Let $(s_1, s_2)$ denote the state at time $k$. Then,

$$(s_1, s_2) = (u_k, s_1')$$

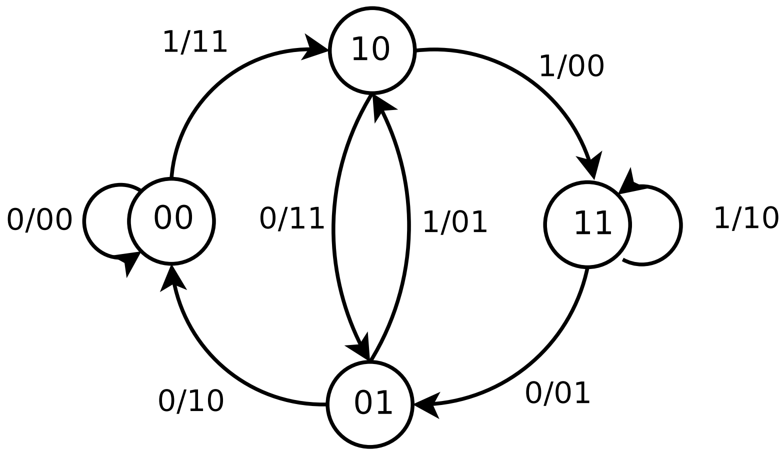



Figure 4: State diagram for [7,6] rate $\frac{1}{2}$ convolutional code with nonrecursive nonsystematic encoding

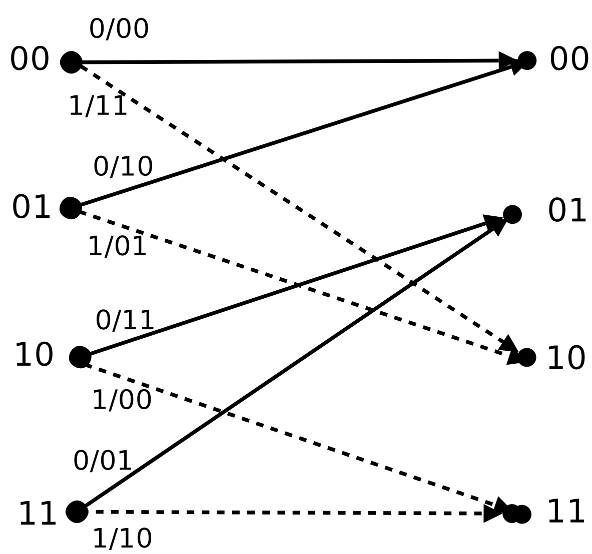



Figure 5: Trellis section for [7,6] rate $\frac{1}{2}$ convolutional code with nonrecursive nonsystematic encoding

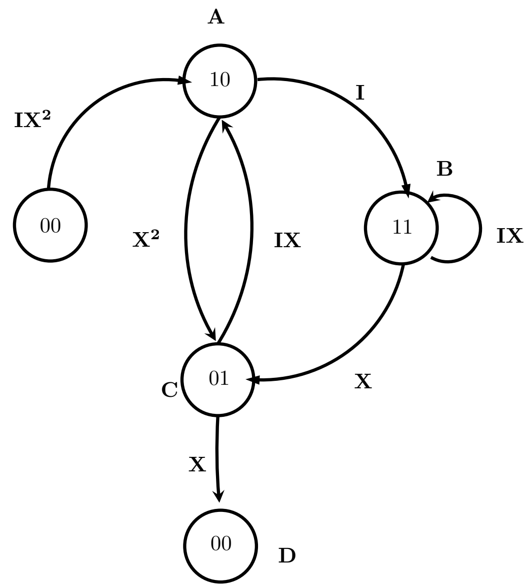



Figure 6: Modified state diagram for [7,5] rate $\frac{1}{2}$ convolutional code with recursive systematic encoding

**Part b**

**Part c**

From the figure, we have the following equations:

$$\begin{aligned}
T_a &= IX^2 + IXT_c \\
T_b &= IXT_b + IT_a \\
T_c &= X^2 T_a + XT_b \\
T_d &= XT_c
\end{aligned}$$

Solving these equations, we get

$$T(I, X) = T_d = IX^4 \frac{X - IX^2 + I}{1 - IX - IX^3 + IX^4 - I^2X^2}$$

**Part d**

To derive the transfer function, $T(X)$, we set $I = 1$ and we end up with

$$T(X) = X^4 \frac{1 + X - X^2}{1 - X - X^2 - X^3 + X^4}$$

We now set $Y = X + X^2 + X^3 - X^4$ and $T(X) = X^4(1 + X - X^2)(1 + Y + Y^2 + \ldots)$. From this it is clear that the lowest power of $X$ in the expansion of $T(X)$ is 4 and therefore, $d_{free} = 4$. This is lower than that of the running example which has $d_{free} = 5$.

## Problem 7.4

### Part a

In this problem, we analyze a rate $\frac{2}{3}$ convolutional code in which two inputs $(u_1[k], u_2[k])$ come in and three outputs $(y_1[k], y_2[k], y_3[k])$ are emitted at every time $k$. The input output relation is described by

$$\begin{aligned} y_1[k] &= u_1[k] + u_1[k-1] + u_2[k-1] \\ y_2[k] &= u_1[k-1] + u_2[k] \\ y_3[k] &= u_1[k] + u_2[k] \end{aligned}$$

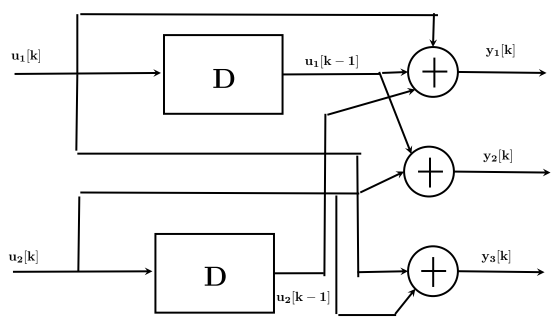



Figure 7: Shift register implementation of the encoding function

We now proceed to evaluate the transfer function $T(X)$. Writing out "conservation" equations at each node, we get

$$\begin{aligned} T_a &= X^2 + X^3 T_a + X^2 T_b + X T_c \\ T_b &= X^2 + X^2 T_b + X T_a + X^3 T_c \\ T_c &= X^2 + X T_c + X T_a + T_b \end{aligned}$$

Therefore, we get

$$\begin{pmatrix} T_a \\ T_b \\ T_c \end{pmatrix} = X^2 \begin{pmatrix} 1-X^2 & -X^2 & -X \\ -X & 1-X^2 & -X^3 \\ -X & -1 & 1-X \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$$

Next, we have

$$\begin{aligned} T(X) = T_d &= \begin{pmatrix} X & X^2 & X \end{pmatrix} \begin{pmatrix} T_a \\ T_b \\ T_c \end{pmatrix} \\ &= X^3 \begin{pmatrix} 1 & X & 1 \end{pmatrix} \begin{pmatrix} 1-X^2 & -X^2 & -X \\ -X & 1-X^2 & -X^3 \\ -X & -1 & 1-X \end{pmatrix}^{-1} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \end{aligned}$$

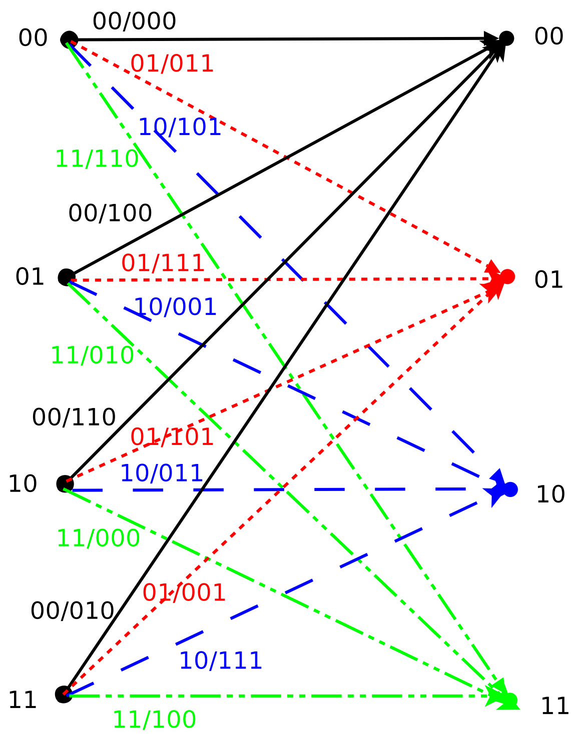



Figure 8: A trellis section for the rate $\frac{2}{3}$ convolutional code

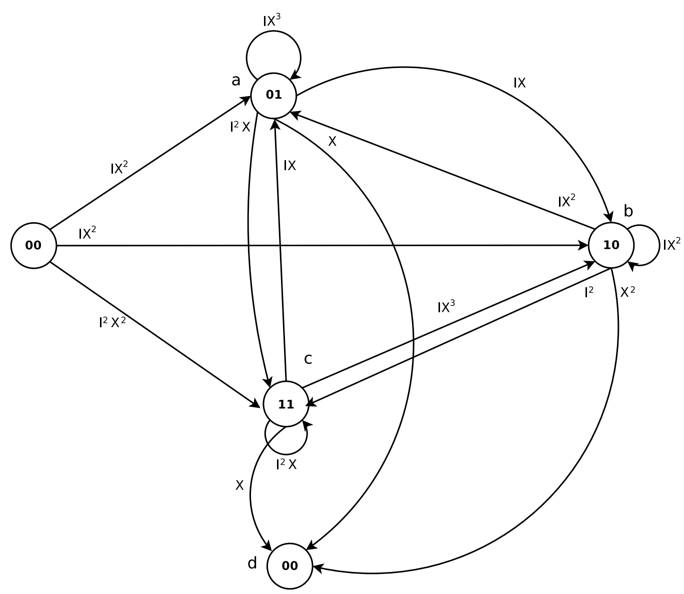



Figure 9: State diagram for the rate $\frac{2}{3}$ convolutional code. The exponent of $I$ indicates the input weight while the exponent of $X$ indicates the output weight

We now use the symbolic toolbox in MATLAB to simplify and we get,

$$T(X) = X^3\left(\frac{X^6 - 2X^5 - 3X^4 + 5X^3 + 3X^2 - 4X - 3}{X^6 - 3X^4 + 4X^2 + X - 1}\right)$$

From the expression, we observe that $d_{free} = 3$ for this code. This can be shown rigorously by expanding the denominator in the form $(1-Y)^{-1} = 1 + Y + Y^2 + \ldots$ and observing that the 3 in the numerator mutliplying the 1 in the expansion of $(1-Y)^{-1}$ results in a constant term and hence the lowest power of $X$ in a power series expansion of $T(X)$ is 3. We will now find the number of error events of weight 5. Firstly, set $Y = X + 4X^2 - 3X^4 + X^6$. Then,

$$\begin{aligned} T(X) &= X^3(3 + 4X - 3X^2 - 5X^3 + 3X^4 + 2X^5 - X^6)(1-Y)^{-1} \\ &= X^3(3 + 4X - 3X^2 - 5X^3 + 3X^4 + 2X^5 - X^6)(1 + Y + Y^2 + Y^3 + \ldots) \end{aligned}$$

Now equating coefficients, we have,

$$\begin{aligned} \text{Coefficient of } X^5 \text{ in } T(X) &= 3 \times (4+1) + 4 \times 1 - 3 \times 1 \\ &= 15 + 4 - 3 = 16 \end{aligned}$$

Therefore, the number of error events of weight 5 is 16.

## Problem 5

### Part a

We implement the Viterbi algorithm in MATLAB (code at the bottom) and the state metrics at various instants of time are given by

| State | $t=0$ | $t=1$ | $t=2$ | $t=3$ | $t=4$ | $t=5$ | $t=6$ | $t=7$ |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 1 | -0.3 | 0.7 | 1 | 3 | 4 | 6.9 |
| 01 | - | - | -1.3 | 0.9 | 3.8 | 3.6 | 7.8 | - |
| 10 | - | -1 | 2.3 | -0.3 | 1.2 | 5.8 | - | - |
| 11 | - | - | -0.7 | 3.7 | 3.6 | 3.6 | - | - |

We also keep track of the "winner" (maximum among two incoming branches) to each state in the matrix $\mathbf{W}$. Also, for each state $s$ at time $t$, we keep track of the state at time $t-1$ from where we are most likely to get to $s$ in the matrix **prev_state**. With the data in $\mathbf{W}$ and **prev_state**, we see that the most likely sequence of state transitions are

$$00 \rightarrow 00 \rightarrow 10 \rightarrow 11 \rightarrow 01 \rightarrow 10 \rightarrow 01 \rightarrow 00$$

which means that the 5 input bits are most likely to be $\{0, 1, 0, 0, 0\}$.

### Part b

In this case, we quantize the decisions (to $\pm 1$) and then implement the Viterbi algorithm on the quantized observations. The main difference with the earlier part is that, the decision metric (Hamming distance - refer to Problem 7) is to be **minimized**. Therefore, the "winner"among incoming branches will be that branch which results in code bits that are closer (Hamming distance) to the received bits. The state metrics at different instants are given by

| State | $t=0$ | $t=1$ | $t=2$ | $t=3$ | $t=4$ | $t=5$ | $t=6$ | $t=7$ |
|---|---|---|---|---|---|---|---|---|
| 00 | 0 | 2 | 2 | 4 | 6 | 6 | 8 | 8 |
| 01 | - | - | 4 | 6 | 6 | 6 | 6 | - |
| 10 | - | 2 | 6 | 4 | 4 | 6 | - | - |
| 11 | - | - | 4 | 4 | 6 | 6 | - | - |

As in the previous part, we keep track of the winner and information about the previous state in **W** and **prev_state**. The most likely sequence of state transitions with the quantized decisions is,

$$00 \to 00 \to 00 \to 00 \to 10 \to 11 \to 01 \to 00$$

which implies that the bits are most likely to be $\{0,0,1,1,0\}$. We see that 3 out of the 5 bits which are estimated don't agree with the estimates made using soft decisions. This occurs because some of the received values are small (in magnitude) and don't give us much confidence to choose either way. However, quantizing these values discards the information about the relative confidence we have in various received symbols and hence, we are prone to making more errors.

```
% This program implements the Viterbi algorithm to recover the ML input
% bit sequence for the rate 1/2 convolutional code

clear
clc

% Defining the received values - Each row represents values received at a
% particular time
rcvd_values = [-0.5 1.5;-0.5 -0.8;1.2 -0.2;0.2 0.1;1 1;-0.5 1.5;-1.1 2;];

% We now store the symbols that would be output for different transitions
% We denote the symbol when the transition is from state i to state j by
% a_ij. Note that 0 maps to 1 and 1 maps to -1.
a_11 = [1;1];
a_13 = [-1;-1];
a_21 = a_13;
a_23 = a_11;

a_32 = [-1;1];
a_34 = [1;-1];
a_42 = [1;-1];
a_44 = [-1;1];

% Let S be the matrix which stores the "state" values at different instants
% of time. Let W denote the winner to each state. Let prev_state(state,t) denote
% the state t - 1 from where we are more likely to have come to state
% "state" at time t
S = zeros(4,7);
W = zeros(4,7);
prev_state = zeros(4,7);

% For the first two time instants, we update the state values manually

% t = 1 : Only states 1 and 3 feasible
temp = rcvd_values(1,:);
S(1,1) = temp*a_11;
```

```
S(3,1) = temp*a_13;

% t = 2 : All states are feasible, but there is nothing to choose from
temp = rcvd_values(2,:);
S(1,2) = S(1,1) + temp*a_11;
prev_state(1,2) = 1;
S(2,2) = S(3,1) + temp*a_32;
prev_state(2,2) = 3;
S(3,2) = S(1,1) + temp*a_13;
prev_state(3,2) = 1;
S(4,2) = S(3,1) + temp*a_34;
prev_state(4,2) = 3;
for t = 3:5

    % Now we choose the winner based on the maximum of two incoming
    % branches into the trellis. We store the "winner" in the matrix W and
    % the most likely previous state (for each state) at the previous time
    % in prev_state
    temp = rcvd_values(t,:);
    [S(1,t),W(1,t)] = max([S(1,t-1) + temp*a_11, S(2,t-1) + temp*a_21]);
    if W(1,t) == 1
        prev_state(1,t) = 1;
    else
        prev_state(1,t) = 2;
    end

    [S(2,t),W(2,t)] = max([S(4,t-1) + temp*a_42, S(3,t-1) + temp*a_32]);
    if W(2,t) == 1
        prev_state(2,t) = 4;
    else
        prev_state(2,t) = 3;
    end

    [S(3,t),W(3,t)] = max([S(2,t-1) + temp*a_23, S(1,t-1) + temp*a_13]);
    if W(3,t) == 1
        prev_state(3,t) = 2;
    else
        prev_state(3,t) = 1;
    end

    [S(4,t),W(4,t)] = max([S(3,t-1) + temp*a_34, S(4,t-1) + temp*a_44]);
    if W(4,t) == 1
        prev_state(4,t) = 3;
    else
        prev_state(4,t) = 4;
    end
end

% Once again we handcraft the values taken by S for the last two time
% instants

% t = 6
temp = rcvd_values(6,:);
```

```
[S(1,6),W(1,6)] = max([S(1,5) + temp*a_11,S(2,5) + temp*a_21]);
if W(1,6) == 1
    prev_state(1,6) = 1;
else
    prev_state(1,6) = 2;
end
[S(2,6),W(2,6)] = max([S(4,5) + temp*a_42,S(3,5) + temp*a_32]);
if W(2,6) == 1
    prev_state(2,6) = 4;
else
    prev_state(2,6) = 3;
end
% t = 7
temp = rcvd_values(7,:);
[S(1,7),W(1,7)] = max([S(1,6) + temp*a_11,S(2,6) + temp*a_21]);
if W(1,7) == 1
    prev_state(1,7) = 1;
else
    prev_state(1,7) = 2;
end
% We now manually backtrack using the data in W and prev_state to recover
% the input bits

% ----------------------------------------------------------------------------

% -------------------- Viterbi algorithm with hard decisions --------------
clear S W prev_state

% The notation and symbols used are the same as in the previous case except
% that we work with hard decisions
rcvd_values_hard = -sign(rcvd_values);
S = zeros(4,7);
W = zeros(4,7);
prev_state = zeros(4,7);

% t = 1: Only states 1 & 3 are feasible
temp = rcvd_values_hard(1,:);
S(1,1) = sum(abs(temp' - a_11));
S(3,1) = sum(abs(temp' - a_13));

% t = 2: All states are feasible but only 1 incoming branch to each state
temp = rcvd_values_hard(2,:);
S(1,2) = S(1,1) + sum(abs(temp' - a_11));
S(2,2) = S(3,1) + sum(abs(temp' - a_32));
S(3,2) = S(1,1) + sum(abs(temp' - a_13));
S(4,2) = S(3,1) + sum(abs(temp' - a_34));

% t = 3 to 5 : Each state has two incoming branches from which we need to
% choose. However note that we have to choose the MINIMUM of the incoming
% branches

for t = 3:5
    temp = rcvd_values_hard(t,:);
```

```
    [S(1,t),W(1,t)] = min([S(1,t-1) + sum(abs(temp' - a_11)), S(2,t-1) + ...
        sum(abs(temp' - a_21))]);
    if W(1,t) == 1
        prev_state(1,t) = 1;
    else
        prev_state(1,t) = 2;
    end
    [S(2,t),W(2,t)] = min([S(4,t-1) + sum(abs(temp' - a_42)), S(3,t-1) + ...
        sum(abs(temp' - a_32))]);
    if W(2,t) == 1
        prev_state(2,t) = 4;
    else
        prev_state(2,t) = 3;
    end
    [S(3,t),W(3,t)] = min([S(2,t-1) + sum(abs(temp' - a_23)), S(1,t-1) + ...
        sum(abs(temp' - a_13))]);
    if W(3,t) == 1
        prev_state(3,t) = 2;
    else
        prev_state(3,t) = 1;
    end
    [S(4,t),W(4,t)] = min([S(3,t-1) + sum(abs(temp' - a_34)), S(4,t-1) + ...
        sum(abs(temp' - a_44))]);
    if W(4,t) == 1
        prev_state(4,t) = 3;
    else
        prev_state(4,t) = 4;
    end
end

% t = 6 : Only states 1 & 2 are possible
temp = rcvd_values_hard(6,:);
[S(1,6),W(1,6)] = min([S(1,5) + sum(abs(temp' - a_11)), S(2,5) + ...
    sum(abs(temp' - a_21))]);
if W(1,6) == 1
    prev_state(1,6) = 1;
else
    prev_state(1,6) = 2;
end
[S(2,6),W(2,6)] = min([S(4,5) + sum(abs(temp' - a_42)), S(3,5) + ...
    sum(abs(temp' - a_32))]);% This program implements the Viterbi algorithm to recover the
% bit sequence for the rate 1/2 convolutional code

clear
clc

% Defining the received values - Each row represents values received at a
% particular time
rcvd_values = [-0.5 1.5;-0.5 -0.8;1.2 -0.2;0.2 0.1;1 1;-0.5 1.5;-1.1 2;];

% We now store the symbols that would be output for different transitions
% We denote the symbol when the transition is from state i to state j by
% a_ij. Note that 0 maps to 1 and 1 maps to -1.
```

```
a_11 = [1;1];
a_13 = [-1;-1];
a_21 = a_13;
a_23 = a_11;

a_32 = [-1;1];
a_34 = [1;-1];
a_42 = [1;-1];
a_44 = [-1;1];

% Let S be the matrix which stores the "state" values at different instants
% of time. Let W denote the winner to each state. Let prev_state(state,t) denote
% the state t - 1 from where we are more likely to have come to state
% "state" at time t
S = zeros(4,7);
W = zeros(4,7);
prev_state = zeros(4,7);

% For the first two time instants, we update the state values manually

% t = 1 : Only states 1 and 3 feasible
temp = rcvd_values(1,:);
S(1,1) = temp*a_11;
S(3,1) = temp*a_13;

% t = 2 : All states are feasible, but there is nothing to choose from
temp = rcvd_values(2,:);
S(1,2) = S(1,1) + temp*a_11;
prev_state(1,2) = 1;
S(2,2) = S(3,1) + temp*a_32;
prev_state(2,2) = 3;
S(3,2) = S(1,1) + temp*a_13;
prev_state(3,2) = 1;
S(4,2) = S(3,1) + temp*a_34;
prev_state(4,2) = 3;
for t = 3:5

    % Now we choose the winner based on the maximum of two incoming
    % branches into the trellis. We store the "winner" in the matrix W and
    % the most likely previous state (for each state) at the previous time
    % in prev_state
    temp = rcvd_values(t,:);
    [S(1,t),W(1,t)] = max([S(1,t-1) + temp*a_11, S(2,t-1) + temp*a_21]);
    if W(1,t) == 1
        prev_state(1,t) = 1;
    else
        prev_state(1,t) = 2;
    end

    [S(2,t),W(2,t)] = max([S(4,t-1) + temp*a_42, S(3,t-1) + temp*a_32]);
    if W(2,t) == 1
        prev_state(2,t) = 4;
    else
```

```
        prev_state(2,t) = 3;
    end

    [S(3,t),W(3,t)] = max([S(2,t-1) + temp*a_23, S(1,t-1) + temp*a_13]);
    if W(3,t) == 1
        prev_state(3,t) = 2;
    else
        prev_state(3,t) = 1;
    end

    [S(4,t),W(4,t)] = max([S(3,t-1) + temp*a_34, S(4,t-1) + temp*a_44]);
    if W(4,t) == 1
        prev_state(4,t) = 3;
    else
        prev_state(4,t) = 4;
    end
end

% Once again we handcraft the values taken by S for the last two time
% instants

% t = 6
temp = rcvd_values(6,:);
[S(1,6),W(1,6)] = max([S(1,5) + temp*a_11,S(2,5) + temp*a_21]);
if W(1,6) == 1
    prev_state(1,6) = 1;
else
    prev_state(1,6) = 2;
end
[S(2,6),W(2,6)] = max([S(4,5) + temp*a_42,S(3,5) + temp*a_32]);
if W(2,6) == 1
    prev_state(2,6) = 4;
else
    prev_state(2,6) = 3;
end
% t = 7
temp = rcvd_values(7,:);
[S(1,7),W(1,7)] = max([S(1,6) + temp*a_11,S(2,6) + temp*a_21]);
if W(1,7) == 1
    prev_state(1,7) = 1;
else
    prev_state(1,7) = 2;
end
% We now manually backtrack using the data in W and prev_state to recover
% the input bits

% ------------------------------------------------------------------------

% -------------------- Viterbi algorithm with hard decisions --------------
clear S W prev_state

% The notation and symbols used are the same as in the previous case except
% that we work with hard decisions
```

```
rcvd_values_hard = -sign(rcvd_values);
S = zeros(4,7);
W = zeros(4,7);
prev_state = zeros(4,7);

% t = 1: Only states 1 & 3 are feasible
temp = rcvd_values_hard(1,:);
S(1,1) = sum(abs(temp' - a_11));
S(3,1) = sum(abs(temp' - a_13));

% t = 2: All states are feasible but only 1 incoming branch to each state
temp = rcvd_values_hard(2,:);
S(1,2) = S(1,1) + sum(abs(temp' - a_11));
S(2,2) = S(3,1) + sum(abs(temp' - a_32));
S(3,2) = S(1,1) + sum(abs(temp' - a_13));
S(4,2) = S(3,1) + sum(abs(temp' - a_34));

% t = 3 to 5 : Each state has two incoming branches from which we need to
% choose. However note that we have to choose the MINIMUM of the incoming
% branches

for t = 3:5
    temp = rcvd_values_hard(t,:);
    [S(1,t),W(1,t)] = min([S(1,t-1) + sum(abs(temp' - a_11)), S(2,t-1) + ...
        sum(abs(temp' - a_21))]);
    if W(1,t) == 1
        prev_state(1,t) = 1;
    else
        prev_state(1,t) = 2;
    end
    [S(2,t),W(2,t)] = min([S(4,t-1) + sum(abs(temp' - a_42)), S(3,t-1) + ...
        sum(abs(temp' - a_32))]);
    if W(2,t) == 1
        prev_state(2,t) = 4;
    else
        prev_state(2,t) = 3;
    end
    [S(3,t),W(3,t)] = min([S(2,t-1) + sum(abs(temp' - a_23)), S(1,t-1) + ...
        sum(abs(temp' - a_13))]);
    if W(3,t) == 1
        prev_state(3,t) = 2;
    else
        prev_state(3,t) = 1;
    end
    [S(4,t),W(4,t)] = min([S(3,t-1) + sum(abs(temp' - a_34)), S(4,t-1) + ...
        sum(abs(temp' - a_44))]);
    if W(4,t) == 1
        prev_state(4,t) = 3;
    else
        prev_state(4,t) = 4;
    end
end
```

```
% t = 6 : Only states 1 & 2 are possible
temp = rcvd_values_hard(6,:);
[S(1,6),W(1,6)] = min([S(1,5) + sum(abs(temp' - a_11)), S(2,5) + ...
    sum(abs(temp' - a_21))]);
if W(1,6) == 1
    prev_state(1,6) = 1;
else
    prev_state(1,6) = 2;
end
[S(2,6),W(2,6)] = min([S(4,5) + sum(abs(temp' - a_42)), S(3,5) + ...
    sum(abs(temp' - a_32))]);
if W(2,6) == 1
    prev_state(2,6) = 4;
else
    prev_state(2,6) = 3;
end
% t = 7
temp = rcvd_values_hard(7,:);
[S(1,7),W(1,7)] = min([S(1,6) + sum(abs(temp' - a_11)), S(2,6) + ...
    sum(abs(temp' - a_21))]);
if W(1,7) == 1
    prev_state(1,7) = 1;
else
    prev_state(1,7) = 2;
end

% Once again we manually backtrack to figure out the bits

if W(2,6) == 1
    prev_state(2,6) = 4;
else
    prev_state(2,6) = 3;
end
% t = 7
temp = rcvd_values_hard(7,:);
[S(1,7),W(1,7)] = min([S(1,6) + sum(abs(temp' - a_11)), S(2,6) + ...
    sum(abs(temp' - a_21))]);
if W(1,7) == 1
    prev_state(1,7) = 1;
else
    prev_state(1,7) = 2;
end

% Once again we manually backtrack to figure out the bits
```

## Problem 6

### Part a

The transfer function bound tells us that the probability of bit-error for the rate $\frac{1}{2}$ nonrecursive, nonsystematic convolutional code with generator [7,5] is upper bounded as

$$P_e \leq \frac{1}{2} \frac{e^{-\frac{5E_b}{2N_0}}}{\left(1 - 2e^{-\frac{5E_b}{2N_0}}\right)^2}$$

The probability of bit-error for uncoded BPSK is given by

$$P_e = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

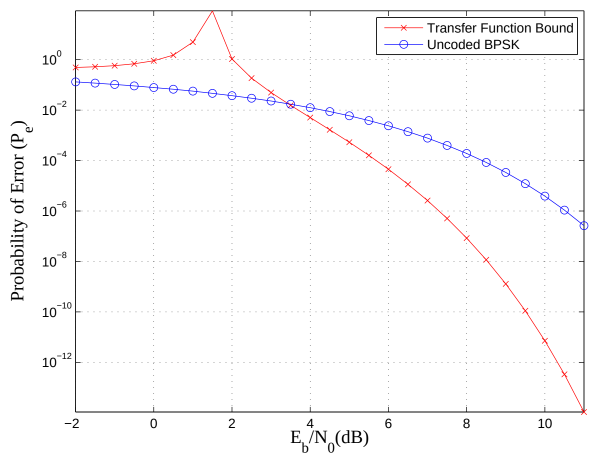



Figure 10: A plot of transfer function bound for the rate $\frac{1}{2}$ convolutional code and the performance of uncoded BPSK

### Part b

By eyeballing the figure, we see that a BER of $10^{-5}$ is attained by uncoded BPSK at $\frac{E_b}{N_0} = 10.5$ dB and by the convolutional code at $\frac{E_b}{N_0} = 7.25$ dB. Therefore, the coding gain is about **3.25 dB**.

### Part c

Let us denote the spectral efficiency of the overall system by $r$. It is given by

$$r = \underbrace{2}_{\text{Spectral efficiency of BPSK}} \times \underbrace{\frac{1}{2}}_{\text{Code Rate}} = 1$$

For unrestricted input, the minimum $\frac{E_b}{N_0}$ is given by

$$\frac{E_b}{N_0} = \frac{2^1 - 1}{1} = 1 = 0 \text{ dB}$$

For BPSK input, with Shannon Codes, the minimum $\frac{E_b}{N_0}$ to achieve a spectral efficiency of 1 is about 0.22 dB (by eyeballing the figure below). Therefore this code is about

1. 7.25 dB away from the Shannon limit for unrestricted input over AWGN
2. 7.03 dB away from the Shannon limit for BPSK over AWGN

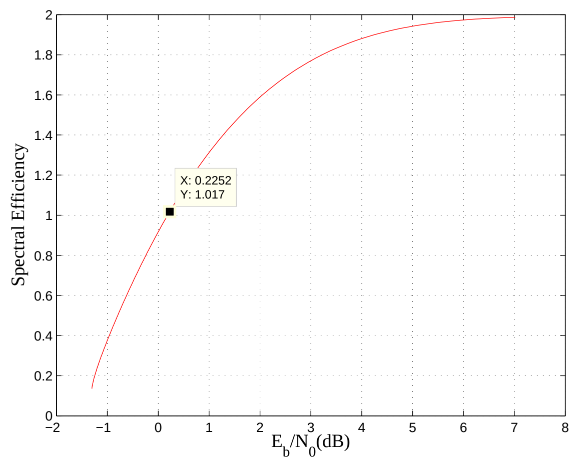



Figure 11: Spectral Efficiency of BPSK input with optimal random codes over an AWGN channel

For the given spectral efficiency, we see that we don't pay a great price by restricting ourselves to BPSK instead of optimal input. However, the performance of this code leaves a lot of room of improvement.

## Problem 7

### Part a

Let us assume that the codeword $\mathbf{c}$ has $N$ bits given by $(c[1], c[2], \ldots, c[n])$. Then, the ML estimate of the codeword $\mathbf{c}$ is given by

$$\hat{\mathbf{c}} = \arg\max_{\mathbf{c}} \sum_{k=1}^{N} \log\, p(r[k]|c[k])$$

We define the error vector $\mathbf{e} = (e[1], e[2], \ldots, e[N])$ as follows,

$$\begin{aligned} e[k] &= 0 \qquad & r[k] = c[k] \\ &= 1 \qquad & r[k] \neq c[k] \end{aligned}$$

Therefore, $p(r[k]|c[k])$ can be written as

$$p(r[k]|c[k]) = (1-p)^{1-e[k]}\, p^{e[k]}$$

where $p$ is the crossover probability. We have,

$$\begin{aligned}\sum_{k=1}^{N} \log_2 p(r[k]|c[k]) &= \sum_{k=1}^{N} (1-e[k])\log_2(1-p) + e[k]\log_2 p \\ &= \sum_{k=1}^{N} \left(\log_2(1-p) - e[k]\log_2\frac{1-p}{p}\right) \\ &= N\log_2(1-p) - \log_2\frac{1-p}{p} d_H(\mathbf{r},\mathbf{c})\end{aligned}$$

where $d_H(\mathbf{r},\mathbf{c}) = \sum_{k=1}^{N} e[k]$ denotes the Hamming distance between $\mathbf{c}$ and $\mathbf{r}$. We note that $\log_2\frac{1-p}{p} > 0$ because $p < \frac{1}{2}$. We have that,

$$\begin{aligned}\hat{\mathbf{c}} &= \arg\max_{\mathbf{c}} \sum_{k=1}^{N} \log\, p(r[k]|c[k]) \\ &= \arg\max_{\mathbf{c}} \left(N\log_2(1-p) - d_H(\mathbf{r},\mathbf{c})\log_2\frac{1-p}{p}\right) \\ &= \arg\max_{\mathbf{c}} \left(-d_H(\mathbf{r},\mathbf{c})\log_2\frac{1-p}{p}\right) \\ &= \arg\min_{\mathbf{c}} d_H(\mathbf{r},\mathbf{c})\end{aligned}$$

Therefore, the ML estimate is the codeword that is closest in terms of the Hamming distance to the received vector.

### Part b

For the $k^{th}$ bit $c[k]$ transmitted using BPSK over AWGN, the channel model is

$$z[k] = \sqrt{E_s}(-1)^{c[k]} + n[k]$$

where $E_s = RE_b$ is the energy per symbol and $n[k]$ is distributed as $N(0, \frac{N_0}{2})$. For this channel model, the probability of error is the same for a 0 or a 1 being transmitted and is given by $P_e = Q\left(\sqrt{\frac{2E_s}{N_0}}\right) = Q\left(\sqrt{\frac{2RE_b}{N_0}}\right)$. Therefore, the crossover probability is $Q\left(\sqrt{\frac{2RE_b}{N_0}}\right)$ for BPSK over AWGN with hard decisions.

## Problem 7.8

## Part a

Let us assume that we are trying to choose between codewords $\mathbf{u}$ and $\mathbf{v}$ where $d_H(\mathbf{u},\mathbf{v}) = x$ when the output of the BSC is $\mathbf{r}$. From the result of the previous problem, we know that we will choose the codeword that is closer in Hamming distance to $\mathbf{r}$. Assume for the sake of argument that $\mathbf{u}$ was transmitted. We need to make **atleast** $\lceil \frac{x}{2} \rceil$ errors to decode $\mathbf{r}$ as $\mathbf{v}$. If the crossover probability of the BSC is $p$, the probability of this happening is

$$q(x) = \sum_{i=\lceil \frac{x}{2} \rceil}^{x} \binom{x}{i} p^i (1-p)^{x-i}$$

## Part b

Since $i \geq \lceil \frac{x}{2} \rceil$, we have $i - \frac{x}{2} \geq 0$. Also, since $p < \frac{1}{2}$, we have $\log \frac{p}{1-p} \leq 0$. Therefore,

$$\begin{aligned}
(i - \frac{x}{2})\log\left(\frac{p}{1-p}\right) &\leq 0 \\
(i - \frac{x}{2})\log p + (\frac{x}{2} - i)\log(1-p) &\leq 0 \\
i\log p + (x-i)\log(1-p) &\leq \frac{x}{2}\log p + \frac{x}{2}\log(1-p) \\
p^i(1-p)^{x-i} &\leq \left(\sqrt{p(1-p)}\right)^x
\end{aligned}$$

We know that

$$\sum_{i=0}^{x} \binom{x}{i} = 2^x$$

Therefore,

$$\begin{aligned}
q(x) &= \sum_{i=\lceil \frac{x}{2} \rceil}^{x} \binom{x}{i} p^i (1-p)^{x-i} \\
&\leq \left(\sqrt{p(1-p)}\right)^x \sum_{i=\lceil \frac{x}{2} \rceil}^{x} \binom{x}{i} \\
&\leq \left(\sqrt{p(1-p)}\right)^x \times 2^x = \left(2\sqrt{p(1-p)}\right)^x
\end{aligned}$$

### Part c

From equation (7.12) in the textbook, we see that

$$P_e \leq \sum_{i=1}^{\infty} \sum_{x=1}^{\infty} iA(i,x)q(x)$$

where $q(x)$ is the probability that a codeword with weight $x$ is more likely than the all-zero codeword (which is assumed to be transmitted). From the previous part, we have

$$q(x) \leq \left(2\sqrt{(1-p)p}\right)^x$$

By definition, we have

$$T(I,X) = \sum_i \sum_x iA(i,x) I^i X^x$$

From this, it follows that

$$\frac{\partial T(I,X)}{\partial I} = \sum_i \sum_x iA(i,x)I^{i-1}X^x$$

Therefore,

$$P_e \leq \left.\frac{\partial T(I,X)}{\partial I}\right|_{I=1, X=2\sqrt{p(1-p)}}$$

Therefore, we have,

$$P_e \leq \left.\frac{X^5}{(1-2X)^2}\right|_{X=2\sqrt{p(1-p)}}$$

where $p = Q\left(\sqrt{\frac{2RE_b}{N_0}}\right)$

By eyeballing the curves, we see that a BER of $10^{-6}$ is attained by the soft decision decoding scheme at $\frac{E_b}{N_0} \approx 7.25$ dB and the corresponding value for the hard decision decoding scheme is 9.85 dB and the loss in performance due to hard decisions is about **2.6 dB**.

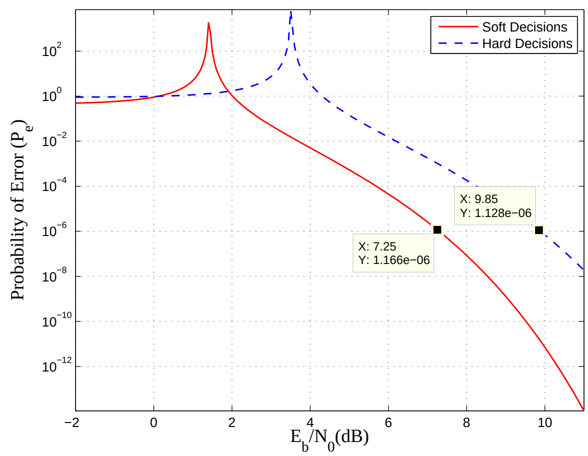



Figure 12: Performance comparison of the rate $\frac{1}{2}$ [7,5] nonrecursive, nonsystematic convolutional code with soft and hard decisions. By eyeballing the curves, we see that we lose about 2.6 dB by making hard decisions

### Part d

Approximating the probability of error by the first term in the union bound, we have,

$$P_e \approx q(d_{free}) \sum_i iA(i, d_{free})$$

As noted in the textbook

$$\sum_i iA(i, d_{free}) = 1$$

Therefore, $P_e \approx q(d_{free})$. In this case, $d_{free} = 5$ and from part (a), we have

$$P_e = \sum_{i=3}^{5} \binom{5}{i} p^i (1-p)^{5-i}$$

where $p = Q\left(\sqrt{\frac{E_b}{N_0}}\right) \approx \frac{1}{2}e^{-\frac{E_b}{2N_0}}$. For moderately high SNR, the term that dominates the summation will the one with the smallest power of $p$ and we can approximate $P_e$ further as

$$\begin{aligned}
P_e &\approx \begin{pmatrix} 5 \\ 3 \end{pmatrix} p^3(1-p)^2 \\
&\approx \begin{pmatrix} 5 \\ 3 \end{pmatrix} p^3 \qquad \because 1-p \approx 1 \\
&\approx \frac{\begin{pmatrix} 5 \\ 3 \end{pmatrix}}{2^3} e^{-\frac{3E_b}{2N_0}} \approx \frac{5}{4} \times e^{-\frac{3E_b}{2N_0}}
\end{aligned}$$

The coefficient of $\frac{E_b}{N_0}$ in the exponent is $\frac{3}{2}$ in the hard decision case and is $\frac{5}{2}$ in the soft decoding case. Therefore, the degradation due to hard decisions is

$$\text{Degradation} = 10\log_{10}\frac{\left(\frac{5}{2}\right)}{\left(\frac{3}{2}\right)} = 2.21\text{dB}$$

## Problem 7.9

### Part a

$$V = \log\frac{p(r|1)}{p(r|0)}$$

For the continuous case,we have,

$$\begin{aligned}
\mathbb{E}[e^{sV}|0] &= \int e^{sV} p(r|0)dr \\
&= \int \exp\left(s\log\frac{p(r|1)}{p(r|0)}\right) p(r|0)dr \\
&= \int (p(r|1))^s (p(r|0))^{1-s} dr = b_0
\end{aligned}$$

If $s = \frac{1}{2}$,

$$b_0 = \int \sqrt{p(r|0)p(r|1)}dr$$

For the discrete case, we have,

$$\begin{aligned}
\mathbb{E}[e^{sV}|0] &= \sum_r \exp\left(s\log\frac{p(r|1)}{p(r|0)}\right) p(r|0) \\
&= \sum_r (p(r|1))^s (p(r|0))^{1-s} = b_0
\end{aligned}$$

If $s = \frac{1}{2}$,

$$b_0 = \sum_r \sqrt{p(r|0)p(r|1)}$$

**Part b**

Let us define $M(s) = \log E[(e^{sV_i}|0)]$ for any $i \in [1, x]$. By the Chernoff Bound, we have,

$$P(V_1 + V_2 + \ldots + V_x > 0|0) \leq \mathbb{E}\left[e^{s(V_1+V_2+\ldots+V_x)}|0\right] = e^{xM(s)} \;\; \forall s > 0$$

We know from part (a) that, $M(\frac{1}{2}) = \log b_0$. We can set $s = \frac{1}{2}$ in the above equation to get

$$q(x) = P(V_1 + V_2 + \ldots + V_x > 0|0) \leq \exp(x\log b_0) \leq b_0^x$$

## Problem 7.10

Let us calculate the constant $b_0$ for a BSC($p$).

$$b_0 = \sqrt{p(0|0)p(0|1)} + \sqrt{p(1|0)p(1|1)} = 2\sqrt{p(1-p)}$$

Therefore, we have the Bhattacharya bound for the BSC to be

$$q(x) \leq \left(2\sqrt{p(1-p)}\right)^x$$

This is the best possible Chernoff bound for a BSC, in the sense that, the minimum of $M(s)$ over all $s$ occurs at $s = \frac{1}{2}$. Set $u = \log \frac{1-p}{p}$. We note that

$$M(s) = (1-p)e^{-su} + pe^{su}$$

Therefore,

$$\begin{aligned} \frac{M(s)}{2} &= \frac{(1-p)e^{-su} + pe^{su}}{2} \\ &= \frac{e^{-su+\log(1-p)} + e^{su+\log p}}{2} \end{aligned}$$

Jensen's inequality tells us that

$$\frac{e^x + e^y}{2} \geq e^{\frac{x+y}{2}}$$

Therefore,

$$\begin{aligned} \frac{M(s)}{2} &\geq e^{\frac{\log(1-p)-su+\log(p)+su}{2}} = e^{\frac{\log(1-p)+\log(p)}{2}} \\ &= e^{\log\sqrt{(1-p)p}} = \sqrt{(1-p)p} \end{aligned}$$

Therefore,

$$M(s) \geq 2\sqrt{(1-p)p}$$

and hence, the bound we had obtained was the best possible Chernoff bound.

## Problem 7.11

### Part a

The channel model is

$$z = \sqrt{E_s}(-1)^c + n$$

where $n \sim N(0,1)$. We have

$$\begin{aligned} p(z|0) &= \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{(z-E_S)^2}{2}\right) \\ p(z|1) &= \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{(z+E_S)^2}{2}\right) \end{aligned}$$

Therefore, the Bhattacharya bound parameter $b_0$ is given by

$$\begin{aligned} b_0 &= \int \sqrt{p(z|0)p(z|1)}dz \\ &= \int \frac{1}{\sqrt{2\pi}}\sqrt{\exp\left(-\frac{(z-E_S)^2+(z+E_S)}{2}\right)}dz \\ &= \int \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{z^2+E_s}{2}\right)dz \\ &= \exp\left(-\frac{E_S}{2}\right) \end{aligned}$$

In our case, $E_S = 10^{0.6} \approx 4 \Rightarrow b_0 = \exp(-2) = 0.135$.

### Part b

Let us define all the conditional probabilities in terms of $\gamma$ and $E_S$. We use the following properties:

$$\begin{aligned} P(N \geq x) &= Q(x) \\ P(N \geq -x) &= 1 - Q(x) \\ P(a \leq N < b) &= Q(a) - Q(b) \end{aligned}$$

$$\begin{array}{lclrl} P(r=3|0) & = & Q(\gamma-\sqrt{E_S}) & P(r=3|1) = & Q(\gamma+\sqrt{E_S}) \\ P(r=-3|0) & = & Q(\gamma+\sqrt{E_S}) & P(r=-3|1) = & Q(\gamma-\sqrt{E_S}) \\ P(r=1|0) & = & Q(-\sqrt{E_S})-Q(\gamma-\sqrt{E_S}) & P(r=1|1) = & Q(\sqrt{E_S})-Q(\gamma+\sqrt{E_S}) \\ P(r=-1|0) & = & Q(\sqrt{E_S})-Q(\gamma+\sqrt{E_S}) & P(r=-1|1) = & Q(-\sqrt{E_S})-Q(\gamma-\sqrt{E_S}) \end{array}$$

Substituting, we get

$$b_0 = 2\sqrt{Q(\gamma-\sqrt{E_S})Q(\gamma+\sqrt{E_S})}+2\sqrt{\left(Q(-\sqrt{E_S})-Q(\gamma-\sqrt{E_S})\right)\left(Q(\sqrt{E_S})-Q(\gamma+\sqrt{E_S})\right)}$$

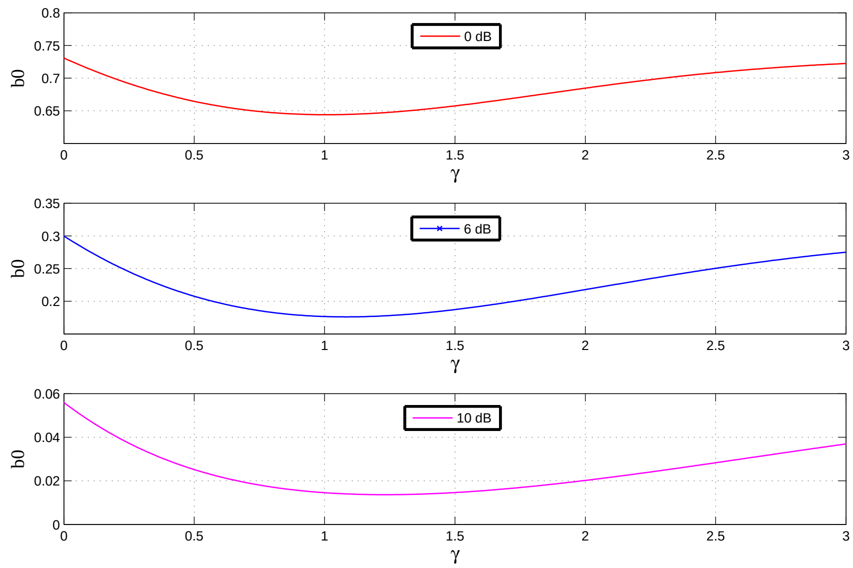



Figure 13: Plot of Bhattacharya Parameter against $\gamma$ for different values of SNR

Now we have $E_S = 1, 4, 10$ corresponding to 0 dB,6 dB and 10 dB and we plot the Bhattacharya parameter as a function of $\gamma$.

**Observations:**

The Bhattacharya parameter for unquantized observations is lower (at the same value of SNR) which is expected. The best choice of $\gamma$ as a function of SNR is as follows:

| SNR(dB) | $\gamma_{opt}$ |
|---:|---|
| 0 | 1.01 |
| 6 | 1.08 |
| 10 | 1.23 |

The optimal value of $\gamma$ increases with an increase in SNR.

## Problem 7.12

Throughout the problem $L(y)$ is taken to be discrete. We are given that $p(y|0) = p(-y|1) \Rightarrow p(-y|0) = p(y|1)$.

### Part a

$$\begin{aligned} L(-y) &= \log \frac{p(-y|0)}{p(-y|1)} \\ &= \log \frac{p(y|1)}{p(y|0)} \\ &= -L(y) \end{aligned}$$

Therefore, $L(y)$ is an odd function.

## Part b

Let $\mathcal{S}_l = \{y : L(y) = l\}$. From the previous part, we have

$$\{y : L(-y) = l\} = \{y : L(y) = -l\} = \mathcal{S}_{-l}$$

Using part (a) and a substitution of $-y$ by $y$, we also observe that

$$\sum_{\mathcal{S}_{-l}} p(-y|0) = \sum_{\mathcal{S}_l} p(y|0)$$

Now, we have

$$\begin{aligned} P(L(-y) = l|1) &= \sum_{\mathcal{S}_{-l}} p(y|1) = \sum_{\mathcal{S}_{-l}} p(-y|0) = \sum_{\mathcal{S}_l} p(y|0) \\ &= P(L(y) = l|0) \end{aligned}$$

## Part c

From part (a) and part (b), we have

$$\begin{aligned} P(L(-y) = l|1) &= P(L(y) = -l|1) && \text{Part (a)} \\ &= P(L(y) = l|0) && \text{Part (b)} \end{aligned}$$

## Part d

We have

$$q(-l) = \sum_{\mathcal{S}_{-l}} p(y|0)$$

Now, we have already seen that

$$\sum_{\mathcal{S}_{-l}} p(-y|0) = \sum_{\mathcal{S}_l} p(y|0)$$

Therefore,

$$\begin{aligned} q(-l) &= \sum_{\mathcal{S}_{-l}} p(y|0) = \sum_{\mathcal{S}_l} p(-y|0) \\ &= \sum_{\mathcal{S}_l} p(y|1) \end{aligned}$$

For all $y \,\epsilon\, \mathcal{S}_l$, $p(y|1) = e^{-l} p(y|0)$ by definition. Therefore,

$$\begin{aligned} q(-l) &= \sum_{\mathcal{S}_l} p(y|1) \\ &= e^{-l} \sum_{\mathcal{S}_l} p(y|0) \\ &= e^{-l} q(l) \end{aligned}$$

### Part e

For BPSK over AWGN, we have

$$
\begin{aligned}
L(y) &= \log\left(\frac{p(y|0)}{p(y|1)}\right) \\
&= \log\left(\frac{\exp\left(-\frac{(y-1)^2}{2\sigma^2}\right)}{\exp\left(-\frac{(y-1)^2}{2\sigma^2}\right)}\right) \\
&= \log\exp\left(\frac{2y}{\sigma^2}\right) = \frac{2y}{\sigma^2}
\end{aligned}
$$

We see that $L(y)$ is just a scaled version of $y$. The pdf of $y$, conditioned on either 0 being transmitted or a 1 being transmitted, is a Gaussian with a mean $\pm 1$ respectively and a variance of $\sigma^2$. Using the rule for pdf's of transformed random variables,

$$
\begin{aligned}
f_{L(y)}(l|0) &= \frac{\sigma^2}{2}\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{\left(\frac{\sigma^2 l}{2}-1\right)^2}{2\sigma^2}\right) \\
&= \frac{\sigma^2}{2}\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{\sigma^2 l^2}{8}-\frac{1}{2\sigma^2}\right)\exp\left(\frac{l}{2}\right) \\
&\triangleq q(l)
\end{aligned}
$$

Therefore, we have

$$
\begin{aligned}
\frac{q(l)}{q(-l)} &= \frac{\frac{\sigma^2}{2}\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{\sigma^2 l^2}{8}-\frac{1}{2\sigma^2}\right)}{\frac{\sigma^2}{2}\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{\sigma^2 l^2}{8}-\frac{1}{2\sigma^2}\right)} \times \frac{\exp\left(\frac{l}{2}\right)}{\exp\left(-\frac{l}{2}\right)} \\
&= \exp(l)
\end{aligned}
$$

Therefore, the consistency condition is satisfied.

### Part f

If the LLR conditioned on 0 being sent is modelled as $N(m, v^2)$, we have

$$
\begin{aligned}
q(l) &= \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(l-m)^2}{2v^2}\right) \\
q(-l)) &= \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{(l+m)^2}{2v^2}\right)
\end{aligned}
$$

From the consistency condition derived in part (e), we have

$$\frac{q(l)}{q(-l)} = e^l$$

$$\Rightarrow \frac{\exp\left(-\frac{(l-m)^2}{2v^2}\right)}{\exp\left(-\frac{(l+m)^2}{2v^2}\right)} = e^l$$

$$\Rightarrow \exp\left(\frac{4lm}{2v^2}\right) = e^l$$

$$\Rightarrow \frac{2m}{v^2} = 1$$

$$\Rightarrow v^2 = 2m$$

Note that for all values of $l > 0$, we have $\frac{q(l)}{q(-l)} = \exp(l) > 1$. Therefore, we have implicitly assumed that $m > 0$. To cover for the case where $m < 0$, we conclude that $v^2 = 2|m|$.

# Problem 13

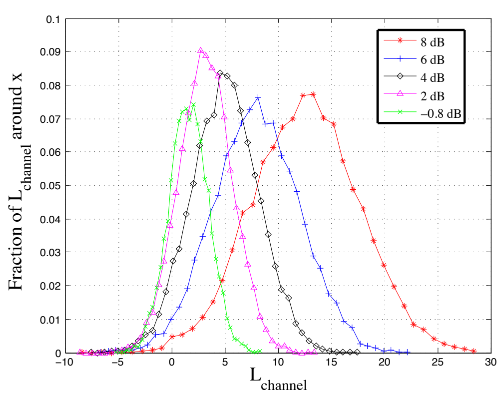
## Part a

**Histograms for a single packet**
We make the following observations from the graphs:

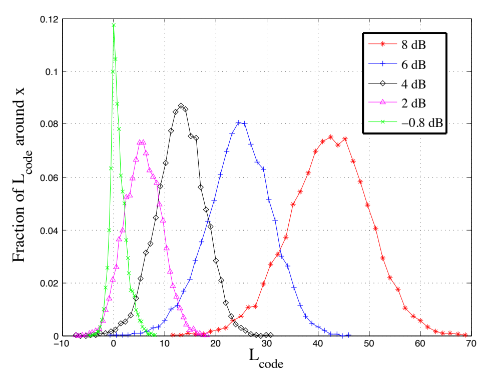
1. The mean values of $L_{channel}, L_{code}$ and $L_{out}$ increase with an increase in $\frac{E_b}{N_0}$.

2. The shape of the curves look reasonably Gaussian. The fact that we are considering these distributions over just one packet reduces the confidence in what we could conclude about the shape.

3. We also observe that the variance of the LLR increases along with the mean. We will make some observations on this fact in the next section.




(a) Distribution of $L_{channel}$ for varying $\frac{E_b}{N_0}$ in a packet




(b) Distribution of $L_{code}$ for varying $\frac{E_b}{N_0}$ in a packet

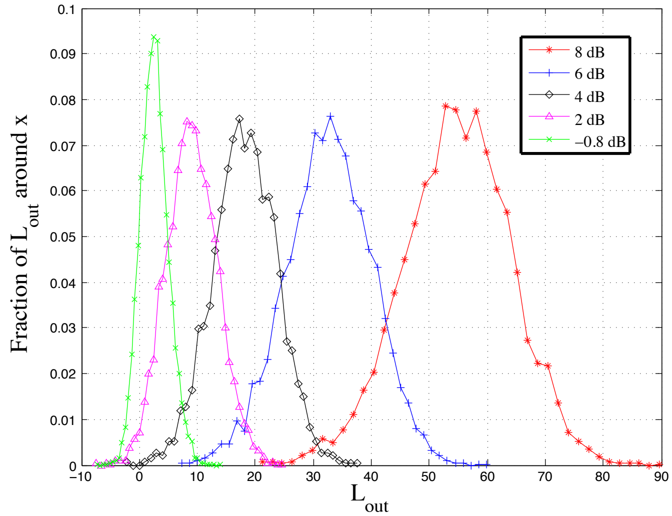
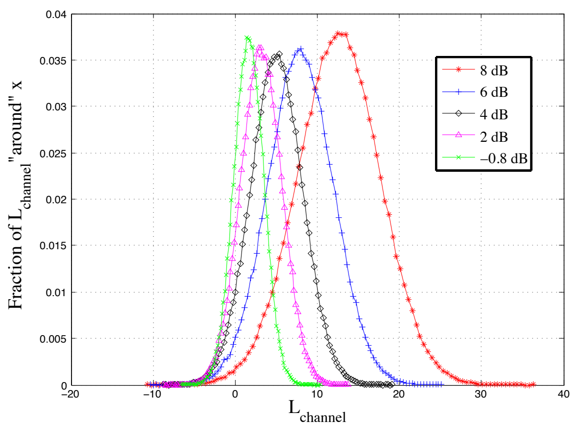


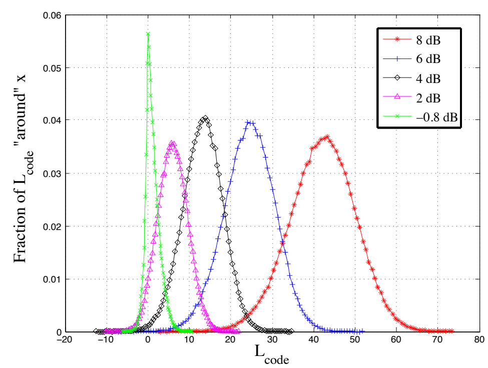

Figure 14: Distribution of $L_{out}$ for different values of $\frac{E_b}{N_0}$ in a packet

**Histograms over multiple packets**

We transmit 30 packets for different values of $\frac{E_b}{N_0}$ and plot the histograms of the various LLR's. By eyeballing the curves, we observe that $L_{channel}, L_{code}$ and $L_{out}$ are well approximated by Gaussians. We now verify that $E[L_{channel}]$ that we get from simulations matches the theoretical value of $2\frac{E_b}{N_0}$.




(a) Distribution of $L_{channel}$ for varying $\frac{E_b}{N_0}$ over 30 packets




(b) Distribution of $L_{code}$ for varying $\frac{E_b}{N_0}$ over 30 packets

| $\frac{E_b}{N_0}$ (dB) | Theoretical mean ($2\frac{E_b}{N_0}$) | Simulated mean |
|---|---|---|
| -0.8 | 1.66 | 1.45 |
| 2 | 3.16 | 3.08 |
| 4 | 5 | 5.36 |
| 6 | 8 | 7.95 |
| 8 | 12.6 | 12.55 |

We see that the values match quite well and we will not attempt to make similar comparisons for $L_{code}$ and $L_{out}$.

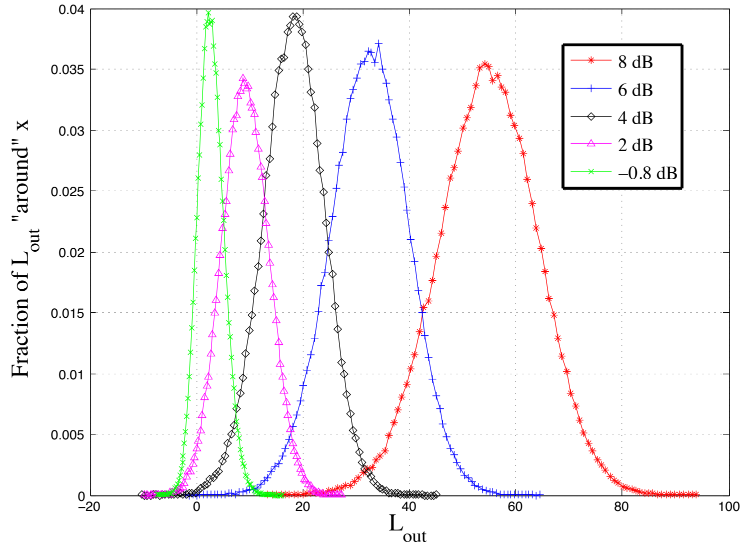



Figure 15: Distribution of $L_{out}$ for different values of $\frac{E_b}{N_0}$ over 30 packets

### Part b

We now compare the performance of the RSC Code with BCJR decoding against the uncoded case and figure out the coding gain. The probability of error for uncoded BPSK is given by

$$P_{e,uncoded} = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

To calculate the bit-error rate for RSC + BCJR, we use Monte Carlo simulations. The parameters used in the simulation are:

1. Rate $\frac{1}{2}$ [7,5] RSC Code.
2. Probability of 0 = Probability of 1 = $\frac{1}{2}$
3. Information bits per packet = 10000
4. Number of iterations is decided by a rule of thumb given by

$$\text{Number of bits observed} = 100 \times \frac{1}{\text{BER}}$$

By eyeballing the curve, we see that a BER of $10^{-4}$ requires an $\frac{E_b}{N_0}$ of roughly 5 dB. In this case, the spectral efficiency $r$ is given by

$$r = \underbrace{2}_{\text{Spectral efficiency of BPSK}} \times \underbrace{\frac{1}{2}}_{\text{Code Rate}} = 1$$

For a spectral efficiency of 1, the minimum required $\frac{E_b}{N_0}$ is given by

$$\left(\frac{E_b}{N_0}\right)_{min} = \frac{2^1 - 1}{1} = 1 = 0\text{dB}$$

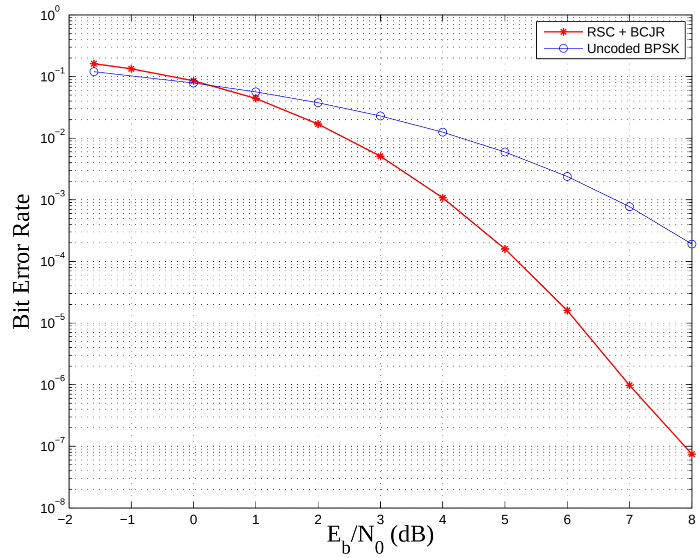



Figure 16: Performance comparison of RSC + BCJR with the transfer function bound

Therefore, we are about **5 dB away from capacity**. We also see that uncoded BPSK requires an $\frac{E_b}{N_0}$ of 8 dB for a BER of $10^{-4}$. Therefore, **the coding gain is roughly 3 dB**.

**RSC + BCJR vs. RSC + Viterbi**

We now move on to comparing the RSC + BCJR scheme with the RSC + Viterbi scheme. We use the transfer function bound as a measure of performance of the Viterbi algorithm. The transfer function for this rate $\frac{1}{2}$ [7,5] RSC Code is

$$T(I,X) = \frac{I^2X^4(X^2 - I^2X^2 + IX)}{1 - 2IX - X^2 + I^2X^2}$$

The transfer function bound tells us that the probability of error for Viterbi decoding is bounded as,

$$P_{e,Viterbi} \leq \frac{1}{2}\frac{\partial}{\partial I}T(I,X)\Big|_{I=1,X=b} \qquad b = \exp\left(\frac{-RE_b}{N_0}\right)$$

In our case,

$$\frac{\partial}{\partial I}T(I,X)\Big|_{I=1} = \frac{X^5}{(1-2X)^2}(3 - 6X + 2X^2)$$

To compare the performance of the BCJR algorithm and the Viterbi algorithm, we compare the results obtained via simulation to the transfer function bound.

Assuming that the transfer function bound approximates the performance of the Viterbi algorithm, we conclude that the RSC + BCJR performs better than the Viterbi algorithm. This is to be expected because the BCJR algorithm is just an efficient implementation of the maximum à posteriori algorithm which minimizes the probability of error.

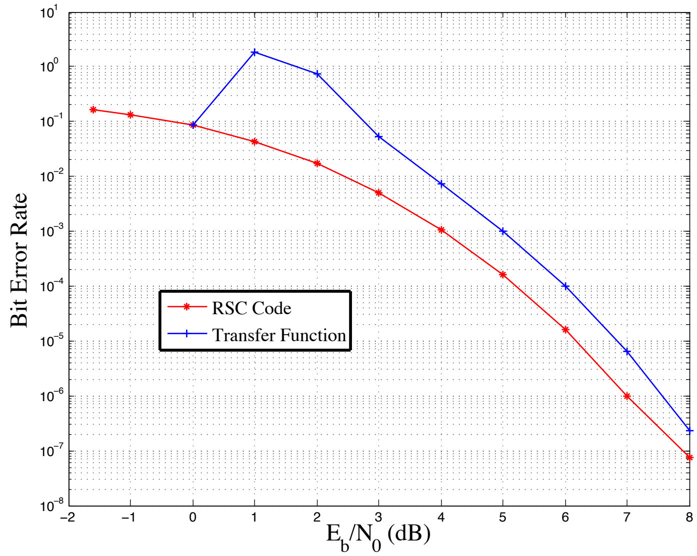



Figure 17: Performance comparison of RSC + BCJR with uncoded BPSK

### Part d

This part gives a sneak preview of what transpires in iterative decoding using serially concatenated codes. The outer decoder has no knowledge of the channel and can only operate with the LLR's passed onto it by the inner decoder. From part (a), we know that the statistics of the LLR's passed on by the inner decoder are well modelled by a Gaussian. Therefore, this part essentially asks for the statistics of $L_{code}$ for the outer decoder given the statistics of the $L_{in}$.

**Mean and variance of $L_{code}$**
We use Monte Carlo simulations once again to estimate the mean and variance of $L_{code}$. We average over 30 packets of 10000 bits for every value of the mean of $L_{in}$ we are interested in.
For larger value of $L_{code}$, we are less likely to make an error and we note that the mean of $L_{code}$ grows linearly with the mean of $L_{in}$. Therefore, for a higher value of $L_{in}$ for this decoder, the next decoder has a much better value of $L_{in}$ and this "positive feedback" makes iterative decoding very powerful. For large values of the mean of $L_{in}$, the variance and mean of $L_{code}$ are related as

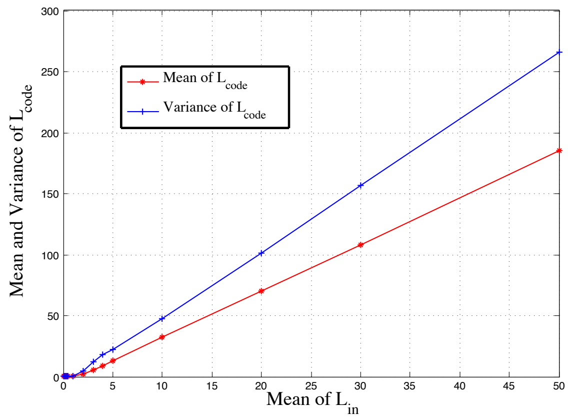
$$\frac{\text{Variance}}{\text{Mean}} \approx 1.44$$

**Histograms of $L_{code}$**
The graph shows that $L_{code}$ increases with an increase in $m$ and looks reasonably Gaussian. However, the key point to be noted is that for low values of $m (\approx 1)$, $L_{code}$ is highly concentrated close to 0. This means that iterative decoding will fail. The lesson to be taken away is that for iterative decoding to converge, we need sufficiently large $m$.

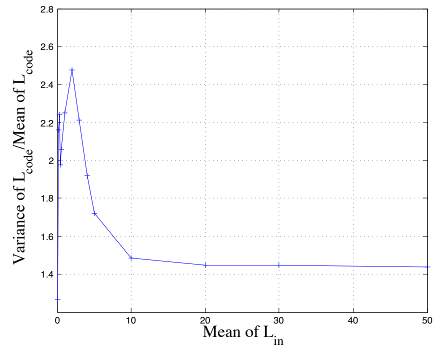
## Problem 7.16

### Part a

The interleaver used for the purposes of simulation is the one outlined by Berrou and Glavieux. In this part, we give a brief description of the interleaver.




(a) Mean and variance of $L_{code}$ as a function of the mean of $L_{in}$




(b) Distribution of $L_{code}$ for varying $\frac{E_b}{N_0}$ in a packet

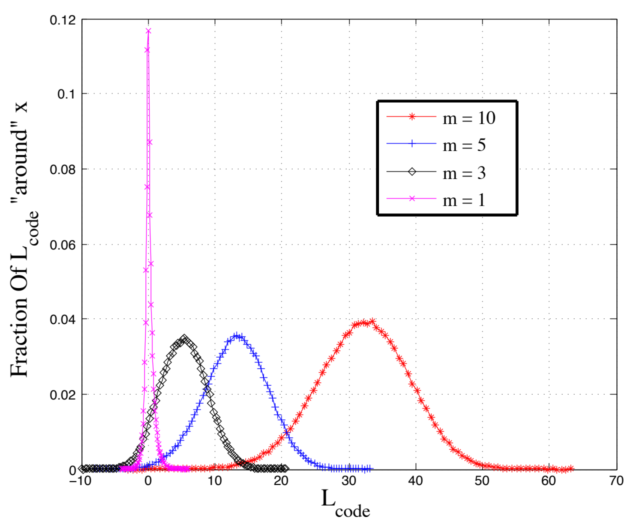



Figure 18: Histogram of $L_{code}$ with Gaussian LLR input for varying values of $m$

Assume that a random bit-stream of length $256^2$ comes in and the bits are written row-wise into a matrix $M_{256\times 256}$. Let us focus our attention on a particular bit $b$ and let us assume it is written into the position $(i, j)$. To accomplish interleaving, the matrix is permuted and let us denote the matrix we end up with by $\tilde{M}$. The bit $b$ which was in position $(i, j)$ in $M$ now finds itself in position $(i_r, j_r)$ in $\tilde{M}$ where $(i_r, j_r)$ is given by the following rules:

$$\begin{aligned}
i_r &= 129(i + j) \mod 256 \\
\zeta &= i + j \mod 8 \\
\mathbf{p} &= [17, 37, 19, 29, 41, 23, 13, 7] \\
j_r &= \mathbf{p}(\zeta) \times (j + 1) - 1 \mod 256
\end{aligned}$$

**Part b**

We now describe the simulation setup briefly and then discuss the results.

1. We use a rate $\frac{1}{4}$ code which is obtained by serial concatenation of the rate $\frac{1}{2}$ [7,5] RSC code.
2. The interleaver used has been described in detail in part (a).
3. BPSK modulation is used the channel is modelled as a discrete time AWGN channel.
4. An iterative decoding scheme is used where the inner and outer decoders are separated by and interleaver/deinterleaver to get the right ordering of the LLR's.

In this simulation, our aim will be to characterize the performance of turbo codes for Bit Error Rates (BER) greater than or equal to $10^{-4}$. To report a "stable" BER of $10^{-4}$, we would need to send a minimum number of bits and a rule of thumb would be,

$$\text{Number of bits observed} = 100 \times \frac{1}{\text{BER}}$$

This translates to a transmission of about 180 packets of 65536 bits each. We now present the results of the simulation.

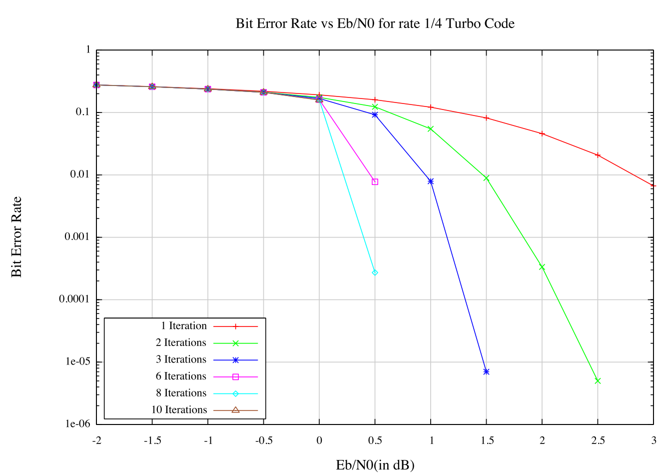



Figure 19: Performance of a rate $\frac{1}{4}$ serially concatenated turbo code

We see that with an increasing number of iterations, the BER falls off steeply beyond a cutoff point. In particular, we make the following observations:

**1.** After 8 "turbo" iterations, the BER for values of $\frac{E_b}{N_0}$ above 0.5 dB is roughly zero.

**2.** After 10 "turbo" iterations, the BER for values of $\frac{E_b}{N_0}$ above 0.5 dB is roughly zero.

From these observations, we can put the **turbo knee point somewhere between 0 and 0.5 dB**. Also, given a that we get a non-zero, albeit very small BER ($\approx 10^{-6}$) at 0 dB after 9 iterations, we would put the knee point closer to 0 dB than 0.5 dB. We now make a rough estimate of how far the code is from the Shannon limit.

From the graph, it is reasonable to suppose that a BER of $10^{-4}$ would be achieved with 8 "turbo" iterations at $\frac{E_b}{N_0} \approx 0.6$ dB. The spectral efficiency, $r$, of this code is given by

$$r = \underbrace{2}_{\text{Spectral efficiency of BPSK}} \times \underbrace{\frac{1}{4}}_{\text{Code Rate}} = \frac{1}{2}$$

The Shannon limit on the minimum SNR required to communicate at this spectral efficiency is given by

$$\left(\frac{E_b}{N_0}\right)_{min} = 2(\sqrt{2} - 1) \approx -0.8\text{dB}$$

Therefore, if we restrict ourselves to 8 turbo iterations, we are **1.4 dB away from the Shannon limit**. However, with an increase in the number of "turbo" iterations, it is conceivable that the $\frac{E_b}{N_0}$ required to achieve a BER of $10^{-4}$ would be lower. Hence, the value of **1.4 dB is an upper bound** to how far we are away from Shannon capacity.

## Part c

We now present the results of an EXIT analysis of the system. The EXIT charts are generated for three values of $\frac{E_b}{N_0}$: -0.5 dB, 0 dB and 0.5 dB. The curve for the outer decoder that doesn't have access to the channel doesn't depend on $\frac{E_b}{N_0}$.

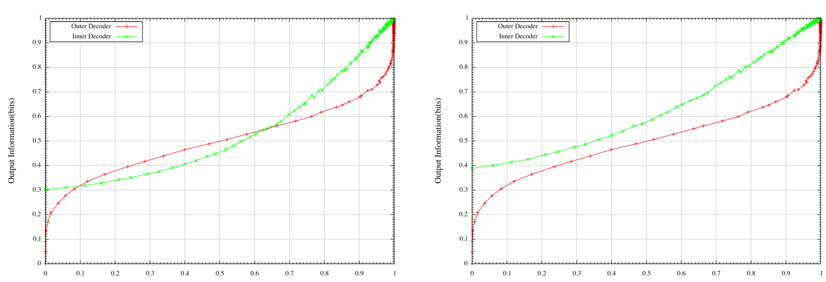



Figure 20: We see that the "tunnel" is closed at -0.5 dB while it is open at 0.5 dB

We see that the tunnel is closed at -0.5 dB and the turbo iterations won't converge while at 0.5 dB, they would converge. We also note that the curve at 0 dB is "just about" closed and therefore, the threshold would be just a bit more than 0 dB which tallies with the earlier observation we made.

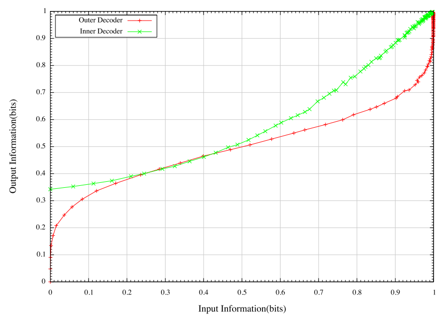



Figure 21: EXIT Chart at 0 dB: The tunnel is just about closed

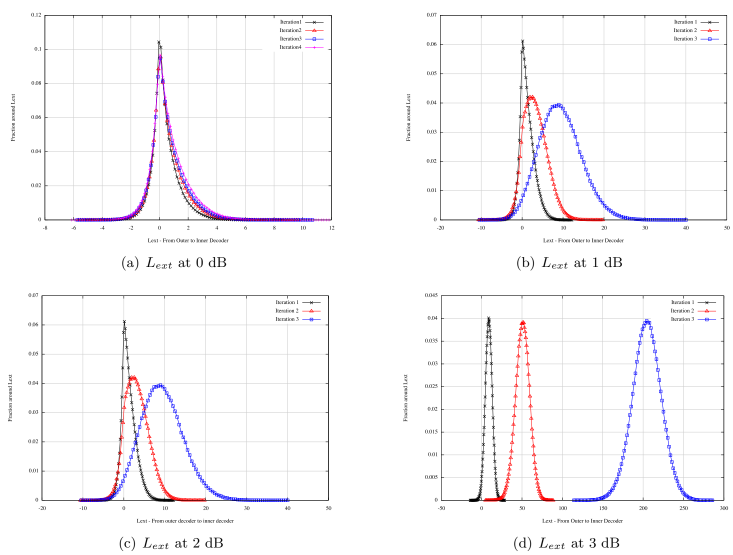



Figure 22: Histograms of $L_{ext}$ passed from the outer decoder to the inner decoder

## Part d

### Histograms of LLR's - Outer decoder to Inner decoder

We make the following observations:

1. At low values of $\frac{E_b}{N_0}(\leq 0\text{dB})$, the extrinsic LLR passed from the outer decoder to the inner decoder doesn't build up with increasing number of turbo iterations.

2. At $\frac{E_b}{N_0} \approx 1\text{dB}$, the LLR builds up slowly with an increase in the number of iterations.

3. At "large" values of $\frac{E_b}{N_0}(\geq 2\text{dB})$, the LLR builds up quite rapidly with an increase in the number of iterations. Hence, the probability of error drops sharply at these values of $\frac{E_b}{N_0}$.

4. The curves look reasonably Gaussian for the first 3 turbo iterations.

5. When we move onto the $6^{th}$ iteration, the curve is no longer Gaussian. A typical histogram for the $6^{th}$ iteration is shown below. At higher iterations, the curves completely lose any Gaussianity.

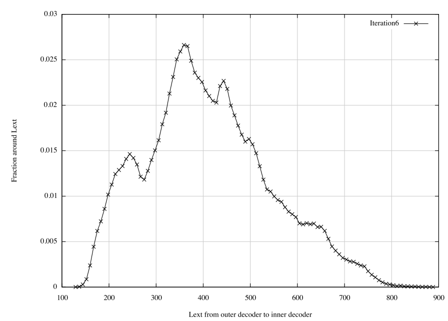
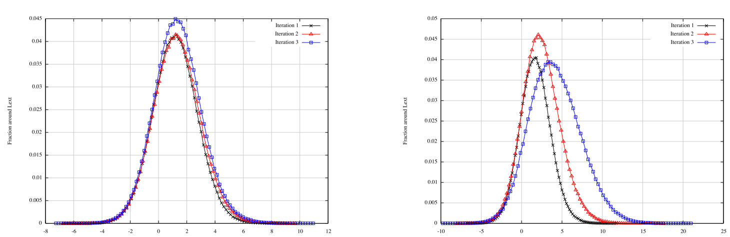



Figure 23: Histograms of $L_{ext}$ passed from the outer decoder to the inner decoder

### Histograms of LLR's - Inner decoder to Outer decoder

From the above figures, it is clear that a similar set of observations hold for the extrinsic information being passed from the inner decoder to the outer decoder. Once again, the curves for the LLR lose their Gaussianity after 6 iterations.




(a) $L_{ext}$ at 0 dB

(b) $L_{ext}$ at 1 dB

Figure 24: Histograms of $L_{ext}$ passed from the inner decoder to the outer decoder

### Estimation of "SNR" using Bit Errors

From the above figures, we see that for lower values of turbo iterations, the LLR's are well approximated by Gaussians. Since we make decisions based on the LLR's, it must be possible to

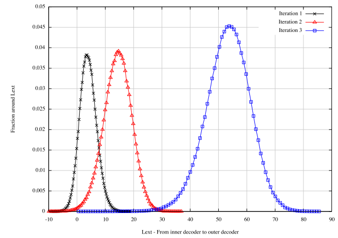


Figure 25: Histogram of $L_{ext}$ passed from the inner decoder to the outer decoder

derive a bit error rate based on the mean and variance of these Gaussians. Conversely, given the bit error rate, we can recover some information about the mean and the variance of the Gaussian. This analysis holds only when the Gaussian approximation is valid.

Let $X$ be a random variable that is distributed as $N(\mu, \sigma^2)$. We are interested in the probability $P(X < 0)$ which is given by

$$P(X < 0) \triangleq \beta = Q\left(\frac{\mu}{\sigma}\right)$$

Conversely, given $\beta$, we see that

$$\frac{\mu}{\sigma} = Q^{-1}(\beta)$$

We now estimate the quantity $\frac{\mu}{\sigma}$ from $\beta$ and compare it with the value obtained from an explicit evaluation of $\mu$ and $\sigma$ for the given realization. We now discuss these results in detail.

| $L_{ext}$ - Outer to Inner | | | |
|---|---|---|---|
| $\frac{E_b}{N_0}$ (in dB) | Iteration Number | $\frac{\mu}{\sigma}$ | $\frac{\hat{\mu}}{\sigma}$ |
| 0 dB | 1 | 0.399 | 0.369 |
| | 2 | 0.461 | 0.436 |
| | 3 | 0.488 | 0.466 |
| | 6 | 0.516 | 0.497 |
| | 10 | 0.525 | 0.509 |
| 2 dB | 1 | 0.693 | 0.696 |
| | 2 | 1.12 | 1.17 |
| | 3 | 1.8 | 1.98 |
| | 6 | 3.11 | - |
| | 10 | 1.67 | - |

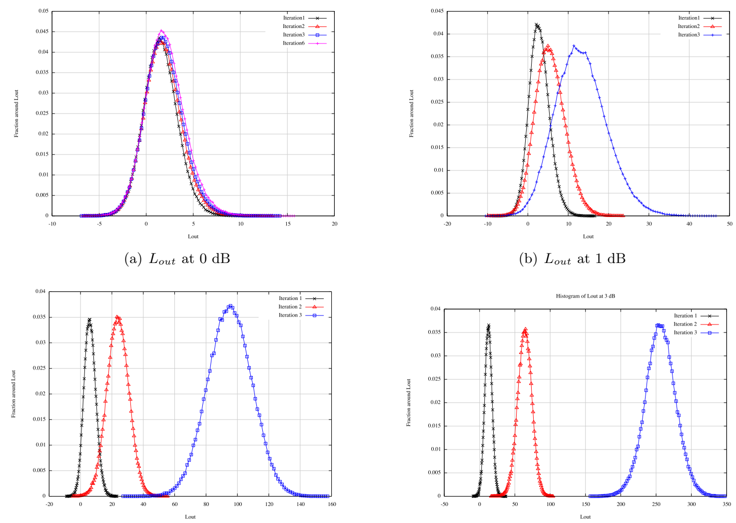
We make the following observations:

1. The predicted value closely matches the empirical value whenever the number of iterations is low. This holds regardless of the value of $\frac{E_b}{N_0}$.
2. The predicted value deviates significantly from the empirical value when the number of iterations becomes large. In particular, the '-' indicated that a value could not be estimated since the observed number of errors is zero.
3. To put it in a nutshell, the prediction closely follows the empirical value whenever the Gaussian approximation holds.

## Part e

### Histograms for $L_{out}$

The observations we made for the extrinsic LLR hold for $L_{out}$ too. For low $\frac{E_b}{N_0}$, $L_{out}$ doesn't



(a) $L_{out}$ at 0 dB

(b) $L_{out}$ at 1 dB

(c) $L_{out}$ at 2 dB

(d) $L_{out}$ at 3 dB

Figure 26: Histograms of $L_{out}$

build up. At moderate values of $\frac{E_b}{N_0}$, $L_{out}$ builds up quite rapidly and therefore, the error rate falls off drastically. Also, the curves lose their Gaussianity after 6 iterations or so.

### Estimating the "SNR" of LLR's using BER

We use the same method outlined before and make an estimate of the "SNR".

| $L_{out}$ | | | |
|---|---|---|---|
| $\frac{E_b}{N_0}$ (in dB) | Iteration Number | $\frac{\mu}{\sigma}$ | $\frac{\hat{\mu}}{\sigma}$ |
| 0 dB | 1 | 0.856 | 0.865 |
| | 2 | 0.913 | 0.928 |
| | 3 | 0.937 | 0.956 |
| | 6 | 0.958 | 0.985 |
| | 10 | 0.962 | 0.996 |
| 1 dB | 1 | 1.128 | 1.16 |
| | 2 | 1.5 | 1.59 |
| | 3 | 2.1 | 2.37 |
| | 6 | 3.11 | - |
| | 10 | 1.69 | - |
| 2 dB | 1 | 1.62 | 1.69 |
| | 2 | 3.42 | 3.49 |
| | 3 | 6.75 | - |
| | 6 | 6.369 | - |
| | 10 | 1.67 | - |

We see that the estimated value matches the empirical value when the iteration number is low, irrespective of the value of $\frac{E_b}{N_0}$. When the iteration number becomes higher, the Gaussian approximation no longer holds and we are not able to make good estimates.

## Problem 7.18

### Part a

The Benedetto bound tells us that the probability of error roughly goes like

$$P_e \approx \frac{1}{K} \sum_{k=1}^{\lfloor \frac{K}{2} \rfloor} 2k \binom{2k}{k} \frac{S^{(2+2p_{min})k}}{(1-S^{p_{min}-2})^{2k}} \Bigg|_{S=\exp\left(\frac{-RE_b}{N_0}\right)}$$

where $K$ is the number of information bits and $p_{min} = 4$.
We plot this expression for $K = 10^3, 10^4$ and $10^5$ for different values of $\frac{E_b}{N_0}$.

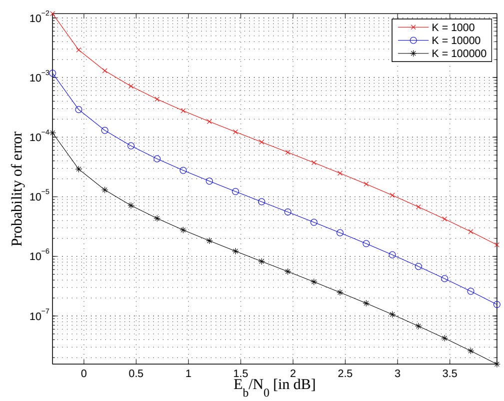



Figure 27: Performance of the rate $\frac{1}{3}$ turbo code formed by parallel concatenation of rate $\frac{1}{2}$ [7,5] RSC code for different number of input bits

We see that there is a significant gain in terms of bit error rate performance by increasing the input blocklength. From the graph, we see that a BER of $10^{-4}$ is achieved at $\frac{E_b}{N_0} = -0.3$ dB. The spectral efficiency $r$ of this code is given by

$$r = \underbrace{2}_{\text{Spectral efficiency of BPSK}} \times \underbrace{\frac{1}{3}}_{\text{Code Rate}} = \frac{2}{3}$$

The minimum value of $\frac{E_b}{N_0}$ to communicate at this spectral efficiency is given by

$$\left(\frac{E_b}{N_0}\right)_{min} = \frac{2^{\frac{2}{3}} - 1}{\frac{2}{3}} = 0.8811 = -0.5497 \text{ dB}$$

Therefore, we are about **0.25 dB** from capacity while using an input bitstream of length $10^5$.

### Part b

From the state diagram, we see that the minimum parity weight sequence corresponding to an input sequence of weight 2 is given by traversing the states $00 \to 10 \to 01 \to 00$. For this sequence of state transitions, the parity sequence has weight $p_{min} = 3$. Once again, we use the Benedetto bound that we employed in part (a) and obtain the graph below.

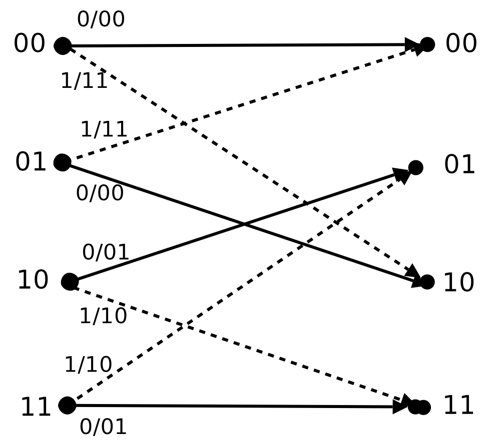



Figure 28: State diagram for the rate $\frac{1}{2}$ [5,7] RSC code

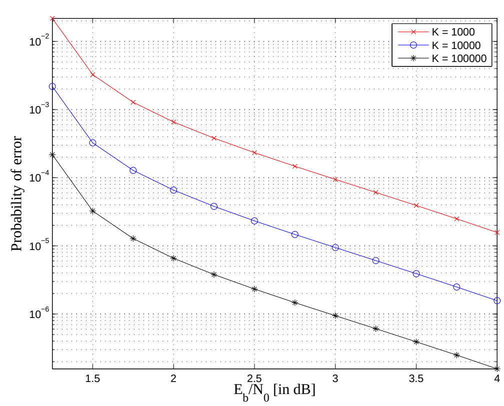



Figure 29: Performance of the rate $\frac{1}{3}$ turbo code formed by parallel concatenation of rate $\frac{1}{2}$ [5,7] RSC code for different number of input bits

### Part c

We now fix the input bitstream's blocklength to $10^4$ and compare the performance of the turbo codes obtained by parallel concatenation of rate $\frac{1}{2}$ [7,5] RSC codes on the one hand with that obtained by parallel concatenation of rate $\frac{1}{2}$ [5,7] RSC codes. We see that the turbo code obtained by the parallel concatenation of [7,5] RSC codes does better because its $p_{min}$ is greater than that obtained by the parallel concatenation of [5,7] RSC codes.

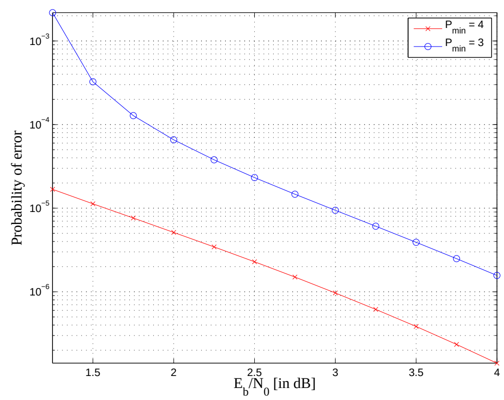



Figure 30: Performance comparison of the turbo codes obtained by parallel concatenation of rate $\frac{1}{2}$ [7,5] RSC codes and $\frac{1}{2}$ [5,7] RSC codes

```
% This program plots the Benedetto bound to get an idea of the performance
% of Turbo Codes

clear
clc
clf

% We use Eqn. 7.86 in the textbook to plot the Benedetto bound
ebnodb = -0.3:0.25:4;
ebnodb = ebnodb(:);
ebno = 10.^(ebnodb/10);
R = 1/3;

% Defining different values of K
K = [1e3;1e4;1e5];

% finding Pe as a function of Eb/N0 for each value of K
Pe = zeros(length(ebnodb),length(K));  % Intitializing probability of error

% Defining p_min
p_min1 = 4;

% Defining S = exp(-R*Eb/N0)
S = exp(-R*ebno);

for ctr_outer = 1:length(K)
    temp = zeros(length(ebnodb),1);
```

```
    % We terminate the summation with 30 terms because higher terms don't
    % contribute significantly
    for ctr_inner = 1:30
        temp = temp + 2*ctr_inner*(factorial(2*ctr_inner)/(factorial(ctr_inner))^2)*...
            (S.^(ctr_inner*(2+2*p_min1)))./((1 - S.^(p_min1-2)).^(2*ctr_inner));
    end
    Pe1(:,ctr_outer) = temp./K(ctr_outer);
end

semilogy(ebnodb,Pe1(:,1),'-xr')
hold on
semilogy(ebnodb,Pe1(:,2),'-ob')
semilogy(ebnodb,Pe1(:,3),'-*k')
axis tight
grid on
legend('K = 1000','K = 10000','K = 100000')
xlabel('E_b/N_0 [in dB]','FontName','Times','FontSize',14)
ylabel('Probability of error','FontName','Times','FontSize',14)

% ---------------------------------------------------------------------------

% --------------------- Part B ----------------------------------------
clear ebnodb ebno
ebnodb = 1.25:0.25:4;
ebnodb = ebnodb(:);
ebno = 10.^(ebnodb/10);
S = exp(-R*ebno);

Pe2 = zeros(length(ebnodb),length(K));
% In this case p_min = 3
p_min2 = 3;
for ctr_outer = 1:length(K)
    temp = zeros(length(ebnodb),1);
    % We terminate the summation with 30 terms because higher terms don't
    % contribute significantly
    for ctr_inner = 1:30
        temp = temp + 2*ctr_inner*(factorial(2*ctr_inner)/(factorial(ctr_inner))^2)* ...
            (S.^(ctr_inner*(2+2*p_min2)))./((1 - S.^(p_min2-2)).^(2*ctr_inner));
    end
    Pe2(:,ctr_outer) = temp./K(ctr_outer);
end
figure
semilogy(ebnodb,Pe2(:,1),'-xr')
hold on
semilogy(ebnodb,Pe2(:,2),'-ob')
semilogy(ebnodb,Pe2(:,3),'-*k')
axis tight
grid on
legend('K = 1000','K = 10000','K = 100000')
xlabel('E_b/N_0 [in dB]','FontName','Times','FontSize',14)
ylabel('Probability of error','FontName','Times','FontSize',14)

% ----------------------------------------------------------------------------
```

```
% ------------------- Part C ----------------------------------------
clear ebnodb ebno
ebnodb = 1.25:0.25:4;
ebnodb = ebnodb(:);
ebno = 10.^(ebnodb/10);
S = exp(-R*ebno);
K = 1e4;

Pe3 = zeros(length(ebnodb),2);

p_min1 = 4;
temp = zeros(length(ebnodb),1);
for ctr_inner = 1:30
    temp = temp + 2*ctr_inner*(factorial(2*ctr_inner)/(factorial(ctr_inner))^2)*...
        (S.^(ctr_inner*(2+2*p_min1)))./((1 - S.^(p_min1-2)).^(2*ctr_inner));
end
Pe3(:,1) = temp./K;

p_min2 = 3;
temp = zeros(length(ebnodb),1);
for ctr_inner = 1:30
    temp = temp + 2*ctr_inner*(factorial(2*ctr_inner)/(factorial(ctr_inner))^2)* ...
        (S.^(ctr_inner*(2+2*p_min2)))./((1 - S.^(p_min2-2)).^(2*ctr_inner));
end
Pe3(:,2) = temp./K;

figure
semilogy(ebnodb,Pe3(:,1),'-xr')
hold on
semilogy(ebnodb,Pe3(:,2),'-ob')
axis tight
grid on
legend('P_{min} = 4','P_{min} = 3')
xlabel('E_b/N_0 [in dB]','FontName','Times','FontSize',14)
ylabel('Probability of error','FontName','Times','FontSize',14)
```

## Problem 7.19

### Part a

Let us fix the value taken by $X_1$ to be $n_1$. This event occurs with a probability $p_1[n_1]$. Also, assume we are trying to evaluate the probability that $X$ takes the value $k$, i.e. $p_X[k]$. For this to happen, $X_2$ must take the value $n_2$ such that $n_1 + n_2 = k (\text{mod } 2)$. This event is independent of the event $X_1$ taking the value $n_1$. Therefore, by the law of total probability,

$$p_X[k] = \sum_{n_1+n_2=k(\text{mod}2)} p_1[n_1]p_2[n_2]$$

**Part b**

We have,

$$P_i[0] = p_i[0] + p_i[1] = 1$$
$$P_i[1] = p_i[0] + p_i[1]e^{-j\pi} = p_i[0] - p_i[1]$$

**Part c**

Using the property that cyclic convolution in the time domain corresponds to multiplication in the frequency domain, we see that

$$\begin{aligned} P_X[1] &= P_1[1]P_2[1] \\ &= (p_1[0] - p_1[1])(p_2[0] - p_2[1]) \end{aligned}$$

But by the result of part (b),

$$P_X[1] = p_X[0] - p_X[1]$$

Therefore,

$$p_X[0] - p_X[1] = (p_1[0] - p_1[1])(p_2[0] - p_2[1])$$

**Part d**

Let us try to derive an expression for $p_i[0]-p_i[1]$. By the definition of $m_i$, we have $p_i[0] = e^{m_i}p_i[1]$ and $p_i[0] + p_i[1] = 1$. Therefore,

$$\frac{p_i[0]}{p_i[1]} = \frac{e^{m_i}}{1}$$

By the principle of componendo and dividendo, we have,

$$\frac{\cancelto{1}{p_i[0] + p_i[1]}}{p_i[0] - p_i[1]} = \frac{e^{m_i} + 1}{e^{m_i} - 1} = \left(\frac{e^{\frac{m_i}{2}}}{e^{\frac{m_i}{2}}}\right)\left(\frac{e^{\frac{m_i}{2}} + e^{-\frac{m_i}{2}}}{e^{\frac{m_i}{2}} - e^{-\frac{m_i}{2}}}\right) = \frac{1}{\tanh(\frac{m_i}{2})}$$

Therefore, $p_i[0] - p_i[1] = \tanh(\frac{m_i}{2})$. By a similar argument, we have $p_X[0] - p_X[1] = \tanh(\frac{m}{2})$. By the result of part (c), we have

$$\tanh(\frac{m}{2}) = \tanh(\frac{m_1}{2})\tanh(\frac{m_2}{2})$$

**Part e**

Let us assume that the result holds true for the sum (modulo - 2) of $k$ independent binary random variables. In more detail, let $Y = X_1 + X_2 + \ldots + X_k$ with $m_Y$ being the LLR of $Y$ and $m_i$ being the LLR of $X_i$. Then, by assumption we have

$$\tanh(\frac{m_Y}{2}) = \tanh(\frac{m_1}{2})\tanh(\frac{m_2}{2})\ldots\tanh(\frac{m_k}{2})$$

We know that this statement is true for $k = 2$.Now, we need to prove the result for $k + 1$ independent binary random variables. Consider $Z = X_1 + X_2 + \ldots + X_{k+1} = Y + X_{k+1}$. Since all the $X_i$'s are independent, so are $Y$ and $X_{k+1}$. Therefore, by the result of part (d),

$$\tanh(\frac{m_Z}{2}) = \tanh(\frac{m_Y}{2})\tanh(\frac{m_{k+1}}{2})$$

By our induction assumption, we have

$$\tanh(\frac{m_Z}{2}) = \tanh(\frac{m_1}{2})\tanh(\frac{m_2}{2})\ldots\tanh(\frac{m_k}{2})\tanh(\frac{m_{k+1}}{2})$$

Hence the result has been proved for $k+1$ random variables and we are through.

# Problem 7.20

## Part a

Let $\alpha$ denote the crossover probability. Let $p_l$ denote the probability that the variable node sends out a 1 in the $l^{th}$ iteration. Then, from the recursion derived in the textbook, we have

$$p_l = \alpha\left(1 - \left\{\frac{1}{2}\left[1 + (1-2p_{l-1})^{d_c-1}\right]\right\}^{d_v-1}\right) + (1-\alpha)\left\{\frac{1}{2}\left[1 - (1-2p_{l-1})^{d_c-1}\right]\right\}^{d_v-1}$$

We see that the threshold of crossover probability, beyond which decoder failure occurs, is given by $\alpha_{threshold} = 0.04$.

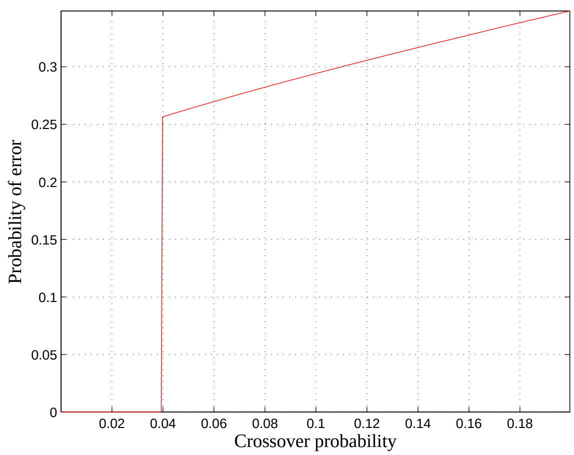



Figure 31: Plot of decoder probability of error vs. channel crossover probability

## Part b

In the first step, $m_v$ is distributed as,

$$\begin{aligned} m_v &= \log\left(\frac{1-\alpha}{\alpha}\right) && \text{with probability } 1-\alpha \\ &= \log\left(\frac{\alpha}{1-\alpha}\right) && \text{with probability } \alpha \end{aligned}$$

Therefore, $m_{u_0} = (1-2\alpha)\log\left(\frac{1-\alpha}{\alpha}\right)$. In the next step, we have $m_u = 2E\left[\tanh^{-1}\left(\prod_{i=1}^{5}\tanh\left(\frac{V_i}{2}\right)\right)\right]$ where

$$
\begin{aligned}
V_i &= \log\left(\frac{1-\alpha}{\alpha}\right) && \text{with probability } 1-\alpha \\
&= \log\left(\frac{\alpha}{1-\alpha}\right) && \text{with probability } \alpha
\end{aligned}
$$

From this stage, we use the Gaussian approximation as outlined in the textbook. The threshold for crossover probability beyond which decoder failure occurs is $\alpha_{threshold} = 0.05$.

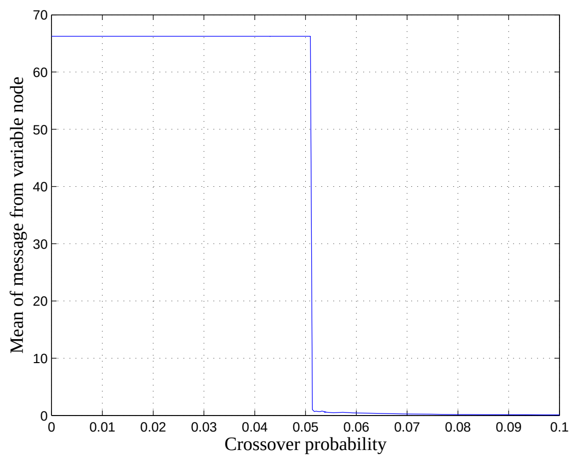



Figure 32: Plot of mean of the message from variable node against the channel crossover probability

**Part c**

For a rate $\frac{1}{2}$ LDPC code, $m_{u_0} = \frac{2E_b}{N_0} = (1-2\alpha)\log\left(\frac{1-\alpha}{\alpha}\right)$. From part (b), we have $\alpha_{threshold} = 0.05 \Rightarrow$

$$
\begin{aligned}
\left(\frac{E_b}{N_0}\right)_{threshold} &= 1.325 \\
&\approx 1.22\text{dB}
\end{aligned}
$$

For this case, the spectral efficiency is given by

$$
r = \underbrace{2}_{Real\ channel} \times \underbrace{\frac{1}{2}}_{Code\ Rate} = 1
$$

Therefore, the Shannon limit for this spectral efficiency is given by

$$
\begin{aligned}
\left(\frac{E_b}{N_0}\right)_{Shannon} &= \frac{2^1-1}{1} \\
&= 0\text{dB}
\end{aligned}
$$

Therefore, this code is **1.22 dB** from the Shannon limit.

### Part d

For hard decisions, we have $\alpha = Q\big(\sqrt{\frac{2RE_b}{N_0}}\big) = Q\big(\sqrt{\frac{E_b}{N_0}}\big)$ ($\because R = \frac{1}{2}$). Once again, to find the threshold for hard decisions we will need,

$$\begin{aligned} Q\Big(\sqrt{\frac{E_b}{N_0}}\Big) &= 0.05 \\ \Rightarrow \left(\frac{E_b}{N_0}\right)_{threshold,hard} &= 4.32\text{dB} \end{aligned}$$

The penalty for making hard decisions is **4.32 - 1.22 dB = 3.1 dB**.

### Part e

The threshold for the rate $\frac{1}{2}$ irregular LDPC Code occurs at $\frac{E_b}{N_0} = 1.1$dB. This is better than part (c) by about 0.12 dB.

```
% This program computes thresholds for LDPC codes using Gallager-A
% algorithm and Gaussian approximation. We do this for both regular and
% irregular LDPC codes

clear
clc

% ---------------------------------------------------------------------------
% Part A: Gallager's algorithm

% Defining degrees of check node and variable node
dv = 3;
dc = 6;

% Define the crossover probabilities of interest
alpha = [logspace(-6,-1.6,30) linspace(0.03,0.045,30) logspace(-1.3,-0.7,30)];
alpha = alpha(:);

% Defining number of iterations of message passing between the variable and
% check nodes
niter = 10000;

% Intitializing probability of error
prob_error = zeros(length(alpha),1);

% Implementing the recursive equations
for ctr_crossover = 1:length(alpha)
    p = alpha(ctr_crossover);
    for iter = 1:niter
        term1 = (1 - alpha(ctr_crossover))*(0.5*(1-(1-2*p)^(dc-1)))^(dv-1);
        term2 = alpha(ctr_crossover)*(1 - (0.5*(1+(1-2*p)^(dc-1)))^(dv-1));
        p = term1 + term2;
    end
    prob_error(ctr_crossover) = p;
```

```
end

plot(alpha,prob_error,'-r')
xlabel('Crossover probability','FontName','Times','FontSize',14)
ylabel('Probability of error','FontName','Times','FontSize',14)
grid on
axis tight

% ----------------------------------------------------------------------------

% ----------------------------------------------------------------------------
% Part B: Gaussian Approximation/Message Passing

clear alpha niter
alpha = [logspace(-6,-1.366,50) linspace(0.043,0.054,30) logspace(-1.27,-1,20)];
alpha = alpha(:);


% Number of random variables generated to estimate mu
num_mu = 10000;

% Number of random variables generated to estimate phi(x)
num_phi = 10000;

% We first generate phi(m) for various values of m
mean_phi = linspace(0,100,10000);
phi = zeros(10000,1);
for ctr = 1:length(phi)
    mean_ctr = mean_phi(ctr);
    Z = sqrt(2*mean_ctr)*randn(num_phi,1) + mean_ctr;
    phi(ctr) = 1 - mean(tanh(Z/2));
end

% Defining number of iterations of message passing between the variable and
% check nodes
niter = 100;

% Now based on the value of alpha we calculate mu - the mean of u
for ctr = 1:length(alpha)
    mu0 = (1-2*alpha(ctr))*log((1-alpha(ctr))/alpha(ctr));
    rand_mu = tanh(log((1-alpha(ctr))/alpha(ctr))*sign(rand(num_mu,dc-1)-alpha(ctr))/2);
    mu = 2*sum(atanh(prod(rand_mu,2)))/num_mu;

    for iter = 1:niter
        mean_iter = mu0 + (dv-1)*mu;
        % Next we compute phi(mean_iter)
        Z = sqrt(2*mean_iter)*randn(num_phi,1) + mean_iter;
        phi_mean_iter = 1 - mean(tanh(Z/2));
        % We now compute phi_inv(1 - (1-phi_mean_iter)^dc-1) using a
        % "look-up" table
        temp = 1 - (1-phi_mean_iter)^(dc-1);
        [val,index] = min(abs(temp - phi));
        mu = mean_phi(index);
```

```
    end

    mean_final(ctr) = mu;

end
figure
plot(alpha,mean_final)
xlabel('Crossover probability','FontName','Times','FontSize',14)
ylabel('Mean of message from variable node','FontName','Times','FontSize',14)
grid on


% -------------------------------------------------------------------------


% -------------------------------------------------------------------------
% PART E : IRREGULAR LDPC CODE

% Defining the parameters
lam2 = 0.2;
lam3 = 0.8;
rho5 = 0.5;
rho6 = 0.5;

% Range of interest of prior mean from channel about u
mean_prior = linspace(0.01,5,100);

% Defining number of iterations of message passing
niter = 500;

% Initializing the mean of the messages
mean_u = zeros(length(mean_prior),1);

for ctr = 1:length(mean_prior)
    mu0 = mean_prior(ctr);
    for iter = 1:niter

        % We now implement the recursive equation 7.114 in the textbook
        term2 = mu0 + mean_u(ctr);
        term3 = mu0 + 2*mean_u(ctr);

        % We now find phi(term2) and phi(term3)
        Z2 = term2 + sqrt(2*term2)*randn(num_phi,1);
        Z3 = term3 + sqrt(2*term3)*randn(num_phi,1);

        phi2 = 1 - mean(tanh(Z2/2));
        phi3 = 1 - mean(tanh(Z3/2));

        temp5 = 1 - (1 - lam2*phi2 - lam3*phi3)^4;
        temp6 = 1 - (1 - lam2*phi2 - lam3*phi3)^5;

        [val,index5] = min(abs(temp5 - phi));
        [val,index6] = min(abs(temp6 - phi));
```

```
        mean_u(ctr) = rho5*mean_phi(index5) + rho6*mean_phi(index6);
    end
end
```

## Problem 7.21

### Part a

We have $\lambda(x) = 0.3x^2 + 0.1x^3 + 0.6x^4$ and $\rho(x) = ax^7 + bx^8$. The rate $r$ is given by

$$r = 1 - \frac{\int_0^1 \rho(x)dx}{\int_0^1 \lambda(x)dx}$$

We have

$$\int_0^1 \lambda(x)dx = \frac{0.3}{3} + \frac{0.1}{4} + \frac{0.6}{5} = 0.245$$

Since the rate $r$ is given to be $\frac{1}{2}$, we have

$$\frac{1}{2} = 1 - \frac{\frac{a}{8} + \frac{b}{9}}{0.245}$$

Also $a + b = 1$. Solving these two equations, we get $a = 0.82$ and $b = 0.18$.

### Part b

The fraction of variable nodes with degree 4 is given by

$$\eta_{var} = \frac{\frac{\lambda_4}{4}}{\frac{\lambda_3}{3} + \frac{\lambda_4}{4} + \frac{\lambda_5}{5}} = \frac{0.1}{4 \times 0.245} = 0.102$$

### Part c

The fraction of check nodes with degree 9 is given by

$$\eta_{check} = \frac{\frac{\rho_9}{9}}{\frac{\rho_8}{8} + \frac{\rho_9}{9}} = \frac{0.02}{0.1025 + 0.02} = 0.1633$$

### Part d

From the figure, we see that the threshold for iterative decoding to converge is $\frac{E_b}{N_0} \approx 1.58$ dB The performance is worse than that of both part (c) & part (e) of Problem 7.20. The exact degradation is

1. **Compared to Part c:** 1.58 dB - 1.22 dB = **0.36 dB**
2. **Compared to Part d:** 1.58 dB - 1.1 dB = **0.48 dB**

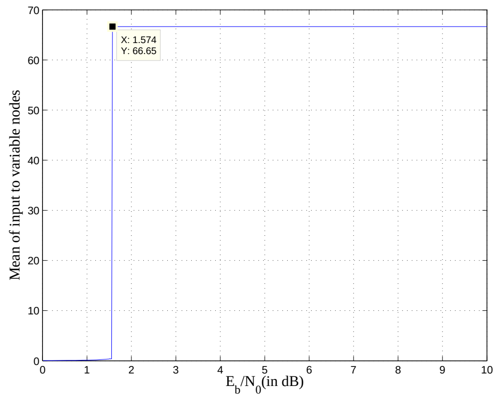



Figure 33: Performance of a rate $\frac{1}{2}$ LDPC code for across channels of varied quality. We see that a threshold $\frac{E_b}{N_0}$ of 1.58 dB is required for iterative decoding to converge

**Part e**

For the irregular LDPC code under consideration, we have

$$\lambda(x) = \sum_{i=3}^{5} \lambda_i x^{i-1} = 0.3x^2 + 0.1x^3 + 0.6x^4$$
$$\rho(x) = \sum_{i=8}^{9} \rho_i x^{i-1} = 0.82x^7 + 0.18x^8$$

Let $p_l$ denote the probability that a variable node sends a 1 in iteration $l$ and let $q_l$ denote the probability that a check node sends a 1 in iteration $l$.

The case of irregular LDPC codes differs from regular LDPC codes in that nodes could have different degrees. We know the density evolution when the degrees of the nodes are the same. Therefore, we condition on the degree of the node, apply the results for the regular LDPC code and then use the law of total probability. By this procedure, we have,

$$q_l = \sum_{i=8}^{9} \rho_i \frac{1 - (1 - 2p_l)^{i-1}}{2}$$

Similarly, we also have

$$p_l = \sum_{i=3}^{5} \lambda_i \Big( \big[\epsilon[1 - (1 - q_l)^{i-1}] + (1 - \epsilon) q_l^{i-1} \Big)$$

where $\epsilon$ is the channel crossover probability.

We observe that iterative decoding converges when the channel crossover probability $\epsilon < 0.04$. Therefore, the threshold value of $\frac{E_b}{N_0}$ above which iterative decoding fails is given by solving

$$0.04 = Q\left( \sqrt{2 \times \frac{1}{2} \frac{E_b}{N_0}} \right)$$
$$\Rightarrow \left( \frac{E_b}{N_0} \right)_{\text{threshold}} \approx 3.0649 \approx 4.86 \text{ dB}$$

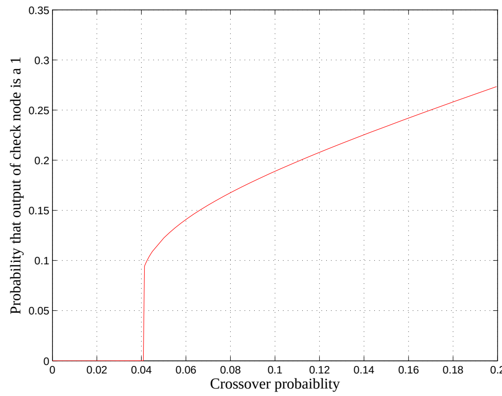



Figure 34: Performance of the rate $\frac{1}{2}$ irregular LDPC code with Gallager bit-flipping decoder

We lose about **3.2 dB** by making hard decisions on the channel output.

```
% This problem investigates the threshold for irregular LDPC Codes
clear
clc
clf

% Defining the lambda's
lam3 = 0.3;
lam4 = 0.1;
lam5 = 0.6;

% Defining the rho's
rho8 = 0.82;
rho9 = 0.18;

% We now calculate phi(m)
mean_phi = linspace(0,100,10000);
phi = zeros(10000,1);
% Number of random variables generated to estimate phi(x)
num_phi = 10000;
for ctr = 1:length(phi)
    mean_ctr = mean_phi(ctr);
    Z = sqrt(2*mean_ctr)*randn(num_phi,1) + mean_ctr;
    phi(ctr) = 1 - mean(tanh(Z/2));
end

% Defining number of iterations of message passing between the variable and
% check nodes
niter = 100;

% Defining the value of alpha
mean_prior = [linspace(2*10^(0.01/10),2*10^(0.15),10) linspace(2*10^(0.151),2*10^(0.17),
         linspace(2*10^(0.171),20,10)];
mean_prior = mean_prior(:);

% Initializing the mean vector
```

```
mean_u = zeros(length(mean_prior),1);

for ctr = 1:length(mean_prior)
    mu0 = mean_prior(ctr);
    for iter = 1:niter
        term3 = mu0 + 2*mean_u(ctr);
        term4 = mu0 + 3*mean_u(ctr);
        term5 = mu0 + 4*mean_u(ctr);

        % We find phi(term3),phi(term4),phi(term5)
        Z3 = term3 + sqrt(2*term3)*randn(num_phi,1);
        Z4 = term4 + sqrt(2*term4)*randn(num_phi,1);
        Z5 = term5 + sqrt(2*term5)*randn(num_phi,1);

        phi3 = 1 - mean(tanh(Z3/2));
        phi4 = 1 - mean(tanh(Z4/2));
        phi5 = 1 - mean(tanh(Z5/2));

        temp8 = 1 - (1 - lam3*phi3 - lam4*phi4 - lam5*phi5)^7;
        temp9 = 1 - (1 - lam3*phi3 - lam4*phi4 - lam5*phi5)^8;

        [val,index8] = min(abs(temp8 - phi));
        [val,index9] = min(abs(temp9 - phi));

        mean_u(ctr) = rho8*mean_phi(index8) + rho9*mean_phi(index9);
    end
end

% We know that the prior mean is 2Eb/N0
ebno = mean_prior/2;
ebnodb = 10*log10(ebno);

plot(ebnodb,mean_u)
grid on
xlabel('E_b/N_0(in dB)','FontName','Times','FontSize',14)
ylabel('Mean of input to variable nodes','FontName','Times','FontSize',14)


% ---------------------------------------------------------------------------
% Comparing with the Gallager bit flipping algorithm

% Define the crossover probabilities of interest
alpha = [logspace(-6,-1.6,30) linspace(0.03,0.045,30) logspace(-1.3,-0.7,30)];
alpha = alpha(:);

% Defining number of iterations of message passing between the variable and
% check nodes
niter = 1000;

% Intitializing probability of error
prob_error = zeros(length(alpha),1);

% Implementing the recursive equations
```

```
for ctr_crossover = 1:length(alpha)
    p = alpha(ctr_crossover);
    epsilon = alpha(ctr_crossover);
    for iter = 1:niter
        q = rho8 * (1 - (1-2*p)^7)/2 + rho9 * (1 - (1-2*p)^8)/2;

        p_term3 = epsilon*(1 - (1-q)^2) + (1-epsilon)*q^2;
        p_term4 = epsilon*(1 - (1-q)^3) + (1-epsilon)*q^3;
        p_term5 = epsilon*(1 - (1-q)^4) + (1-epsilon)*q^4;

        p = lam3*p_term3 + lam4*p_term4 + lam5*p_term5;
    end
    prob_error(ctr_crossover) = p;
end

figure
plot(alpha,prob_error,'-r')
grid on
xlabel('Crossover probaiblity','FontName','Times','FontSize',14)
ylabel('Probability that output of check node is a 1','FontName','Times','FontSize',14)
```

## Problem 7.22

### Part a

At the end of the first iteration, the only way a check node **cannot** output an erasure on a given branch is if none of its (other) inputs are erasures. This event occurs with the probability $(1-q_v)^5$. Therefore, the probability that a check node outputs an erasure is

$$q_u = 1 - (1-q_v)^5 = 1 - (1-q_0)^5 = 0.6723$$

The check node will output a 1 (an error) on a given branch if 1, 3 or 5 of the inputs on the other branches to check node are a 1 with the other inputs being a 0. This occurs with the probability

$$p_u = \binom{5}{1} p_0(1-p_0-q_0)^4 + \binom{5}{3} p_0^3(1-p_0-q_0)^2 + \binom{5}{5} p_0^5 = 0.0529$$

### Part b

Let $C$ represent the output of the channel and let $X_1$ and $X_2$ denote the messages coming into the variable node from the check nodes. Let us define a triple $(C, X_1, X_2)$ to represent the various possibilities of messages that come into a variable node. Note that the triple determines the message going out of the variable node. We will now outline the possible values the triple can take such that the outgoing message is a 1. The possibilities are $CX_1X_2 = \{111, 110, 101, 011, e11, 1e1, 11e, ee1, e1e, 1ee\}$. Therefore, the probability of the outgoing message being a 1 is the same as the probability that any one of the listed triple occurs and is given by,

$$\begin{aligned} p_v &= p_0p_u^2 + 2p_u(1-p_u-q_u)p_0 + (1-p_0)p_u^2 + q_0p_u^2 + 2p_0p_uq_u + 2q_0q_up_u + p_0q_u^2 \\ &= 0.033 \end{aligned}$$

The values of the triple for which the outgoing message is an erasure are given by, $CX_1X_2 = \{0e1, 01e, 1e0, 10e, e10, e01, eee\}$. Therefore, the probability that the variable node sends back an erasure is given by,

$$\begin{aligned}
q_v &= 2(1-p_0-q_0)p_uq_u + 2p_0q_u(1-p_u-q_u) + 2q_0(1-p_u-q_u)p_u + q_0q_u^2 \\
&= 0.162
\end{aligned}$$

## Problem 7.23

### Part a

Let us define the double $(x_1, x_2)$ based on the bits to be coded. The pairing of the doubles with the 4-PAM alphabet is shown in the diagram. For example, $(x_1, x_2) = (0, 1)$ is mapped to $+A$. We now proceed to calculate the log-likelihood ratios when the received value is $y$.

$$\begin{aligned}
L_1(y) &= \log\frac{P(X_1=0|Y=y)}{P(X_1=1|Y=y)} = \log\left(\frac{p(y|X_1=0)}{p(y|X_1=1)} \times \cancelto{1}{\frac{P(X_1=0)}{P(X_1=1)}}\right) \\
&= \log\left[\frac{\exp\left(-\frac{(y-A)^2}{2\sigma^2}\right) + \exp\left(-\frac{(y-3A)^2}{2\sigma^2}\right)}{\exp\left(-\frac{(y+A)^2}{2\sigma^2}\right) + \exp\left(-\frac{(y+3A)^2}{2\sigma^2}\right)}\right] \\
&= \log\left[\frac{\exp\left(\frac{Ay}{\sigma^2}\right)}{\exp\left(\frac{-Ay}{\sigma^2}\right)} \times \frac{1+\exp\left(\frac{2Ay}{\sigma^2}\right)\exp\left(\frac{-4A^2}{\sigma^2}\right)}{1+\exp\left(\frac{-2Ay}{\sigma^2}\right)\exp\left(\frac{-4A^2}{\sigma^2}\right)}\right] \\
&= \frac{2Ay}{\sigma^2} + \log\left[\frac{1+\exp\left(\frac{2A}{\sigma^2}(y-2A)\right)}{1+\exp\left(\frac{-2A}{\sigma^2}(y+2A)\right)}\right]
\end{aligned}$$

The likelihood ratio for the second bit is given by,

$$\begin{aligned}
L_2(y) &= \log\frac{P(X_2=0|Y=y)}{P(X_1=2|Y=y)} = \log\left(\frac{p(y|X_2=0)}{p(y|X_2=1)} \times \cancelto{1}{\frac{P(X_2=0)}{P(X_2=1)}}\right) \\
&= \log\left[\frac{\exp\left(-\frac{(y-3A)^2}{2\sigma^2}\right) + \exp\left(-\frac{(y+3A)^2}{2\sigma^2}\right)}{\exp\left(-\frac{(y-A)^2}{2\sigma^2}\right) + \exp\left(-\frac{(y+A)^2}{2\sigma^2}\right)}\right] \\
&= \log\left[\frac{\exp\left(-\frac{9A^2}{2\sigma^2}\right)}{\exp\left(\frac{-A^2}{2\sigma^2}\right)} \times \frac{\exp\left(\frac{-3Ay}{\sigma^2}\right) + \exp\left(\frac{3Ay}{\sigma^2}\right)}{\exp\left(-\frac{Ay}{\sigma^2}\right) + \exp\left(\frac{Ay}{\sigma^2}\right)}\right] \\
&= -\frac{4A^2}{\sigma^2} + \log\frac{\cosh\left(\frac{3Ay}{\sigma^2}\right)}{\cosh\left(\frac{Ay}{\sigma^2}\right)}
\end{aligned}$$

### Part b

Firstly, we have

$$E_s = \frac{1}{4}(2A^2 + 18A^2) = 5A^2$$

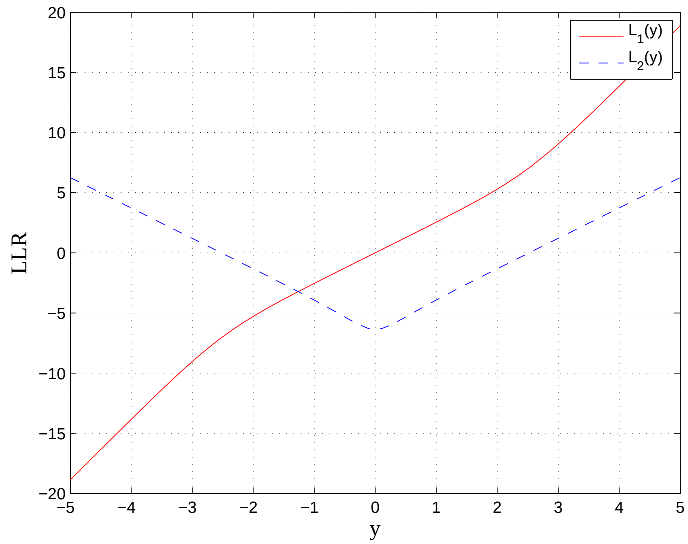



Figure 35: Plot of log-likelihood ratios of the received values when $\frac{E_b}{N_0} = 6$ dB

Next, we have,

$$\begin{aligned} E_s &= \text{Energy per symbol} = \text{Energy per 2 code bits} \\ &= \text{Energy per information bit} \qquad \because \text{rate} = \frac{1}{2} \\ &= E_b \end{aligned}$$

We also have $\sigma^2 = \frac{N_0}{2} = 1$. Therefore, we have, $\frac{E_b}{N_0} = \frac{5A^2}{2}$. Also, $\frac{E_b}{N_0} = 6\text{dB} \Rightarrow \frac{E_b}{N_0} \approx 4$. Finally, we have

$$\begin{aligned} \frac{5A^2}{2} &= 4 \\ \Rightarrow A &= \sqrt{\frac{8}{5}} \approx 1.265 \end{aligned}$$

As usual, let SNR denote $\frac{E_s}{\sigma^2}$. In general, we then have,

$$A = \sqrt{\frac{\text{SNR}}{5}}$$

**Part c**

We see that the capacity of BICM is almost the same as the capacity of equiprobable 4-PAM and the degradation is negligible, especially at higher values of SNR.

**Part d**

Once again, we see that the spectral efficiencies of BICM and 4-PAM differ at low values of $\frac{E_b}{N_0}$. However, when $\frac{E_b}{N_0} > 2$ dB, the performance of BICM is virtually optimal.

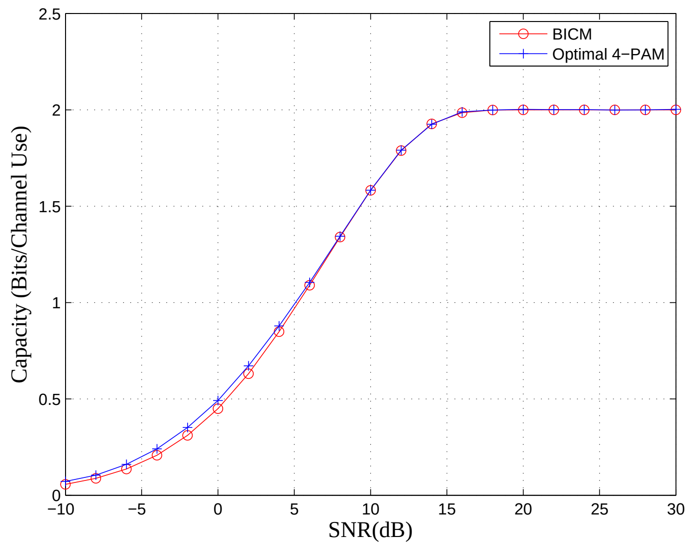



Figure 36: Comparing the capacity of the BICM scheme with that of equiprobable 4-PAM

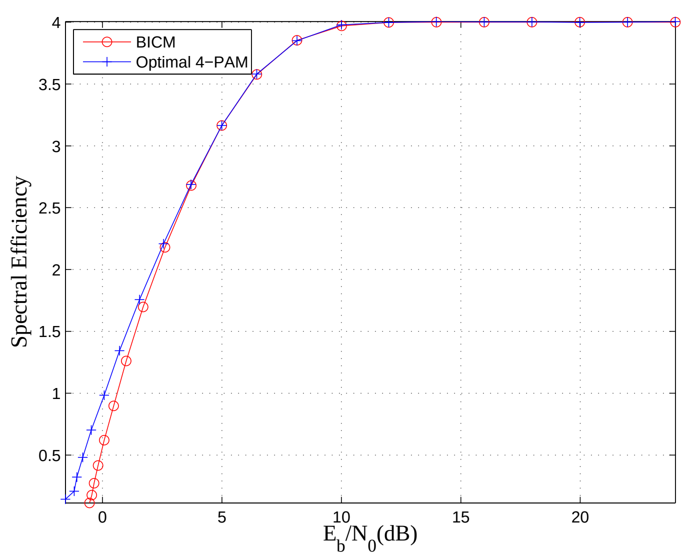



Figure 37: Comparing the spectral efficiency of the BICM scheme with that of equiprobable 4-PAM

```
% This program computes the capacity of the BICM scheme with a 4-PAM
% alphabet and compares the capacity of 4-PAM

clear
clc
clf

rand('twister',sum(100*clock));
randn('state',sum(100*clock));

% First, we plot the log-likelihood functions for both bits as a function
% of the received symbol values when Eb/No = 6 dB

% Defining Eb/N0 and A
ebnodb = 6;
ebno = 10.^(ebnodb/10);
A = sqrt(2*ebno/5);

% Defining a vector of received values
y = linspace(-5,5,50);

% Computing the log-likelihood ratios
L1 = 2*A*y + log((1 + exp(2*A*(y-2*A)))./(1 + exp(-2*A*(y+2*A))));
L2 = -4*A^2 + log(cosh(3*A*y)./cosh(A*y));

% Plotting the log-likelihood functions
plot(y,L1,'-r')
hold on
grid on
plot(y,L2,'--b')
set(gca,'XTick',-5:1:5)
xlabel('y','FontName','Times','FontSize',14)
ylabel('LLR','FontName','Times','FontSize',14)
legend('L_1(y)','L_2(y)','Location','NorthEast')

% ---------------------------------------------------------------------------


% ---------------------------------------------------------------------------
% Computing capacity of BICM scheme
clear L1 L2
% Generating bits in an equiprobable fashion. These actually represent code
% bits
num_bits = 500000;              % Number of bits
bits = round(rand(num_bits,1));

% Mapping bits into 4-PAM symbols
bits_odd_posn = bits(1:2:end);      % Getting bits in "odd" positions
bits_even_posn = bits(2:2:end);     % Getting bits in "even" positions
pam_symbols = bits_odd_posn + 2*bits_even_posn;

% These PAM symbols take values 0,1,2,3. We first shift the symbols so that
% the mean is zero and take values {-3,-1,1,3}. We then set sigma^2 = 1 and
```

```
% scale the symbols based on the SNR
pam_symbols = 2*(pam_symbols - 1.5);
snrdb = -10:2:30;
snr = 10.^(snrdb/10);
snrdb = snrdb(:);
snr = snr(:);

% Intializing variables
cap_bicm = zeros(length(snr),1);
cap_4pam = zeros(length(snr),1);

for ctr_snr = 1:length(snr)
    A = sqrt(snr(ctr_snr)/5);
    pam_symbols_snr = A*pam_symbols;    % Noiseless symbols at a given SNR
    pam_symbols_rec = pam_symbols_snr + randn(num_bits/2,1); % Adding noise


    % -------------------- CAPACITY OF BICM -----------------------------
    % Computing the log-likelihood ratios
    L1 = 2*A*pam_symbols_rec + log((1 + exp(2*A*(pam_symbols_rec-2*A)))./(1 + ...
        exp(-2*A*(pam_symbols_rec+2*A))));
    L2 = -4*A^2 + log(cosh(3*A*pam_symbols_rec)./cosh(A*pam_symbols_rec));

    % We now compute the posterior probabilities of the bits
    p1 = 1./(exp(L1) + 1);
    p2 = 1./(exp(L2) + 1);

    % Next we compute the conditional entropies H(X1|Y) and H(X2|Y)
    % H(X1|Y) = average(-p1(k)*log(p1(k)) and similarly for H(X2|Y)
    cond_entropy1 = cond_entropy(p1);   % refer to cond_entropy function below
    cond_entropy2 = cond_entropy(p2);   % refer to cond_entropy function below

    % Next we compute the capacities seen by the "odd" and "even" bits
    C1 = 1 - cond_entropy1;     % Capacity of channel seen by odd bits
    C2 = 1 - cond_entropy2;     % Capacity of channel seen by even bits

    % Capacity of BICM at the current value of SNR
    cap_bicm(ctr_snr) = C1 + C2;

    % --------------------- CAPACITY OF PAM ------------------------------

    % First, we compute the probabilities of receiving the various symbols
    % To this end, we compute the conditional probabilities and then
    % average it out. ASSUMPTION : EQUIPROBABLE 4-PAM AND WE DON'T OPTIMIZE
    % OVER THE ALPHABET
    const = 1/sqrt(2*pi);
    prob_A = const*exp(-(pam_symbols_rec - A).^2/2);        % p(y|A)
    prob_negA = const*exp(-(pam_symbols_rec + A).^2/2);     % p(y|-A)
    prob_3A = const*exp(-(pam_symbols_rec - 3*A).^2/2);     % p(y|-3A)
    prob_neg3A = const*exp(-(pam_symbols_rec + 3*A).^2/2);  % p(y|3A)
    prob_rec = 0.25*(prob_A + prob_negA + prob_3A + prob_neg3A);

    % We know that capacity of PAM is given by C = h(Y) - h(Y|X) where
```

```
    % h(Y|X) = 0.5*log2(2*pi*e)
    cap_4pam(ctr_snr) = (2/num_bits)*sum(log2(1./prob_rec)) - 0.5*log2(2*pi*exp(1));


end

figure
plot(snrdb,cap_bicm,'-or')
hold on
plot(snrdb,cap_4pam,'-+b')
grid on
xlabel('SNR(dB)','FontName','Times','FontSize',14)
ylabel('Capacity (Bits/Channel Use)','FontName','Times','FontSize',14)
legend('BICM','Optimal 4-PAM')


% -------------------------------------------------------------------------

% -------------------- Converting to Eb/N0 ------------------------------
% Let us use r to denote the spectral efficiency

r_bicm = 2*cap_bicm;        % Spectral efficiency of BICM
r_4pam = 2*cap_4pam;        % Spectral efficiency of 4-PAM

ebno_bicm = snr./r_bicm;    % Corresponding Eb/N0 for BICM
ebno_4pam = snr./r_4pam;    % Corresponding Eb/N0 for 4-PAM

ebnodb_bicm = 10*log10(ebno_bicm);
ebnodb_4pam = 10*log10(ebno_4pam); % Converting Eb/N0 to dB

figure
plot(ebnodb_bicm,r_bicm,'-or')
hold on
plot(ebnodb_4pam,r_4pam,'-+b')
grid on
xlabel('E_b/N_0(dB)','FontName','Times','FontSize',14)
ylabel('Spectral Efficiency','FontName','Times','FontSize',14)
legend('BICM','Optimal 4-PAM','Location','NorthWest')
axis tight

% ------------------------ CONDITIONAL ENTROPY FUNCTION -----------------
% This function calculates the conditional entropy from a given set of
% probabilities taking care of the convention that 0 log 0 = 0

function entropy = cond_entropy(prob)

len = length(prob);
entropy = 0;
for ctr = 1:len
    if (prob(ctr) == 0 || prob(ctr) == 1)
        ;       % From the convention 0 log 0 = 1 log 1 = 0
    else
        entropy = entropy - prob(ctr)*log2(prob(ctr)) - (1-prob(ctr))*log2(1-prob(ctr));
```

```
    end
end

entropy = entropy/len;
```

## Problem 24

### Part a

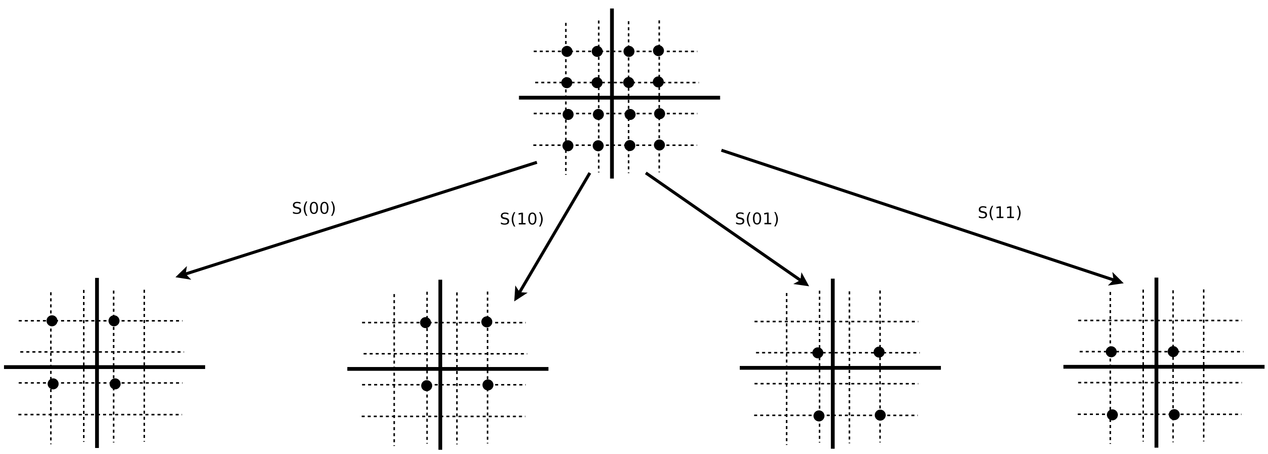



Figure 38: An Ungerboeck partitioning of 16-QAM into 4 sets of "translated" QPSK. Note that $S(c_1c_2)$ denotes the subset to be selected when the code bits from the convolutional encoder are $(c_1, c_2)$

### Part b

Let $2a$ denote the distance between adjacent constellation points as shown in the figure. In this case, the average energy per symbol (assuming all symbols are equiprobable) is given by

$$
\begin{aligned}
E_s &= \frac{1}{16}\left[4 \times (a^2 + a^2) + 8 \times (9a^2 + a^2) + 4 \times (9a^2 + 9a^2)\right] \\
&= \frac{160a^2}{16} = 10a^2
\end{aligned}
$$

Also, since we are comparing 16-QAM + TCM with 8-QAM, let us set $E_b = \frac{1}{3}$ so that $E_s =$ 3bits/symbol $\times \frac{1}{3} = 1$. From the previous equation, we have $a^2 = \frac{1}{10}$. The subset which has minimum Euclidean distance to the all-zero codeword corresponds to the $(c_1, c_2)$ sequence (01,10,01) - the same as that depicted in Figure 7.26 of the textbook. The squared Euclidean distance between the corresponding subset sequences is given by

$$
\begin{aligned}
d^2_{subset} &= d^2(S(00), S(01)) + d^2(S(00), S(10)) + d^2(S(00), S(01)) \\
&= [(2a)^2 + (2a)^2] + [(2a)^2] + [(2a)^2 + (2a)^2] \\
&= 20a^2
\end{aligned}
$$

Therefore, $d_{subset} = \sqrt{20}a$, which is greater than $d_{TCM} = 4a$ - the distance between the uncoded bits $(c_3, c_4)$ in any of the 4 subsets. Since $10a^2 = 1$, we have $d_{TCM}^2 = 16a^2 = 1.6$. For uncoded 8-PSK, with $E_s = 1$, the distance between nearest neighbours is given by $d_{8-PSK} = 2\sin\frac{\pi}{8}$. Therefore, the performance improvement of the TCM scheme when compared to uncoded 8-PSK is given by

$$\begin{aligned}
\text{Asymptotic Coding Gain} &= 10\log_{10}\left(\frac{d_{TCM}^2}{d_{8-PSK}^2}\right) \\
&= 10\log_{10}\left(\frac{1.6}{4\sin^2\frac{\pi}{8}}\right) = 4.36\text{dB}
\end{aligned}$$

# Chapter 8 - Fundamentals of Digital Communication

## Problem 8.1

The channel gain $h = h_R + jh_I$. Since $h$ is $CN(m, 2v^2)$, $h_R \sim N(m_R, v^2)$ and $h_I \sim N(m_I, v^2)$, with $m_R$ and $m_I$ being real and imaginary parts of $m$. The nominal SNR is

$$\mathbb{E}[|h|^2] = \mathbb{E}[h_R^2] + \mathbb{E}[h_I^2] = |m|^2 + 2v^2 = 2v^2(K+1)$$

Let $L$ represent the additional power gain used to combat fading. Then, the total average SNR is $L(2v^2(1+K))$ and the instantaneous SNR is $L|h|^2$. Probability of outage calculated using nominal SNR is given as

$$\begin{aligned} P_{out} &= P(L|h|^2 < 2v^2(1+K)) \\ &= P\left(G < \frac{2v^2(1+K)}{L}\right) \qquad\qquad G = |h|^2 \end{aligned}$$

### Part a

Rayleigh channel ($K = 0$) implies $G = |h|^2$ is exponential random variable with mean $2v^2$.

$$\begin{aligned} P_{out} &= \frac{1}{2v^2}\int_0^{2v^2/L} e^{-\frac{g}{2v^2}}\,dg \\ &= 1 - e^{-1/L} \end{aligned}$$

$P_{out} = 5\%$ implies $L = 19.5 \approx 12.9$dB.

### Part b

For AWGN, channel with $K = \infty$, $h \sim CN(m, 0)$ or $h$ is just a constant. So, instantaneous SNR equals average SNR. No additional link margin is needed.

### Part c

To find the probability of outage, we need to find the density of $G$ which becomes involved for general $K$. We use simulations in MATLAB to find the probability. Let us first show that $G$ only depends on $|m|$ and not on the exact choice of $m_R$ and $m_I$. $G$ can be written as

$$\begin{aligned} G &= [m_R + n_1 v]^2 + [m_I + n_2 v]^2 \qquad n_1, n_2 \text{ are independent } N(0,1) \\ &= |m|^2 + v^2(n_1^2 + n_2^2) + 2v(m_R n_1 + m_I n_2) \\ &= |m|^2 + v^2(n_1^2 + n_2^2) + 2vn \qquad n \sim N(0, |m|^2) \end{aligned}$$

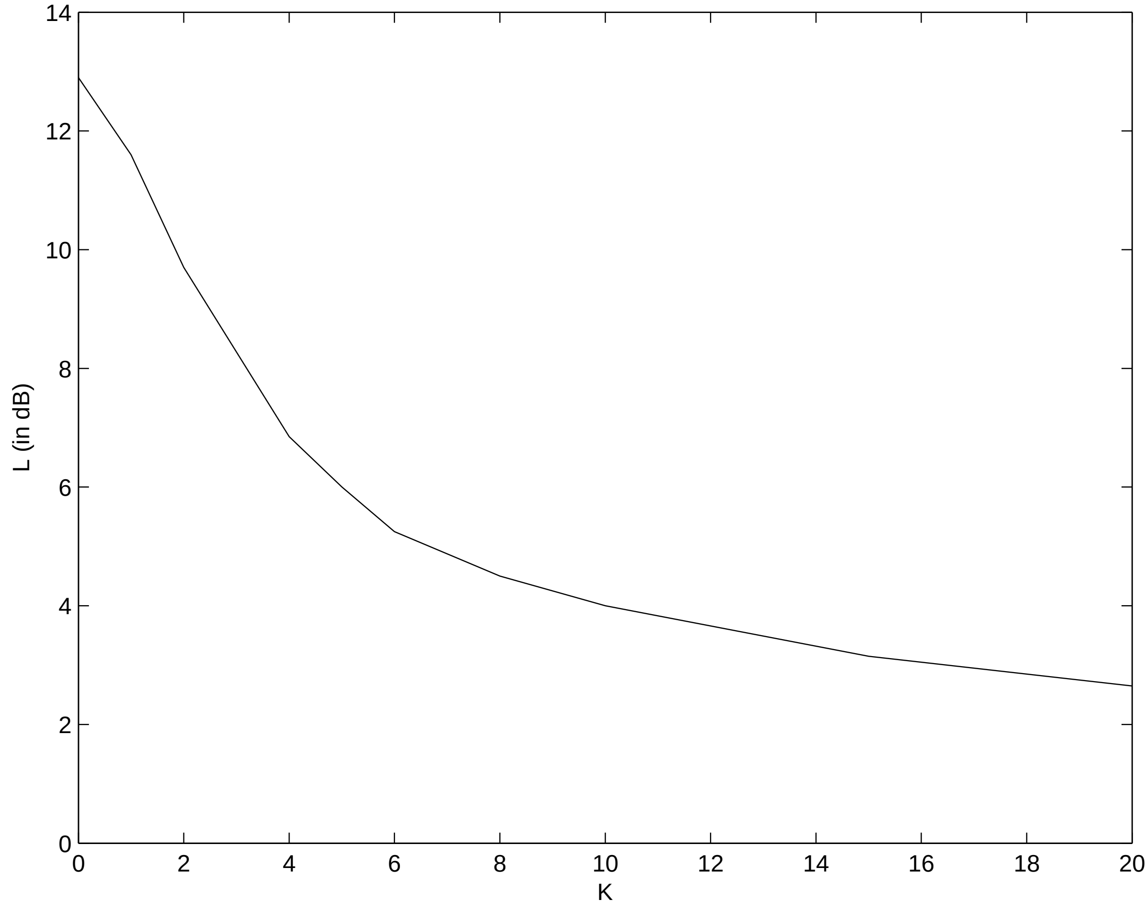



Figure 1: Additional SNR needed as a function of $K$ for Problem 8.1

Hence, we take $m_R = |m|$ and $m_I = 0$. Now, the outage probability is given as

$$
\begin{aligned}
P_{out} &= P\left(G < \frac{2v^2(1+K)}{L}\right) \\
&= P\left(G' < \frac{1}{L}\right) \qquad G' = \frac{G}{2v^2(1+K)} \\
&= \frac{1}{M}\sum_{i=1}^{M} I\left(G'_i < \frac{1}{L}\right)
\end{aligned}
$$

where $G'_i$ is the $i^{\text{th}}$ instance of $G'$ generated in MATLAB and $I(\mathbb{A})$ is Indicator function which is 1 if $\mathbb{A}$ is true and 0 if $\mathbb{A}$ is false. Now, $L$ is varied so that $P_{out} \approx 0.05$. The variation of $L$ w.r.t $K$ is plotted in Fig. 1.

The additional SNR $L = 3$dB, for $K \approx 16$.

### Generation of $G'$

We can write $G' = {h'_R}^2 + {h'_I}^2$, with

$$h'_R = \frac{h_R}{\sqrt{2v^2(1+K)}} \sim N(\frac{|m|}{\sqrt{2v^2(1+K)}}, \frac{1}{2(1+K)}) \sim N(\sqrt{\frac{K}{1+K}}, \frac{1}{2(1+K)})$$

$$h'_I \sim N(0, \frac{1}{2(1+K)})$$

Given $K$, both $h'_R$ and $h'_I$ can be generated using *randn* function in MATLAB.

## Problem 8.2

### Part a

We have $h(t)$ are $CN(0, P(t))$ and independent for different $t$. Also, given that

$$H(f) = \int h(t)e^{-j2\pi ft}dt$$

The above integral is treated as a limiting sum of large number of $CN$ variables and hence the integral is also a $CN$. So, $H(f)$ is $CN$ random variable with mean 0 and variance given as

$$\begin{aligned}
\mathbb{E}[H(f)H^*(f)] &= \mathbb{E}\left[\int\int h(t_1)h^*(t_2)e^{-j2\pi f(t_1-t_2)}dt_1dt_2\right] \\
&= \int \mathbb{E}[|h(t_1)|^2]dt_1 \qquad\qquad \text{(Uncorrelated } h(t)\text{)} \\
&= \int P(t_1)dt_1 = 1
\end{aligned}$$

### Part b

Since $H(f)$ is zero mean, the required covariance is given as

$$\begin{aligned}
\text{Cov}(H(f_1), H(f_2)) &= \mathbb{E}[H(f_1)H^*(f_2)] \\
&= \mathbb{E}\left[\int\int h(t_1)h^*(t_2)e^{-j2\pi(f_1t_1-f_2t_2)}dt_1dt_2\right] \\
&= \int \mathbb{E}[|h(t_1)|^2]e^{-j2\pi t_1(f_1-f_2)}dt_1 \\
&= \int P(t_1)e^{-j2\pi t_1(f_1-f_2)}dt_1 \\
&= \hat{P}(f_1-f_2)
\end{aligned}$$

### Part c

The power delay profile is given as

$$P(\tau) = \frac{1}{\tau_{rms}}e^{-\tau/\tau_{rms}} \qquad \tau > 0$$

Normalized correlation, $\rho$, is given as

$$\begin{aligned}
\rho &= \left|\frac{\mathbb{E}[H(f_1)H^*(f_2)]}{\sqrt{\mathbb{E}[|H(f_1)|^2]}\sqrt{\mathbb{E}[|H(f_2)|^2]}}\right| \\
&= |\hat{P}(f_1 - f_2)| \\
&= \left|\int P(t)e^{-j2\pi t(f_1-f_2)}dt_1\right| \\
&= \frac{1}{1+4\pi^2\tau_{rms}^2(f_1-f_2)^2}
\end{aligned}$$

For $\rho = 0.1$, minimum $(f_1 - f_2)$ can be found as

$$(f_1 - f_2)_{min} = \frac{3}{2\pi\tau_{rms}} = 47.7\text{KHz}$$

### Part d

Similarly as in (c), for $\rho = 0.9$,

$$(f_1 - f_2)_{min} = \frac{1}{6\pi\tau_{rms}} = 5.3\text{KHz}$$

## Problem 8.3

### Part a

The ergodic capacity is given as

$$\begin{aligned}
C_{erg} &= \mathbb{E}[\log_2(1 + G\cdot\text{SNR})] \\
&\leq \log_2(\mathbb{E}[1 + G\cdot\text{SNR}]) \qquad \text{Jensen Inequality} \\
&= \log_2(1+\text{SNR}) \qquad \text{As } \mathbb{E}[G] = 1
\end{aligned}$$

### Part b

The ergodic capacities of Rayleigh and AWGN channels are given as

$$C_{ray} = \int_0^\infty \log_2(1 + g\text{SNR})e^{-g}dg$$

$$C_{awgn} = \log_2(1+\text{SNR})$$

These are plotted as a function of SNR in Fig. 2.

### Part c

Suppose that the Rayleigh fading capacity at $\text{SNR}_1$ equals the AWGN capacity at $\text{SNR}_2$. We wish to find the ratio of $\text{SNR}_1/\text{SNR}_2$ as the SNRs get large in order to determine the asymptotic performance loss due to fading. We have

$$\mathbb{E}[\log_2(1 + G\cdot\text{SNR})] - \log_2(1+\text{SNR}_2) = 0$$

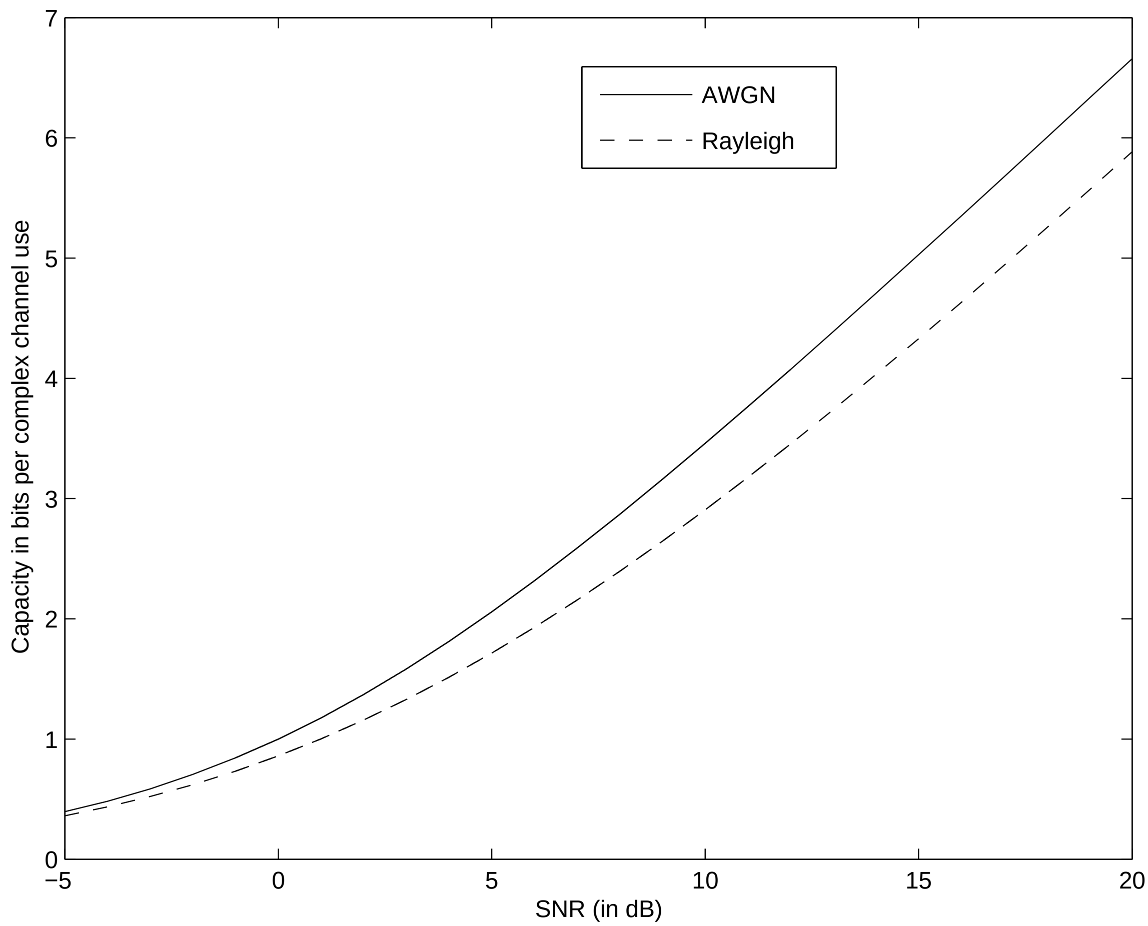



Figure 2: Capacity as a function of SNR for Problem 8.3

which implies

$$\mathbb{E}\left[\log_2\left(\frac{1+G\cdot\mathrm{SNR}}{1+\mathrm{SNR}}\right)\right]-\log_2\left(\frac{1+\mathrm{SNR}_2}{1+\mathrm{SNR}_1}\right)=0$$

Taking the limit of large SNR, we obtain that

$$\mathbb{E}[\log_2 G]=\log_2\frac{\mathrm{SNR}_2}{\mathrm{SNR}_1}$$

The left hand side is (the negative of) the well known Euler's constant $\gamma$

$$-\gamma=\mathbb{E}[\log_2 G]=\int_0^\infty \log_2(g)e^{-g}dg=-0.577$$

so that the penalty or SNR difference in dB is $10\gamma/\log_2 10 = 2.5$dB.

**Part d**

For SNR$\to 0$, $1+G{\cdot}\mathrm{SNR}\to 1$. This implies

$$C_{ray}\to\int_0^\infty \log_2 1e^{-g}dg=0$$

$$C_{awgn} \to \log_2 1 = 0$$

Hence, the penalty in dB also tends to 0.

### Part e

Waterfilling across time can be used if the transmitter knows the fading coefficeints.

## Problem 8.4

### Part a

Given $G$, the channel is AWGN. So, the cross-over probability is just the uncoded BER of BPSK on AWGN channel given as

$$p(G) = Q\left(\sqrt{\frac{2\overline{E}_b G}{N_o}}\right)$$

Hence, $C = 1 - H_B(p(G))$ using results for Binary Symmetric Channel in Chapter 6. Because of ideal interleaving, each codeword sees all instances of $G$. This implies, $C_{ergodic} = \mathbb{E}[1 - H_B(p(G))]$.

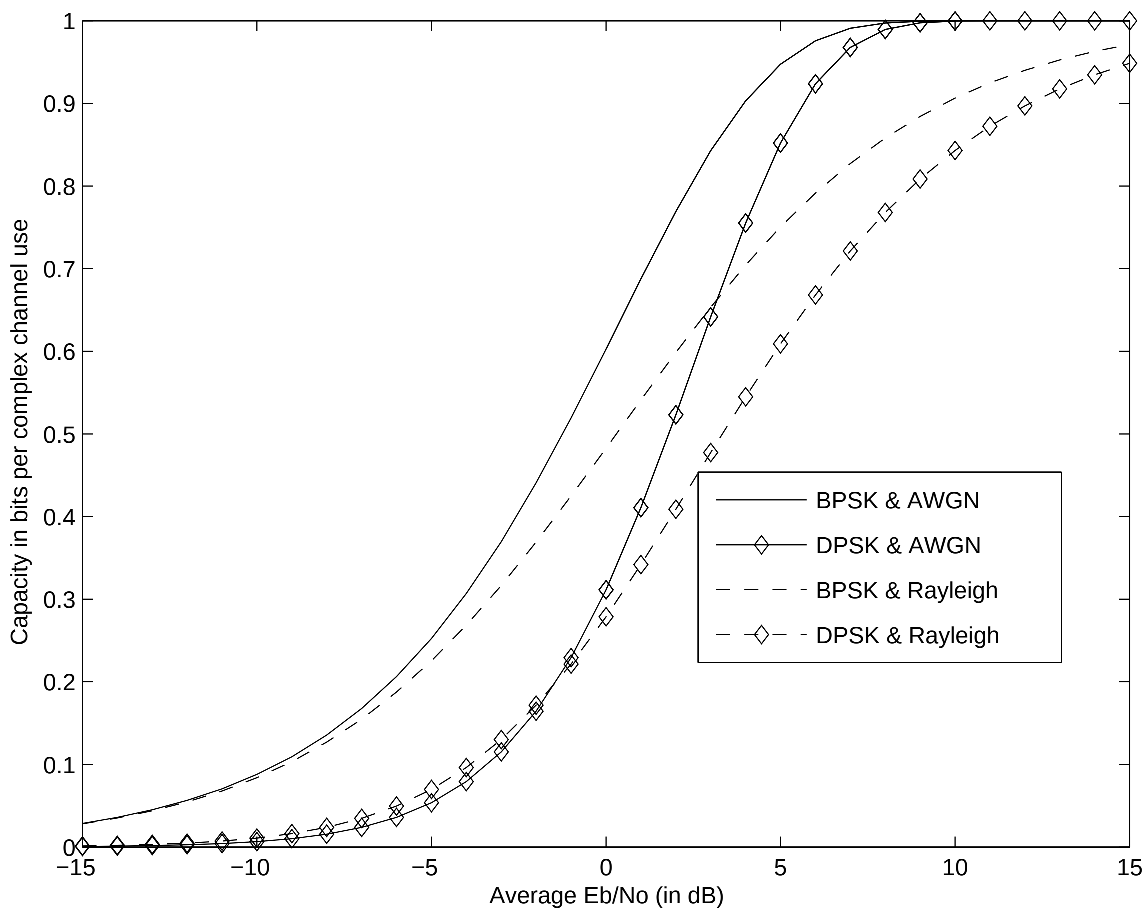



Figure 3: Capacity as a function of $\overline{E}_b/N_o$ for Problem 8.4

### Part b

For DPSK system in AWGN, from Chapter 4, uncoded BER is given as

$$P_e = \frac{1}{2}e^{-E_b/N_o}$$

By taking $E_b/N_o = G\overline{E}_b/N_o$, capacity can be found using arguments in Part a.

### Part c

The plots are shown in Fig. 3. The capacities for coherent systems in AWGN channel are obtained by replacing $G$ with 1 in the formulae derived in parts (a) and (b).

As expected, AWGN channels have higher capacity than Rayleigh channels and BPSK results in higher capacity than DPSK. However, for high SNR all capacities tend to 1 and for low SNR all capacities tend to 0.

## Problem 8.5

### Part a

$$K_X(s) = \int_0^\infty e^{(s-1)x}dx$$

The integral converges to $1/(1-s)$ only when $\mathrm{Re}(s-1) < 0$ or $\mathrm{Re}(s) < 1$.

### Part b

$$K_Y(s) = \mathbb{E}[e^{sY}] = \mathbb{E}[e^{sX_1}e^{sX_2}]$$

Since, $(X_1, X_2)$ are independent, $(e^{sX_1}, e^{sX_2})$ are also independent. Hence, $K_Y(s) = K_{X_1}(s)K_{X_2}(s)$.

### Part c

Differentiating with respect to $s$,

$$K_X^{'}(s) = \int e^{sx}xp(x)dx$$

So, $K_X^{'}(s)$ is the transform of $xp(x)$.

### Part d

The result of part (b) is repeatedly applied along with the result in part (a) to arrive at this result.

$$\begin{aligned}
K_Y(s) &= K_{X_1+X_2+\cdots X_N} \\
&= \frac{1}{1-s}K_{X_2+\cdots X_N} \qquad \mathrm{Re}(s) < 1 \qquad \text{(Using (a) and (b))} \\
&= \frac{1}{(1-s)^2}K_{X_3+\cdots X_N} \qquad \mathrm{Re}(s) < 1
\end{aligned}$$

This is continued to yield the answer.

### Part e

Let $p(y,N)$ represent the probability density function of $Y = X_1 + \cdots + X_N$. Then, from part (d),

$$p(y, N-1) \Leftrightarrow \frac{1}{(1-s)^{N-1}}, \qquad \mathrm{Re}(s) < 1$$

Using (c),

$$yp(y, N-1) \Leftrightarrow \frac{N-1}{(1-s)^{N}}, \qquad \mathrm{Re}(s) < 1$$

So $yp(y, N-1)$ has the same transform and ROC as $(N-1)p(y,N)$. Hence, $p(y,N) = (y/(N-1))p(y, N-1)$. This implies

$$p(y,N) = \frac{y}{N-1}\frac{y}{N-2}\cdots\frac{y}{1}p(y,1)$$

Since, $p(y,1)$ is just $p(y) = e^{-y}, \quad y > 0$, we get the result.

### Part f

We have $Y = X_1 + \cdots X_N = (X_1' + \cdots X_N')/\mu$, where $X_i' = \mu X_i'$ is exponential with mean 1. Let $Y' = (X_1' + \cdots X_N')$. Then, $Y = Y'/\mu$. Since, $Y'$ has density $p(y)$ as given in (e), $Y$ has density $p_Y(y) = \mu p(\mu y)$. Substituting the expression of $p(y)$ from part (e), the density can be obtained as stated in the question.

## Problem 8.6

### Part a

$$\begin{aligned} Y \leq y \quad &\Rightarrow \quad \sum_{i=1}^{N} X_i \leq y \\ &\Rightarrow \quad t_N \leq y \end{aligned}$$

But $t_N$ is the time for $N^{\text{th}}$ arrival and hence $N$ arrivals arrived before $y$. This implies $N(y) \geq N$.

### Part b

$$\begin{aligned} P(Y \leq y) \quad &\Rightarrow \quad P(N(y) \geq N) \\ P(Y \in (y, y+\delta] \quad &\Rightarrow \quad 1 - P(N(y) < N) \\ &\Rightarrow \quad 1 - e^{-\mu y} \sum_{k=0}^{N-1} \frac{(\mu y)^k}{k!} \end{aligned}$$

### Part c

We need to show equivalence in both directions.

$$\begin{aligned} Y \in (y, y+\delta] &\Rightarrow t_N \in (y, y+\delta] \\ &\Rightarrow N(y) \leq N-1, N(y+\delta) \geq N \\ &\Rightarrow N(y) < N, N(y+\delta) \geq N \end{aligned}$$

Also,

$$N(y) < N, N(y+\delta) \geq N \Rightarrow t_N > y, t_N \leq y+\delta$$

Hence,

$$Y \in (y, y+\delta] \Leftrightarrow N(y) < N, N(y+\delta) \geq N$$

### Part d

Since $N(y) < N, N(y+\delta) \geq N$, atleast one increment has to happen in $(y, y+\delta]$. This implies,

$$P(N(y+\delta) - N(y) = k) = \frac{(\mu\delta)^k}{k!} e^{-\mu\delta}$$

For $k \geq 2$ and $\delta \to 0$,

$$\frac{P(N(y+\delta) - N(y) = k)}{P(N(y+\delta) - N(y) = 1)} = \frac{(\mu\delta)^{k-1}}{(k-1)!} \to 0$$

So, one increment is much more dominating for small $\delta$. Hence, $(N(y) = N-1, N(y+\delta) = N)$ is the most probable event.

### Part e

$$\begin{aligned} P(Y \in (y, y+\delta]) &\approx P(N(y) = N-1, N(y+\delta) = N) \\ &= P(N(y) - N(0) = N-1, N(y+\delta) - N(y) = 1) \qquad N(0) = 0 \end{aligned}$$

By independent increment property,

$$P(Y \in (y, y+\delta]) \approx P(N(y) - N(0) = N-1) \cdot P(N(y+\delta) - N(y) = 1)$$

Using the expression for the pmf of increments,

$$P(Y \in (y, y+\delta]) \approx \mu e^{-\mu y} \frac{(\mu y)^{N-1}}{(N-1)!} \delta$$

As, $P(Y \in (y, y+\delta]) = p(y)\delta$, we can obtain $p(y)$ as in 8.139.

## Problem 8.7

### Part a

The indefinite integral can be integrated by parts as

$$\begin{aligned}\Gamma(x) &= \int t^{x-1}e^{-t}dt \\ &= -t^{x-1}e^{-t} + (x-1)\int t^{x-2}e^{-t}dt\end{aligned}$$

Substituting the limits 0 and $\infty$, we get the result $\Gamma(x) = (x-1)\Gamma(x-1)$.

### Part b

$$\Gamma(1) = \int_0^\infty e^{-t}dt = 1$$

### Part c

$$\begin{aligned}\Gamma(N) &= (N-1)\Gamma(N-1) \\ &= (N-1)(N-2)\Gamma(N-2) \\ &= (N-1)(N-2)\cdots 1\Gamma(1) \\ &= (N-1)!\end{aligned}$$

### Part d

$$\Gamma(\frac{1}{2}) = \int_0^\infty t^{-\frac{1}{2}}e^{-t}dt$$

Substitution $t = z^2/2$ implies

$$\begin{aligned}\Gamma(\frac{1}{2}) &= \sqrt{2}\int_0^\infty e^{-\frac{z^2}{2}}dz \\ &= 2\sqrt{\pi}\int_0^\infty \frac{1}{\sqrt{2\pi}}e^{-\frac{z^2}{2}}dz \\ &= 2\sqrt{\pi}\cdot\frac{1}{2} \\ &= \sqrt{\pi}\end{aligned}$$

### Part e

Using the expression of the density function of $Y$ from Problem 8.5, we have

$$\begin{aligned}
\mathbb{E}[\sqrt{Y}] &= \mu \int_0^{\infty} \sqrt{y} \frac{(\mu y)^{N-1}}{(N-1)!} e^{-\mu y} dy \\
&= \frac{\mu^N}{(N-1)!} \int_0^{\infty} y^{N-\frac{1}{2}} e^{-\mu y} dy \\
&= \frac{\mu^N}{(N-1)!} \Gamma \left( N + \frac{1}{2} \right)
\end{aligned}$$

## Problem 8.8

### Part a

Received output at each element is given as

$$r_i = h_i x + n_i$$

where $n_i \sim CN(0, N_o)$ and $h_i \sim CN(0, 1)$. Then, we have

$$e_b = \mathbb{E}[|h_i|^2 ||x||^2] = ||x||^2$$

Similarly, $E_b = N||x||^2$. On maximal ratio combining, we get,

$$Z = \left( \sum_{i=1}^{N} |h_i|^2 \right) x + \sum_{i=1}^{N} h_i^* n_i$$

where the real part of $Z$, represents the decision statistic. Taking the real parts, this can be simplified as

$$\Re Z = Yx + n_1 \qquad \text{where } n_1 \sim N(0, N_o Y/2$$

where $Y = \sum_{i=1}^{N} |h_i|^2$. So, the probability of error for BPSK alphabet $x$ can be written as,

$$\begin{aligned}
P_e &= \mathbb{E} \left[ Q \left( \sqrt{\frac{Y^2 ||x||^2}{\frac{N_o Y}{2}}} \right) \right] \\
&= \mathbb{E}[Q(\sqrt{aY})]
\end{aligned}$$

with $a = 2e_b/N_o = 2E_b/(N_o N)$.

### Part b

Using the bound, we have

$$P_e = \mathbb{E}[Q(aY)] \leq \frac{1}{2} \mathbb{E}[e^{-\frac{aY}{2}}]$$

The expectation can now be evaluated using the Laplace transform of the density function of $Y$ derived in Problem 8.5 (d) as

$$P_e \leq \frac{1}{2} \mathbb{E}[e^{-\frac{aY}{2}}] = \frac{1}{2\left(1 + \frac{a}{2}\right)^N}$$

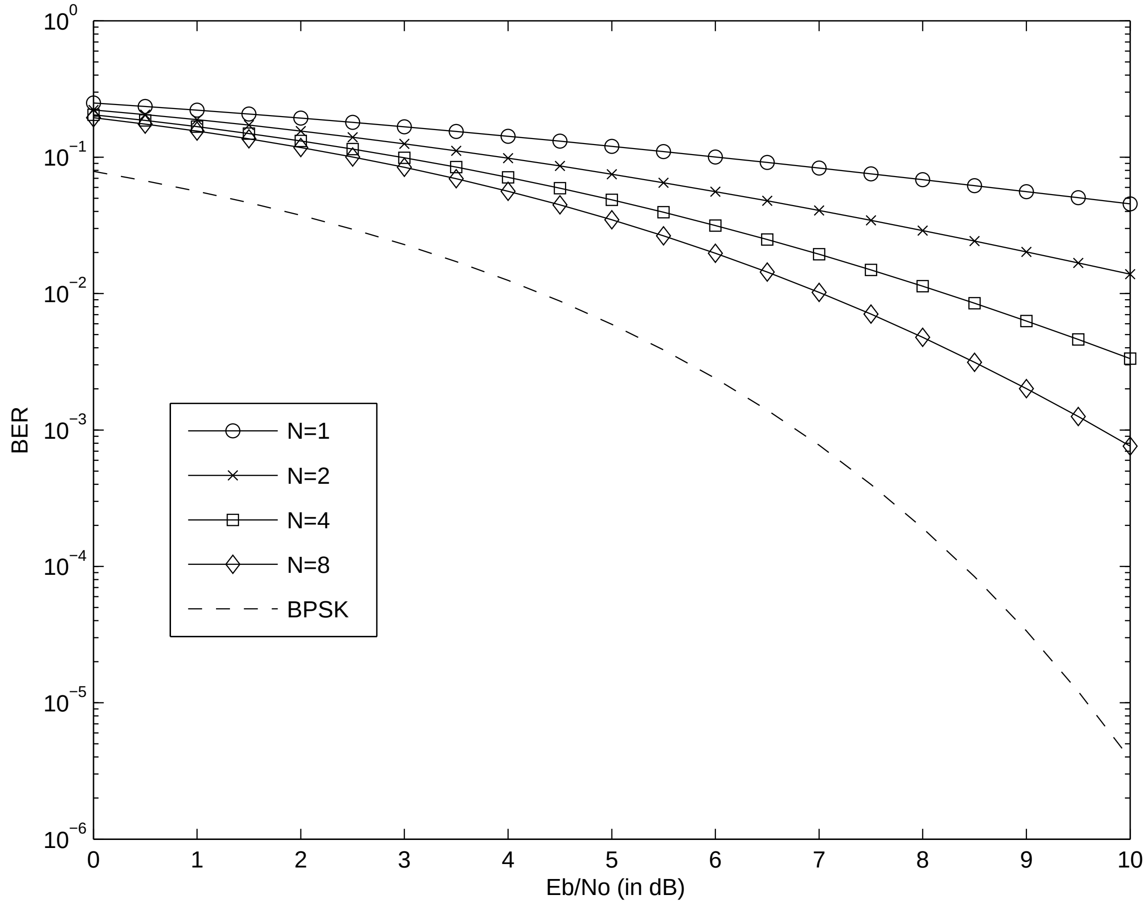



Figure 4: Rayleigh fading with receive diversity : BER as a function of $E_b/N_o$ for Problem 8.8

### Part c

Fig. 4 depicts the plot of BER versus $E_b/N_o$ for Rayleigh fading with various receive diversity orders. Also shown is the exact BER of BPSK. It can be seen that as receive diversity, $N$, increases, BER decreases and the graph becomes closer to the graph of BPSK. It has to be noted that the comparision is not fair as graphs for Rayleigh fading correspond to bounds on BER while the graph of BPSK represents the exact BER.

## Problem 8.9

### Part a

Easy

### Part b

It is easy to see that the first term in the expression for $P_e$ evaluates to 0. To evaluate the second term we use the definition of $Q(x)$,

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-x^2} 2dx$$

Hence, we have,

$$\frac{d}{dy}(yQ(\sqrt{ay})) = Q(\sqrt{ay}) - \frac{\sqrt{ay}}{2\sqrt{2\pi}} e^{-\frac{ay}{2}}$$

Using this expression in the result of part (a) gives the result for part (b).

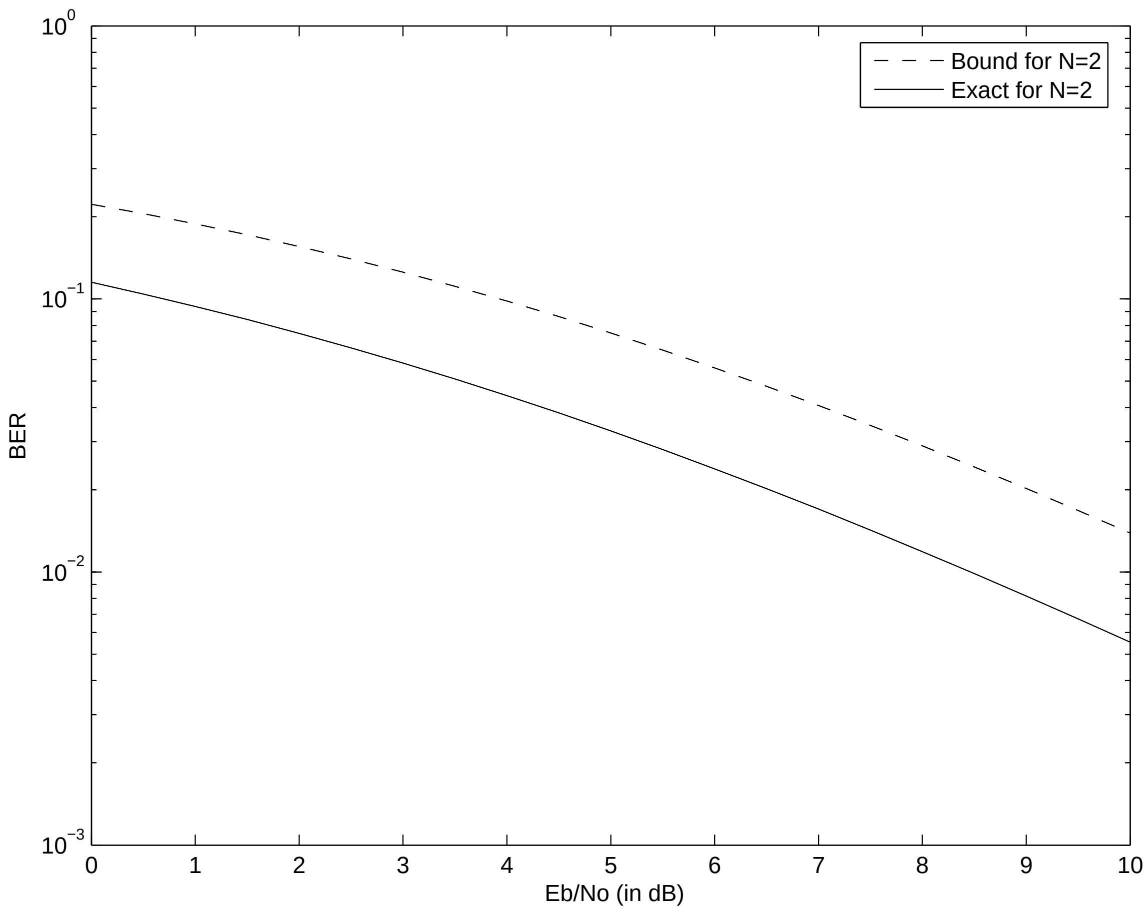



Figure 5: BER (exact and bound) as a function of $E_b/N_o$ for Problem 8.9

### Part c

The second term is given as,

$$S = \frac{a}{2\sqrt{2\pi}} \int_0^\infty y^{\frac{1}{2}} e^{-\left(1+\frac{a}{2}\right)y} dy$$

Using substitution $z = \left(1 + \frac{a}{2}\right) y$, we have,

$$\begin{aligned} S &= \frac{\sqrt{a}}{2\sqrt{2\pi}} \left(1 + \frac{a}{2}\right)^{-3/2} \int_0^\infty z^{\frac{1}{2}} e^{-z} dz \\ &= \frac{\sqrt{a}}{2\sqrt{2\pi}} \left(1 + \frac{a}{2}\right)^{-3/2} \Gamma\left(\frac{3}{2}\right) \\ &= \frac{\sqrt{a}}{4\sqrt{2}} \left(1 + \frac{a}{2}\right)^{-3/2} \\ &= \frac{1}{2a} \left(1 + \frac{a}{2}\right)^{-3/2} \end{aligned}$$

Substituting $S$ for the second term in the result for part (b) yields the result for part (c).

**Part d**

Plot of the exact BER and the bound on BER is shown in Fig. 5.

**Part e**

For $N = 1$, we have,

$$\begin{aligned} P_e &= \frac{1}{2}\left(1 - \left(1 + \frac{2}{a}\right)^{-\frac{1}{2}}\right) \\ &= \frac{1}{2}\left(1 - \left(1 - \frac{1}{a}\right)\right) \\ &= \frac{1}{2a} \end{aligned}$$

For $N = 2$, we have,

$$\begin{aligned} P_e &= \frac{1}{2}\left(1 - \left(1 + \frac{2}{a}\right)^{-\frac{1}{2}} - \frac{1}{a}\left(1 + \frac{2}{a}\right)^{-\frac{3}{2}}\right) \\ &= \frac{1}{2}\left(1 - \left(1 - \frac{1}{a}\right) - \frac{1}{a}\left(1 - \frac{3}{a}\right)\right) \\ &= \frac{3}{2a^2} \end{aligned}$$

**Part f**

Using the density function of $Y$, we have

$$\begin{aligned} P_e(N) &= \int_0^\infty Q(\sqrt{ay}) \frac{y^{N-1} e^{-y}}{(N-1)!} dy \\ &= -e^{-y} Q(\sqrt{ay}) \frac{y^{N-1}}{(N-1)!}\Big|_0^\infty + \int_0^\infty \left\{ Q(\sqrt{ay}) \frac{y^{N-2}}{(N-2)!} - \frac{\sqrt{a} e^{-ay/2} y^{N-\frac{3}{2}}}{2\sqrt{2\pi}} \right\} e^{-y} dy \\ &= P_e(N-1) - \int_0^\infty \frac{\sqrt{a} e^{-ay/2} y^{N-\frac{3}{2}}}{2\sqrt{2\pi}} e^{-y} dy \end{aligned}$$

Now, $f(N)$ the second term in the expression of $P_e(N)$ can be simplified by substituion $z = y(1+a/2)$ as

$$\begin{aligned} f(N) &= \frac{\sqrt{a}}{2\sqrt{2\pi}}\left(1+\frac{a}{2}\right)^{N-\frac{1}{2}}\int_0^\infty e^{-z}z^{N-\frac{3}{2}}dz \\ &= \frac{\sqrt{a}}{2\sqrt{2\pi}}\left(1+\frac{a}{2}\right)^{N-\frac{1}{2}}\Gamma\left(N-\frac{1}{2}\right) \end{aligned}$$

## Problem 8.10

### Part a

For $y > 0$, we have,

$$\begin{aligned} F_Y(y) &= P(Y \le y) \\ &= P(\max(X_1, X_2, \cdots, X_N) \le y) \\ &= P(X_1 \le y, X_2 \le y, \cdots, X_N \le y) \\ &= P(X_1 \le y)P(X_2 \le y)\cdots P(X_N \le y) \qquad \text{as } X_i \text{ are independent} \\ &= F_X^N(y) \end{aligned}$$

As $X_i$ are exponential with mean 1, we have

$$F_Y(y) = (1-e^{-y})^N \qquad y > 0$$

### Part b

Differentiating $F_Y(y)$, we get,

$$f_Y(y) = Ne^{-y}(1-e^{-y})^{N-1} \qquad y > 0$$

For $N = 2$, we have,

$$f_Y(y) = 2(e^{-y} - e^{-2y}) \qquad y > 0$$

### Part c

$$P_e = 2\int_0^\infty Q(\sqrt{ay})e^{-y}dy - 2\int_0^\infty Q(\sqrt{ay})e^{-2y}dy$$

The first term in $P_e$ can be found directly using 8.144 while for the second term, the substitution $z = 2y$ can be used to convert it to the form of 8.144.

$$\begin{aligned} P_e &= 2\left[1 - \frac{1}{2}\left(1+\frac{2}{a}\right)^{-\frac{1}{2}}\right] - \int_0^\infty Q\left(\sqrt{\frac{az}{2}}\right)e^{-z}dz \\ &= \frac{1}{2} - \left(1+\frac{2}{a}\right)^{-\frac{1}{2}} + \frac{1}{2}\left(1+\frac{4}{a}\right)^{-\frac{1}{2}} \end{aligned}$$

### Part d

Using the given approximation of $(1+x)^b$, we have,

$$\begin{aligned} P_e &\approx \frac{1}{2} - \left(1 - \frac{1}{a} + \frac{3}{2a^2}\right) + \frac{1}{2}\left(1 - \frac{2}{a} + \frac{6}{a^2}\right) \\ &= \frac{3}{2a^2} \end{aligned}$$

## Problem 8.11

### Part a

Since, $Y = X_1 + \cdots + X_N$ with $X_i$ i.i.d exponential with mean 1, we have

$$\begin{aligned} P(Y < 0.1\mathbb{E}[Y]) &= P(X_1 + \cdots + X_N < 0.1N \\ &\leq e^{-\beta N} \end{aligned}$$

with

$$\beta = \min_{s<0} f(s) \qquad \text{where } f(s) = \left(M(s) - \frac{s}{10}\right)$$

The moment generating function for $X_1$, $M(s)$, is given as

$$M(s) = log\mathbb{E}[e^{sX_1}] = \log \int_0^\infty e^{sx_1} e^{-x_1} dx_1 = -\log(1-s)$$

The minimum value of $f(s)$ is found as follows

$$\frac{\partial f}{\partial s} = 0 \Rightarrow s = -9$$

It can be verified that $\partial^2 f/\partial s^2 > 0$ for $s = -9$. Hence, $\beta$ can be found as 1.

### Part b

$$e^{-N} < 10^{-3} \Rightarrow N > 3\log_e 10 \Rightarrow N \geq 7$$

### Part c

Using the definition of cdf $F_Y(y)$, we have

$$P(Y < 0.1\mathbb{E}[Y]) = 1 - F_Y(0.1N)$$

Using the expression of $F_Y(y)$ from problem 8.6 with $\mu = \mathbb{E}[X_1] = 1$, we have $P(Y < 0.1\mathbb{E}[Y]) = 1.8e - 11$.

## Problem 8.12

### Part a

Since $X_1 = V_1 - U_1$ with $V_1$ and $U_1$ independent, we have

$$\mathbb{E}[e^{sX_1}] = \mathbb{E}[e^{sV_1}]\mathbb{E}[e^{-sU_1}]$$

Since, $V$ is exponential random variable with mean $1/\mu_V$, we have

$$\mathbb{E}[e^{sV_1}] = \mu_V \int_0^\infty e^{sv-\mu_V v} dv = \frac{\mu_V}{\mu_V - s} \qquad s < \mu_V$$

Similarly,

$$\mathbb{E}[e^{-sU_1}] = \frac{\mu_U}{\mu_U + s} \qquad s > -\mu_U$$

Substituting $\mathbb{E}[e^{sV_1}]$ and $\mathbb{E}[e^{-sU_1}]$ in $\mathbb{E}[e^{sX_1}]$ yields the result with range of $s$ to be $(-\mu_U, \mu_V)$.

### Part b

Using result in part (a), we have

$$M^*(0) = \min_{s>0} \log(\mu_V\mu_U) - \log(\mu_V - s) - \log(\mu_U + s)$$

Solving $\partial M^*(0)/\partial s = 0$, we get $s^* = (\mu_V = \mu_U)/2$. It can be verified that $\partial^2 M^*(0)/\partial s^2 > 0$ at $s = s^*$ implying the minimizing $s = s^*$.

### Part c

Using the result of part (b), we have

$$M^*(0) = \log\left(\frac{\mu_V\mu_V}{\left(\frac{\mu_V+\mu_U}{2}\right)^2}\right)$$

Simplifying, we get

$$M^*(0) = 2\log\left(\frac{\sqrt{\mu_V\mu_V}}{\frac{\mu_V+\mu_U}{2}}\right)$$

Hence, $P_e$ is given as $e^{-\alpha N}$ with $\alpha = -M^*(0)$.

### Part d

Substituting the values of $\mu_U$ and $\mu_V$, we get

$$\alpha = -2\log\left(\frac{\sqrt{4(2\sigma^2E(2\sigma^2E + E^2))}}{E^2 + 4\sigma^2E}\right)$$

Simplifying, we get

$$\alpha = \frac{\left(1 + \frac{E}{4\sigma^2}\right)^2}{1 + \frac{E}{2\sigma^2}}$$

Substituting $\alpha$ with $E = \overline{e}_b$ and $\sigma^2 = N_o/2$, we get the result in 8.38.

## Problem 8.13

### Part a

Consider,

$$(h \odot g_n)[k] = \sum_{m=0}^{N-1} h[m] e^{j2\pi n([k-m]\text{mod}N)/N}$$

Since 0 $leqk, m \leq N-1$, $kmodN = k$ and $mmodN = m$. Using this fact, we get

$$(h \odot g_n)[k] = e^{j2\pi nk/N} \sum_{m=0}^{N-1} h[m] e^{-j2\pi nm/N}$$

Using the definition of DFT, $H[n]$, we have

$$(h \odot g_n)[k] = g_n[k]H[n] \qquad \forall 0 \leq k \leq N-1$$

### Part b

Using the inverse DFT formula, we have

$$g[k] = \sum_{n=0}^{N-1} G[n] e^{j2\pi nk/N}$$

Since $g_n[k] = e^{j2\pi nk/N}$, we have

$$g[k] = \sum_{n=0}^{N-1} G[n] g_n[k] \qquad \forall \quad 0 \leq k \leq N-1$$

### Part c

The procedure is clearly mentioned in the problem.

## Problem 8.14

### Part a

The signal $u(t)$ can be rewritten using change of index $m = n - (N-1)/2$ as

$$u(t) = \sum_{m=-(N-1)/2}^{(N-1)/2} B\left[m + \frac{N-1}{2}\right] e^{j2\pi mt/T} I_{[0,T]}$$

Now, $u(t)$ is real iff $u(t) = u^*(t)$. $u^*(t)$ can be simplified using change of index $m' = -m$ as

$$\begin{aligned} u^*(t) &= \sum_{m=-(N-1)/2}^{(N-1)/2} B^*\left[m + \frac{N-1}{2}\right] e^{-j2\pi mt/T} I_{[0,T]} \\ &= \sum_{m'=-(N-1)/2}^{(N-1)/2} B^*\left[-m' + \frac{N-1}{2}\right] e^{j2\pi m't/T} I_{[0,T]} \end{aligned}$$

Equating $u(t) = u^*(t)$, we get,

$$\sum_{m=-(N-1)/2}^{(N-1)/2} \left( B\left[m + \frac{N-1}{2}\right] - B^*\left[-m + \frac{N-1}{2}\right]\right) e^{j2\pi mt/T} I_{[0,T]} = 0$$

Since, $e^{j2\pi mt/T}$ are orthogonal functions over $[0, T]$, all coefficients in the above summation have to be zero implying

$$B\left[m + \frac{N-1}{2}\right] = B^*\left[-m + \frac{N-1}{2}\right]$$

Reverting back to the old index $n$, we get the constraint on $B[n]$ to be

$$B[n] = B * [N-1-n] \qquad \forall \quad 0 \leq n \leq N-1$$

It can be seen that this condition doesnot require $B[n]$ to be real. Hence, $B[n]$ can be chosen from a complex constellation. Theoretically $I_{[0,T]}$ (Fourier Transform : $T$ Sinc$(Tf)$) has infinite bandwidth and hence $u(t)$ also has infinite bandwidth. But, the bandwidth of the baseband signal $u(t)$ is approximately $N/2T$ neglecting the sidelobes of Sinc function.

### Part b

Bandwidth is half the Nyquist sampling rate and hence, the bandwidth is 1MHz. So, $N/2T =$ 1MHz. Hence, the inter-carrier spacing is $1/T \approx 3.91$KHz. For a channel of delay spread $LT_s$ ($T_s$ is sampling time which equals 1/2MHz = $0.5\mu$s), we need $(L-1)$ length prefix. Hence, $L = 51$. Maximum delay spread the system is designed for is $25.5\mu$s.

## Problem 8.15

### Part a

Maximum delay spread of the channel = $2\mu$s. Defining the coherence bandwidth, $W_c$, as the inverse of this quantity, we have $W_c = 0.5M$Hz. The fourier transform is given as

$$H(f) = 1 - 0.7je^{-j2\pi f} - 0.4e^{-j4\pi f}$$

Fig. 6 shows the plot of $|H(f)|$ vs $f$ over frequency range $[-5, 5]$MHz.

### Part b

First for fair comparision, $H(f)$ is normalized by $\int_{-5\text{M}}^{5\text{M}} |H(f)|^2 df$ in the frequency band is properly normalized. Input SNR equals 10. This means,

$$\frac{\int S_X(f)df}{\int S_N(f)df} = 10 \tag{1}$$

where, $S_X(f)$ and $S_N(f)$ represent the transmitted signal and noise power spectral densities (PSD). Since noise and transmitted signal are white, $S_X(f)$ and $S_N(f)$ are constant w.r.t $f$. This implies,

$$\frac{S_X(f)}{S_N(f)} = 10$$

From Chapter 6, Shannon capacity is given as

$$C = \int_{-5\text{ M}}^{5\text{M}} \log_2 \left(1 + \frac{|H(f)|^2 S_X(f)}{S_N(f)}\right) df = 8.61\text{Mbps}$$

For the non-dispersive channel, $|H(f)| = 1$. This implies,

$$\frac{S_X(f)}{S_N(f)} = 10$$

$$C = \int_{-5\text{M}}^{5\text{M}} \log_2(1 + 10) df = 34.6\text{Mbps}$$

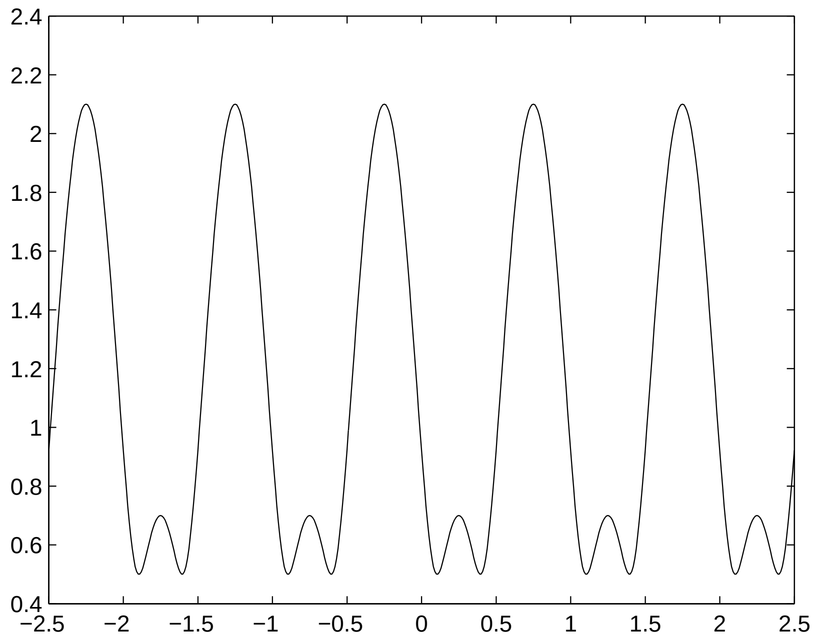



Figure 6: Plot of $|H(f)|$ for Problem 8.15

### Part c

Using the waterfilling solution as given in chapter 6, the capacity is found as $C = 10.57$Mbps. Waterfilling does improve the capacity but it is still far from the non-dispersive case.

### Part d

The effective SNR, $S_X(f)|H(f)|^2/S_N(f)$, experienced for part (b) as a function of frequency is plotted in fig. 7. Over the band of each sub-carrier, the effective SNR should remain approximately constant. A good choice is 32 or 64 sub-carrier system. The constellation to be used

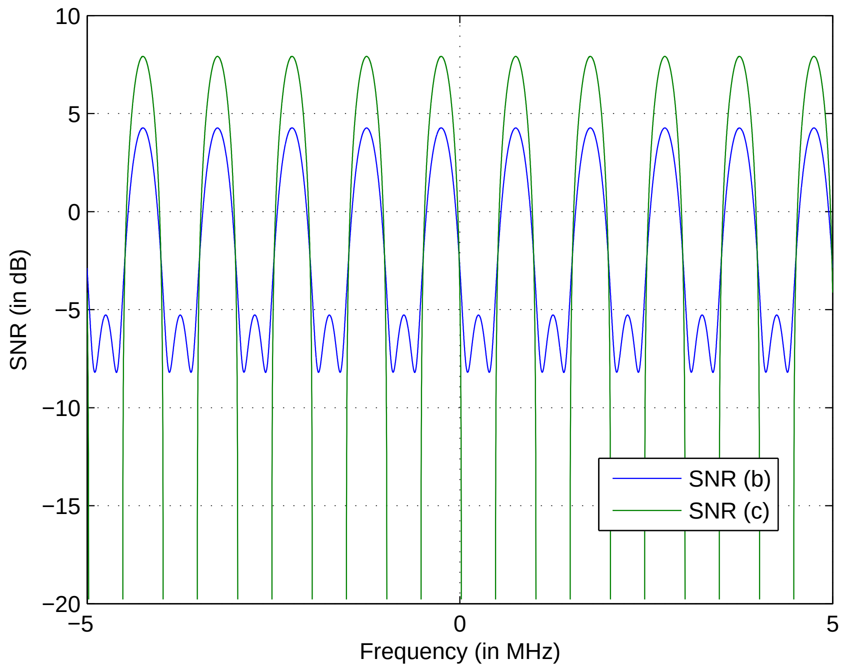



Figure 7: Figure for Problem 8.15 (d)

depends on the value of effective SNR available for each sub-carrier. The value of capacity for different constellations as a function of SNR can be obtained from Chapter 6. We need to use the same constellation as the channel is unknown for case (b). For SNR< 5dB, QPSK is a good choice to achieve AWGN capacity. 16-QAM can be used for even better performance.

In case (c), the effective SNR, $S_X(f)|H(f)|^2/S_N(f)$, experienced for the waterfilling solution as a function of frequency is also plotted in fig. 7. 32 or 64 sub-carrier system is a good choice. Again, for bands where SNR $\approx$ 5dB, QPSK or 16 QAM can be used and no signalling on bands with no power allocation.

## Problem 8.16

### Part a

Using equation (8.8), for 90% of the power to lie within a delay of $d_o$, we need

$$\int_0^{d_o} \tau_{rms} e^{-\tau/\tau_{rms}} d\tau = 0.9$$

where, $\tau_{rms} = 100$ns. Integrating the above equation, we get $d_o = 100 \log_e(10) \approx 230$ns. Let $1/T$ be the intercarrier spacing. To combat the delay $d_o$, the length of cyclic prefix to be used is $L = d_o/T_s$, with $T_s$ as the sampling time which equals $T/N$. Hence, the overhead is given as

$$L/N = d_o/T$$

It can be seen that $L$ depends only on the sub-carrier spacing. For $1/T = 100$KHz, overhead is 2.3%.

### Part b

Same as the result of (a)

### Part c

Half of the result of (a)

## Problem 8.17

### Part a

Plots are shown in Fig. 8 and Fig. 9.

### Part b

The histogram of PAR is shown in Fig. 10. Also, we have from simulations, probablility that PAR exceeds 10dB as 0.0025. The theoretical result is $64e^{-10} = 0.0029$ which is very close. The PAR histogram doesnot looks Gaussian. This is expected because using the approximate CDF of PAR, $Pr(PAR < x)$, we can calculate the probability density function which can be shown to be different from Gaussian.

### Part c

For same symbol energy, the results are shown in Fig. 11, Fig. 12 and Fig. 13. The results look similar to (a) and (b) indicating weak dependence on constellation. Also, we have from simulations, probablility that PAR exceeds 10dB as 0.004 which is again close to the theoretical result given in (b).

### Part d

From simulations, the symbol error rate (SER) for clip levels 0dB and 3dB for 16-QAM are given as 0.25 and 0.005 respectively. For clip level of 10dB, no symbol errors were observed in 1000000 symbols suggesting very low SER.

### Part e

From simulations, for symbol error rate of 1%, the clip level needed for 16-QAM is 2.7dB.

### Part f

From simulations, the symbol error rate for a clip level of 0dB for QPSK is given as $3e-5$. For clip levels of 3dB and 10dB, no symbol errors were observed in 1000000 symbols suggesting low SER. From simulations, for symbol error rate of 1%, the clip level needed is -4.8dB. This looks counter-intuitive. It has to be noted that information is stored in the phase of the signal in

QPSK and clipping did not have significant effect on the signal's phase. But most importantly, the presence of small amount of noise might completely change the situation and result in large symbol error rates for the signal.

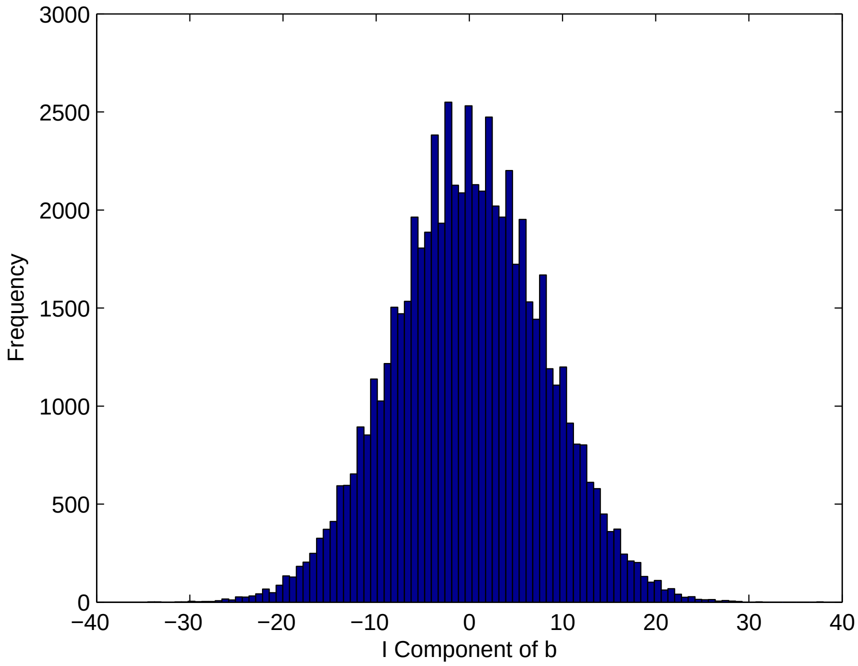



Figure 8: Histogram of the $I$ component of time-domain samples for Problem 8.17 (a)

## Problem 8.18

### Part a

Let $< f(t), g(t) >= \int f(t)g^*(t)dt$, $||f(t)||^2 = \int |f|^2(t)dt$ and $f_D(t) = f(t-D)$. For AWGN, we have the joint ML estimate of $(A, D)$ as

$$(\hat{A}, \hat{D}) = \text{argmin}||y - As_D||^2 dt \tag{2}$$

R.H.S can be simplified as

$$||y||^2 + |A|^2||s_D||^2 - A < s_D, y > -A^* < s_D, y >^* \tag{3}$$

Clearly, $||s|| = ||s_D||$. To find the minimum of (3) w.r.t $A$, differentiate w.r.t real and imaginary parts of $A$ to obtain,

$$2\Re(A)||s||^2 - 2\Re(< y, s_D >) = 0$$

$$2\Im(A)||s||^2 - 2\Im(< y, s_D >) = 0$$

Hence, $\hat{A}$ can be obtained as $\hat{A} =< y, s_d >^* /||s||^2$. Using $\hat{A}$ in (2), we have $\hat{D} = \text{argmin}||y||^2 - | < y, s_D > |^2/||s||^2$. This is equivalent to $\hat{D} = \text{argmax}| < y, s_D > |^2$ which is required to be shown.

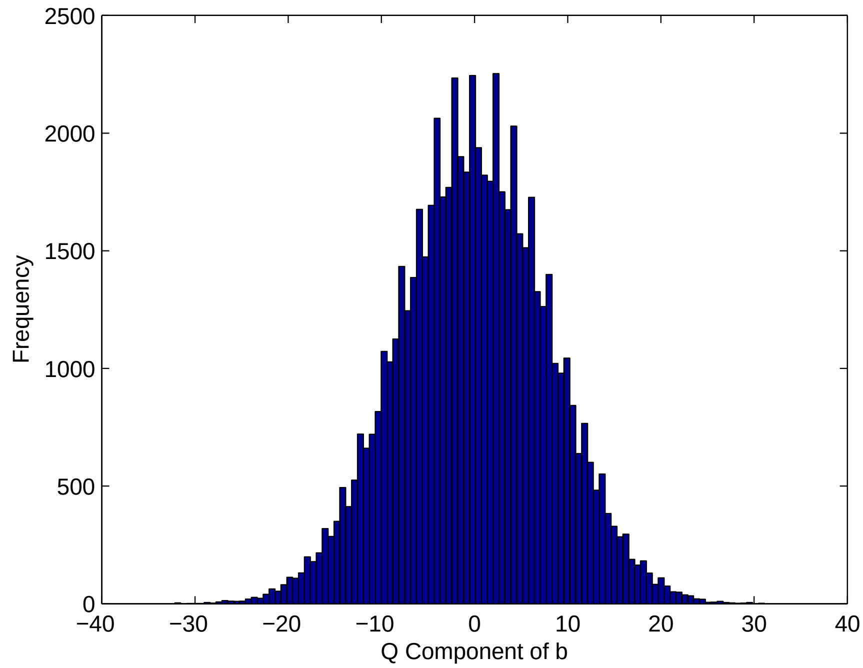



Figure 9: Histogram of the $Q$ component of time-domain samples for Problem 8.17 (a)

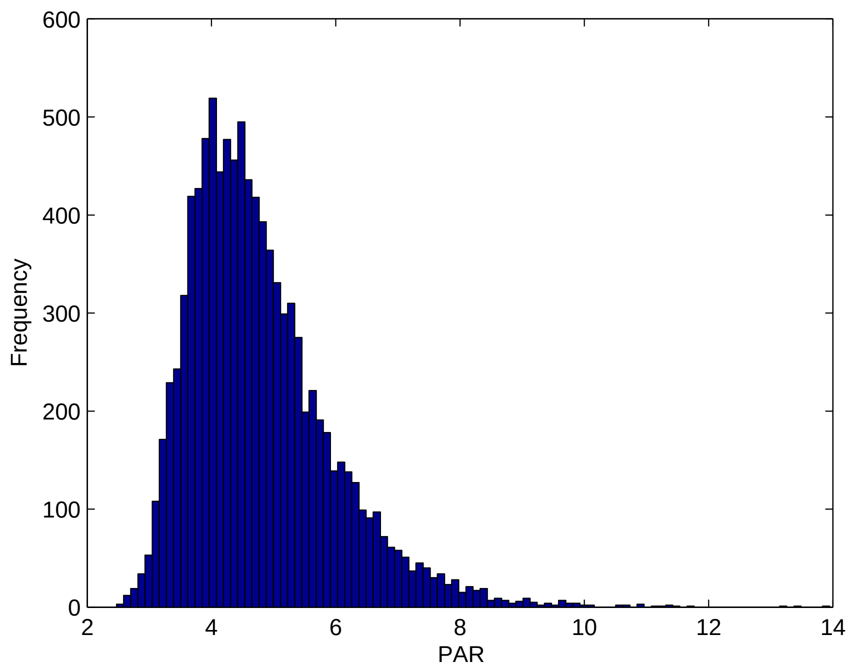



Figure 10: Histogram of PAR with QPSK for Problem 8.17 (b)

### Part b

$Z(u)$ can be written as $Z(u) = A < s_D, s_u > + < n, s_u >$. Signal part $A < s_D, s_u >$ clearly equals $R_s(D-u)$ with $R_s$ denoting the auto-correlation function of $s(t)$.

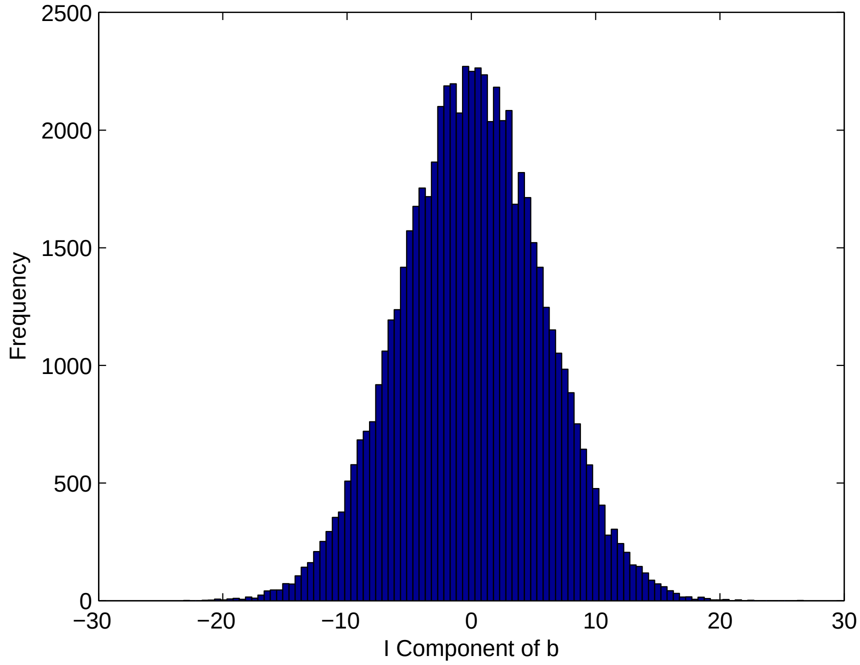



Figure 11: Histogram of the $I$ component of time-domain samples with 16-QAM for Problem 8.17 (c)

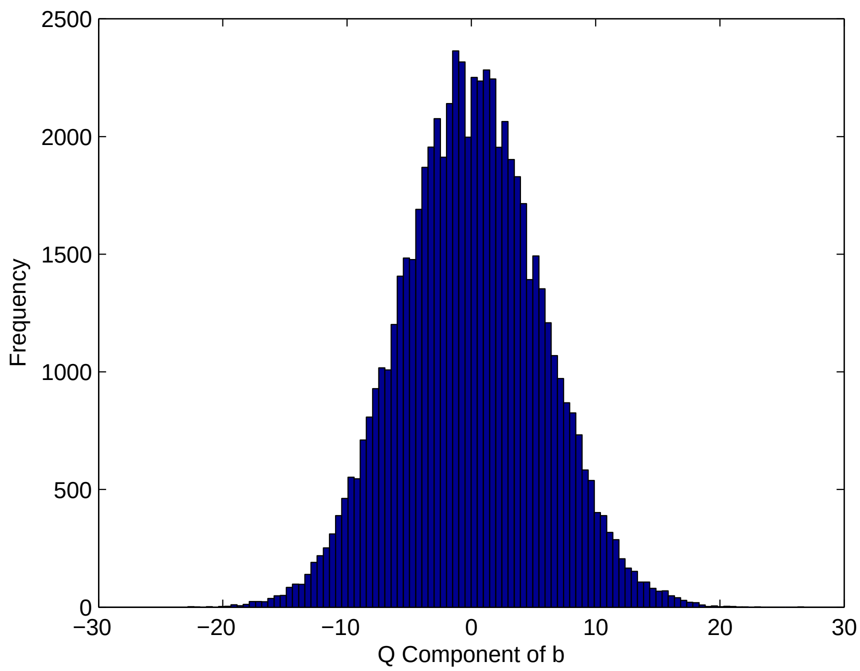



Figure 12: Histogram of the $I$ component of time-domain samples with 16-QAM for Problem 8.17 (c)

## Part c

Using the given assumptions, it can be shown

$$Z(kT_c) = \begin{cases} A||s||^2 + < n, s_{kT_c} >, & k = K \\ < n, s_{kT_c} >, & k \neq K \end{cases}$$

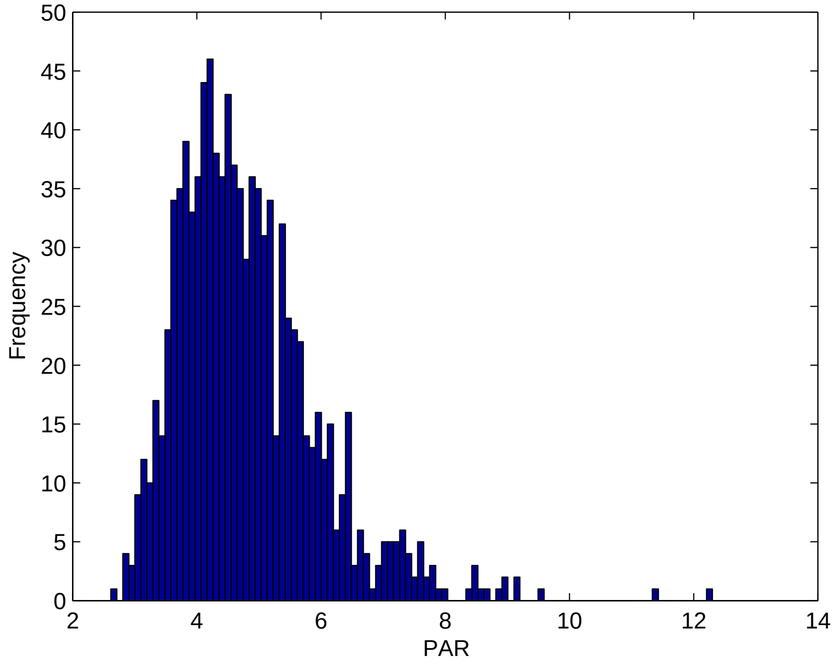



Figure 13: Histogram of PAR with 16-QAM for Problem 8.17 (c)

In statistical sense, $Z(kT_c)$ is equivalent to $||s||CN(A||s||, N_o)$ for $k = K$ and $||s||CN(0, N_o)$ for $k \neq K$. Also, $\mathbb{E}[Z(k_1T_c)Z(k_2T_c)] = 0$ for $k_1 \neq k_2$ because using given assumptions on $s(t)$, it implies $< s_{k_1T_c}, s_{k_2T_c} >= 0$. Hence, $Z(kT_c)$ for $k = \{0, \cdots, M-1\}$ are independent. This implies,

$$\begin{aligned} Pr(\hat{k} \neq K) &= Pr(\text{argmax}|Z(kT_c)|^2 \neq K) \\ &= 1 - Pr(\text{argmax}|Z(kT_c)|^2 = K) \\ &= 1 - \prod_{k \neq K} Pr(|Z(KT_c)|^2 > |Z(kT_c)|^2) \end{aligned} \tag{4}$$

Using the proof of proposition 4.5.6 (Error probability for orthogonal signaling), $Pr(|CN(\sqrt{E_s}, N_o)| < |CN(0, N_o)|) = 0.5e^{-E_s/2N_o}$. Hence, the probability in (4) becomes $1 - (1 - 0.5e^{-E_s/2N_o})^{M-1}$.

**Part d**

The ML estimation problem can be written as

$$(\hat{A}, \hat{K}, \hat{\delta}) = \text{argmin}||y - As_{kT_c+\delta}||^2 dt$$

Like in part (a), the estimate of $A$ can be found as $\hat{A} =< y, s_{kT_c+\delta} > / < s, s >$. This implies $(\hat{K}, \hat{\delta}) = \text{argmax}|Z(kT_c + \delta)|^2$ with $Z(u) =< y, s_u >$.

*Sub-optimal estimators:*
Since, in the case of no noise, only $Z(KT_c)$ and $Z((K+1)T_c$ can be non-zero, an estimator of $K$ is given as

$$\hat{K} = \text{argmax}|Z(kT_c)| + |Z((k+1)T_c)|$$

The auto-correlation of $(s(t)$ in no noise is a triangular function. Using similar triangles, it can be shown that $\delta/T_c = 1 - Z(KT_c)/(A||s||^2)$. Hence, an estimate of $\delta$ in the noisy case is $\hat{\delta}/T_c = 1 - Z(\hat{k}T_c)/(A||s||^2)$.

## Problem 8.19

### Part a

Let the output of the shift register at $(n-1)T$ ($T$ =clock period) be $[s_{n-1}(0), s_{n-1}(1), s_{n-2}(2)]$. Now define the input of the register as $s_{n-1}(1) + s_{n-1}(2), s_{n-1}(0), s_{n-2}(1)$. The addition can be implemented as a binary adder. By setting the initial input to one of the states (given in part (b)), the PN sequence can be generated.

### Part b

The 3 bit vector takes 4 distinct values given as $(1,0,1),(1,1,0),(1,1,1),(0,1,1)$.

### Part c

Binary sequence obtained is 1011. It is periodic.

### Part d

Auto-correlation function is $4\delta[n]$ for n=$\{0,\cdots,3\}$, the period of the sequence. This is a good choice for DS waveform due to narrow auto-correlation peak but may not be practically very useful due to its small length.

## Problem 8.20

### Part a

From section 2.5.3, the time domain expression of Raised Cosine pulse for $\alpha = 0.5$ can be written as

$$r_\psi = \text{sinc}(t/T) \cdot \frac{cos(\pi t/2T)}{1-(t/T)^2}$$

### Part b

From example 8.4.1 with $D$ an integer and $\delta \in [0,1)$,

$$y_s(t) = \sum_l s[l] r_\psi(t - (l+D)T_c - \delta)$$

Hence, $y_s(mT_c) = (s \star g)[m]$ with $g[m] = r_\psi((m - D - \delta)T_c)$.

### Part c

Let $\tau = (D + \delta) \cdot T_c$. Consider the chip rate channel as $g[m] = r_\psi((m+\delta)T_c)$. Consider discrete time auto-correlation function $C_{u,v}[m] = \sum_l u[l]v^*[l+m]$. The convolution between $C_{u,v}[m]$ and $g[m]$ with $m = D$ can be simplified as

$$\begin{aligned} C_{u,v}[m] \star g[m]|_{m=D} &= \sum_l \sum_k u[k]v^*[k+l]g[D-l] \\ &= \sum_k \sum_{l_1} u[k]v^*[l_1]g[D-l_1+k] \\ &= \sum_k \sum_{l_1} u[k]v^*[l_1]r_\psi((D+\delta-l_1+k)T_c) \\ &= R_{u,v}(\tau) \end{aligned}$$

## Problem 8.21

### Part a

The signal $u$ is given as

$$u(t) = \sum_l c(l)\psi(t - lT_c)$$

with $c(l) = \tilde{b}(l)s(l)$. Then, $\mathbb{E}[c(l)] = 0$ and $\mathbb{E}[c^*(l)c(m)] = \delta(l-m)$. Thus, $c(l)$ is WSS with correlation function $R_c(k) = \delta(k)$. Using the result from problem 2.22, the PSD is given as

$$S_u(f) = \frac{|\mathbf{\Psi}(f)|^2}{T_c}$$

### Part b

$$\psi * (-t) \leftrightarrow \mathbf{\Psi} * (f)$$

$$S_z(f) = S_u(f)|\mathbf{\Psi}^*(f)|^2 = \frac{|\mathbf{\Psi}(f)|^4}{T_c}$$

### Part c

$$\begin{aligned} S[l] &= A_1 u_1(t) \star \psi^*(-t)|_{lT_c} \\ &= A_1 \int u_1(\tau)\psi^*(\tau - lT_c)d\tau \\ &= A_1 \sum_{l_1} \tilde{b}(l_1)s(l_1) \int \psi(\tau - l_1T_c)\psi^*(\tau - lT_c)d\tau \end{aligned}$$

Assuming the chip matched filter is chosen so that the correlation function $R(kT_c) = E_c\delta(k)$, we obtain $S[l] = A_1\tilde{b}(l)s(l)E_c$. For rectangular chip waveform, $E_c = T_c$. Hence, $S[l] = A_1\tilde{b}(l)s(l)T_c$.

### Part d

Assuming the noise and interferers are all independent random processes,

$$\begin{aligned} S_I(f) &= S_N(f) + \sum_{k=2}^{K} S_k(f) \\ &= N_o|\mathbf{\Psi}(f)|^2 + |A_k|^2 \frac{|\mathbf{\Psi}(f)|^4}{T_c} \end{aligned}$$

The result in (b) is used to write the PSD of the interferers.

### Part e

$$\begin{aligned} \mathbb{E}[|I[l]|^2] &= \int S_I(f)df \\ &= N_o \int |\mathbf{\Psi}(f)|^2 df + |A_k|^2 \frac{\int |\mathbf{\Psi}(f)|^4 df}{T_c} \end{aligned}$$

Using $\int |\Psi(f)|^2 df = E_c$, we get the result.

$$\int |\mathbf{\Psi}(f)|^4 df = \int (|\mathbf{\Psi}(f)|^2)^2 df = \int |R(t)|^2 dt$$

where $R(t)$ is the inverse Fourier transform of $|\Psi(f)|^2$. It is easy to obtain

$$T_c \mathrm{sinc}(T_c f) e^{-j\pi f T_c} \leftrightarrow I_{[0,T_c]}(t)$$

Hence, $|\Psi(f)|^2 = T_c^2 \mathrm{sinc}^2(T_c f)$. Also,

$$T_c^2 \mathrm{sinc}^2(T_c f) \leftrightarrow I_{[-T_c/2, T_c/2]}(t) \star I_{[-T_c/2, T_c/2]}(t)$$

Thus, $R(t)$ is given as

$$\mathrm{R}(t) = T_c \left(1 - \frac{|t|}{T_c}\right), \qquad |t| < T_c$$

$R(t)$ is 0 elsewhere. Hence,

$$\int |\mathbf{\Psi}(f)|^4 df = \int_{-T_c}^{T_c} T_c^2 \left(1 - \frac{|t|}{T_c}\right)^2 dt = \frac{2T_c^2}{3}$$

### Part f

$$\sum_{l=0}^{N-1} s_1^*[l] S[l] = bA_1 E_c \sum_{l=0}^{N-1} |s_1(l)|^2$$

Using Central limit theorem and since $\mathbb{E}[s(l)^2] = 1$,

$$\sum_{l=0}^{N-1} |s_1(l)|^2 = N$$

This yields the result.

### Part g

Signal Power is $\mathbb{E}[b^2]N^2E_c^2 = N^2E_c^2$. The total interference plus noise power is given as

$$\sum_{l_1}\sum_{l_2}\mathbb{E}[s_1^*[l]s_1[m]]\mathbb{E}[I*[l]I[m]] = \mathbb{E}[|I[l]|^2]$$

The fact that $s_1[l]$ are i.i.d, zero mean has been used to arrive at the above result. SIR can be obtained using the result in part e.

### Part h

For QPSK, $E_s = 2E_b$. For rectangular waveform, $\int|\mathbf{\Psi}(f)|^4df = 2T_c^2/3$ and $E_c = T_c$. The result follows.

### Part i

Asynchronous reception results in lower interference than the synchronous reception.

## Problem 8.22

### Part a

The MMSE correlator $c_{\text{MMSE}}$ is given as

$$\begin{aligned} R\mathbf{c} &= \mathbf{p} \\ A_1^2\mathbf{s}_1\mathbf{s}_1^T\mathbf{c}_{\text{MMSE}} + A_2^2\mathbf{s}_2\mathbf{s}_2^T\mathbf{c}_{\text{MMSE}} + \sigma^2\mathbf{c}_{\text{MMSE}} &= s_1 \end{aligned}$$

Putting $\mathbf{c}_{\text{MMSE}} = \alpha_1 s_1 + \alpha_2 s_2$ in the above equation, we obtain,

$$\begin{aligned} A_1^2(\alpha_1 + \alpha_2\rho) + \sigma^2\alpha_1 &= 1 \\ A_2^2(\alpha_2 + \alpha_1\rho) + \sigma^2\alpha_2 &= 0 \end{aligned}$$

Solving,

$$\begin{aligned} \alpha_1 &= \frac{A_2^2 + \sigma^2}{(A_2^2 + \sigma^2)(A_1^2 + \sigma^2) - \rho^2A_1^2A_2^2} \\ \alpha_2 &= \frac{-\rho A_2^2}{(A_2^2 + \sigma^2)(A_1^2 + \sigma^2) - \rho^2A_1^2A_2^2} \end{aligned}$$

### Part b

Using the result in Problem 5.9,

$$\text{MMSE} = 1 - \mathbf{s_1}^*\mathbf{c}_{\text{MMSE}} = 1 - \mathbf{s}_1^*(\alpha_1\mathbf{s}_1 + \alpha_2\mathbf{s}_2) = 1 - \alpha_1 - \alpha_2\rho$$

### Part c

At $\sigma^2 = 0$, $\alpha_1 = 1/(A_1^2(1-\rho^2)$ and $\alpha_2 = -\rho/(A_1^2(1-\rho^2)$. The corelator output is given as

$$\mathbf{c}_{\text{MMSE}}^* \mathbf{r} = \frac{b_1}{A_1} + \mathbf{c}_{\text{MMSE}}^* \mathbf{n}$$

This suggests that as $\sigma^2 \to 0$, MMSE correlator becomes the ZF correlator.

### Part d

Dividing the numerator and denominator of $\alpha_1$ and $\alpha_2$ by $A_2^2$ and then taking the limit $A_2/A_1 \to \infty$,

$$\alpha_1 = \frac{1}{(A_1^2 + \sigma^2) - \rho^2 A_1^2}$$

$$\alpha_2 = \frac{-\rho}{(A_1^2 + \sigma^2) - \rho^2 A_1^2}$$

This correlator is similar to the one in part (c) except for a scaling factor. Hence, it is also a ZF correlator.

## Problem 8.23

### Part a

When user 1 has larger amplitude than user 2, then the estimate of $\hat{b_1}$ is corrupted only by noise and hence at high SNR, it is very reliable. This makes the estimate of $\hat{b_2}$ also limited by only SNR. Hence, the scheme given works well.

In the case when user 2 has larger amplitude, estimate $\hat{b_1}$ is wrong most of the time due to its dependence on $b_2$ and not on $b_1$. This makes the estimate $\hat{b_2}$ also wrong. Hence, the given scheme works poor.

### Part b

For user 1, $\hat{b_1}$ is just the output of Matched filter reception. The efficieny $\eta$ is given by Example 8.4.3 and is non-zero when $A_1 > A_2|\rho|$.

### Part c

$$z_2 \approx \rho A_1(b_1 - \hat{b}_1) + A_2 b_2 + n_2$$

As $A_2/A_1 \to \infty$, $z_2 \approx A_2 b_2 + n_2$. This is same as the single user case. Hence, $\eta = 1$.

## Problem 8.24

### Part a

The system model is given as $\mathbf{r} = [A_1\mathbf{s}_1, A_2\mathbf{s}_2, A_3\mathbf{s}_3](b_1, b_2, b_3)^t + n$. From Chapter 5, the ZF correlator for user 1 can be found as $U(U * U)^{-1}e_1$ with $e_1 = (1, 0, 0)^T$ and $U = [\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3]$. For user 1, the ZF correlator is given as

$$\mathbf{c}_1 = (0, 0.5, 0.5, -0.5, 0.5)$$

Similarly,

$$\mathbf{c}_2 = (0.82, -0.41, 0, 0.41, 0)$$

$$\mathbf{c}_3 = (0.82, 0, -0.41, 0, 0.41)$$

The correlators are chosen so that $||\mathbf{c}_i|| = 1$. The asymptotic efficiency is then given by $\eta_i = |\mathbf{c}_i * \mathbf{s}_i|^2/||\mathbf{s}||^2$ can be computed for the three users as $\eta_1 = 0.8, \eta_2 = 0.53, \eta_3 = 0.53$.

### Part b

The matched filter statistic for user 1 is given as

$$z_1 = A_1||\mathbf{s}_1||^2 b_1 + A_2 < \mathbf{s}_1, \mathbf{s}_2 > b_2 + A_3 < \mathbf{s}_1, \mathbf{s}_3 > b_3 + < \mathbf{s}_1, \mathbf{n} >$$

Normalizing to $||\mathbf{s}_1||^2$,

$$z_1 = A_1 b_1 + 0.2A_2 b_2 + 0.2A_3 b_3 + n_1$$

with $n_1 \sim N(0, \sigma^2)$. Due to symmetry,

$$\begin{aligned} P_{e|(b_2,b_3)} &= Pr(z_1 > 0|b_1 = -1) \\ &= Q\left(\frac{A_1 - 0.2A_2 b_2 - 0.2A_3 b_3}{\sigma}\right) \end{aligned}$$

The error probability can be written as

$$P_e = \frac{P_{e|(1,1)} + P_{e|(-1,-1)} + P_{e|(1,-1)} + P_{e|(-1,1)}}{4}$$

Given, $A_i^2/\sigma^2 = 2E_b/N_o = 11.25$. This implies $P_e = 2.2e - 7$.
For ZF reception, effective SNR is given as $A_1^2\eta_1/\sigma^2$. This implies $P_e = Q(\sqrt{SNR}) = 5.7e - 13$.

### Part c

Given,

$$\frac{A_1^2}{\sigma^2} = 11.25, \frac{A_2^2}{\sigma^2} = 35.56, \frac{A_3^2}{\sigma^2} = 35.56$$

For these values, $P_e = 0.25$.

### Part d

User 2 is equivalent to 2 independent interference vectors,

$$\mathbf{s}_{2,1} = (1, -1, 0, 0, 0)^T$$

$$\mathbf{s}_{2,2} = (0, 0, 1, -1, 1)^T$$

The effective system model is similar to part (a) with $U = [A_1\mathbf{s}_1, A_{2,1}\mathbf{s}_{2,1}, A_{2,2}\mathbf{s}_{2,2}, A_3\mathbf{s}_3]$. The ZF correlator for user 1 is given as

$$\mathbf{c}_1 = (0.4, 0.4, 0.4, -0.27, -0.67)$$

$$\eta_1 = 0.91$$

## Problem 8.25

### Part a

Since $\mathbf{N} \sim N(0, \sigma^2\mathbf{I})$, it implies $\mathbf{r}|\mathbf{b} \sim N(\mathbf{SAb}, \sigma^2 I)$. Hence,

$$p(\mathbf{r}|\mathbf{b}) = \frac{1}{(2\pi\sigma^2)^{n/2}} e^{-\frac{|\mathbf{r}-\mathbf{SAb}|^2}{\sigma^2}}$$

This implies

$$\log(p(\mathbf{r}|\mathbf{b})) = -\frac{n}{2}\log(2\pi\sigma) - \frac{1}{\sigma^2}[||\mathbf{r}||^2 + ||\mathbf{SAb}||^2 - 2(\mathbf{SAb})^T\mathbf{r}]$$

Sufficient statistic is $(\mathbf{SAb})^T\mathbf{r}$. An equivalent sufficient statistic is $\mathbf{S}^T\mathbf{r}$.

### Part b

ML rule is given as

$$\text{argmax}_{\mathbf{b}} \log(p(\mathbf{r}|\mathbf{b})) = \text{argmax}_{\mathbf{b}} (\mathbf{SAb})^T\mathbf{r} - \frac{||\mathbf{SAb}||^2}{2}$$

Hence, $\mathbf{u} = \mathbf{SAb}$ and $\mathbf{R} = \mathbf{A^T S^T S A}$.

### Part c

## Problem 8.26

### Part a

Given, $P_F \leq 10^{-3}$. This implies

$$\sum_{h=n-k+1}^{n} \binom{n}{h} p_{\text{hit}}^h (1 - p_{\text{hit}})^{n-h} \leq 10^{-3}$$

The maximum value of $k$ for which the above inequality is satisfied has to be found. $p_{\text{hit}}$ can be found as

$$p_{\text{hit}} = 1 - \left(\frac{q-1}{q}\right)^{K-1} = 0.13$$

Using $n = 31$, the maximum allowed $k$ is 21.

### Part b

Given

$$\sum_{h=n-k+1}^{n} \binom{n}{h} p_{\text{hit}}^{h}(1-p_{\text{hit}})^{n-h} = 10^{-3}$$

Also, $n = 31$ and $k/n \geq 0.8$ or $k \geq 25$. Minimizing $q$ is equivalent to maximizing $p_{\text{hit}}$. Also, using MATLAB, $p_{\text{hit}}$ is maximum for $k = 25$. This can also be infered from the above equation analytically assuming small $p_{\text{hit}}$. Using MATLAB,

$$(p_{\text{hit}})_{max} = 0.053, q_{min} = 166$$

## Problem 8.27

### Part a

$$\begin{aligned} a[k] &= \frac{1}{P}\int_0^{P/2} e^{-j2\pi kt/P}dt - \frac{1}{P}\int_{-P/2}^{0} e^{-j2\pi kt/P}dt \\ &= \frac{1 - e^{-j\pi k} - e^{j\pi k} + 1}{2j\pi k} \\ &= \begin{cases} \frac{2}{j\pi k}, & k \text{ odd} \\ 0, & k \text{ even} \end{cases} \end{aligned}$$

### Part b

$$\begin{aligned} u(t-\tau) &= \sum_k a[k]e^{j2\pi kf_o(t-\tau)} \\ &= \sum_k a[k]e^{-j2\pi kf_o\tau}e^{j2\pi kf_ot} \end{aligned}$$

This implies $u(t-\tau) \leftrightarrow a[k]e^{-j2\pi kf_o\tau}$.

$$\frac{d}{dt}u(t) = \sum_k a[k](j2\pi kf_o)e^{j2\pi kf_ot}$$

This implies $\frac{d}{dt}u(t) \leftrightarrow (j2\pi kf_o)a[k]$.

### Part c

$$a[k] = \frac{1}{P}\int_{-P/2}^{P/2}\sum_{k_1}\delta(t-k_1P)e^{-j2\pi k f_o t}dt = \frac{1}{P}$$

### Part d

For part (a), the derivative cna be found as

$$\frac{d}{dt}u(t) = 2\sum_{k_1}\delta(t-k_1P) - 2\sum_{k_1}\delta(t-k_1P-\frac{P}{2})$$

This implies

$$\frac{d}{dt}u(t) \leftrightarrow \frac{2}{P} - \frac{2}{P}e^{-j\pi k}$$

Using (b),

$$u(t) \leftrightarrow \frac{1}{j2\pi k}\left(\frac{2}{P} - \frac{2}{P}e^{-j\pi k}\right)$$

Thus,

$$u(t) \leftrightarrow a[k] = \begin{cases} \frac{2}{j\pi k}, & k \text{ odd} \\ 0, & k \text{ even} \end{cases}$$

This is same as that obtained in part (a).

## Problem 8.28

### Part a

Plots are shown in figures 14 and 15.

### Part b

$$\begin{aligned} \theta(t) &= \sum_{n=0}^{\infty} a[n]\phi(t-nT) \\ &= \phi(t) - \phi(t-T) - \phi(t-2T) - \phi(t-3T) + \phi(t-4T) + \phi(t-5T) \end{aligned}$$

The plot is given in Fig. 16.

### Part c

$$\begin{aligned} \theta(kT) &= \sum_{n=0}^{\infty} a[n]\phi((k-n)T) \\ &= a(k-1)\frac{\pi}{12} + a(k-2)\frac{\pi}{6} + \frac{\pi}{4}\sum_{i=3}^{k} a(k-i) \end{aligned}$$

Since, $a(k-i)$ can take values $\pm 1$, $\theta(kT)$ can take values of the form $(n\pi/4 \pm \pi/12 \pm \pi/6)$. This corresponds to 16 values of $\theta(kT)$ in the range $(\pi, \pi]$.

**Part d**

The system model is given as

$$y(t) = s_{\mathbf{a}}(t) + n(t) \text{ with } s_{\mathbf{a}}(t) = Ae^{j\sum_m a(m)\phi(t-mT)}$$

From Chapter 2, MLSE estimate has sufficient statistics

$$< y, s_{\mathbf{a}} >= A\int y(t)e^{j\sum_m a(m)\phi(t-mT)}dt$$

In $[nT, (n+1)T]$,

$$\begin{aligned}\theta(t) &= \sum_m a(m)\phi(t-mT) \\ &= a[n]\phi(t-nT) + a[n-1]\phi(t-(n-1)T) + a[n-2]\phi(t-(n-2)T) + \frac{\pi}{4}\sum_{i=3}^{n} a(n-i)\end{aligned}$$

Using the substitution $\tilde{t} = t - nT$,

$$\theta(t) = a[n]\phi(\tilde{t}) + a[n-1]\phi(\tilde{t}+T) + a[n-2]\phi(\tilde{t}+2T) + \frac{\pi}{4}\sum_{i=3}^{n} a(n-i)$$

Since, $\tilde{t} \in [0, T]$,

$$\theta(t) = \frac{\pi t}{12}(a[n] + a[n-1] + a[n-2]) + \frac{\pi}{4}\left(1 + \sum_{i=3}^{n} a(n-i)\right)$$

Now, the sufficient statistics can be written as

$$< y, s_{\mathbf{a}} >= f(\mathbf{a})\int y(t)e^{-\frac{j\pi}{12}(a[n]+a[n-1]+a[n-2])t}dt$$

with $f(\mathbf{a})$ as some function of $\mathbf{a}$. Thus, an equivalent sufficient statistic is $Z_n(\alpha)$ with $\alpha = \pi(a[n] + a[n-1] + a[n-2])/12$. Since $a(n-i) = \pm 1$, $\alpha$ takes the values

$$\left\{-\frac{\pi}{4}, -\frac{\pi}{12}, \frac{\pi}{12}, \frac{\pi}{4}\right\}$$

**Part e**

It can be written

$$\theta(kT) - \theta((k-1)T) = \frac{\pi}{12}(a[n] + a[n-1] + a[n-2])$$

Since $a(n-i) = \pm 1$, $(a[n]+a[n-1]+a[n-2])$ can take values $\{\pm 1, \pm 3\}$. Hence, given $\theta((k-1)T)$, $\theta(kT)$ can take 4 possible values. Hence, the minimum number of states for MLSE reception is 4.

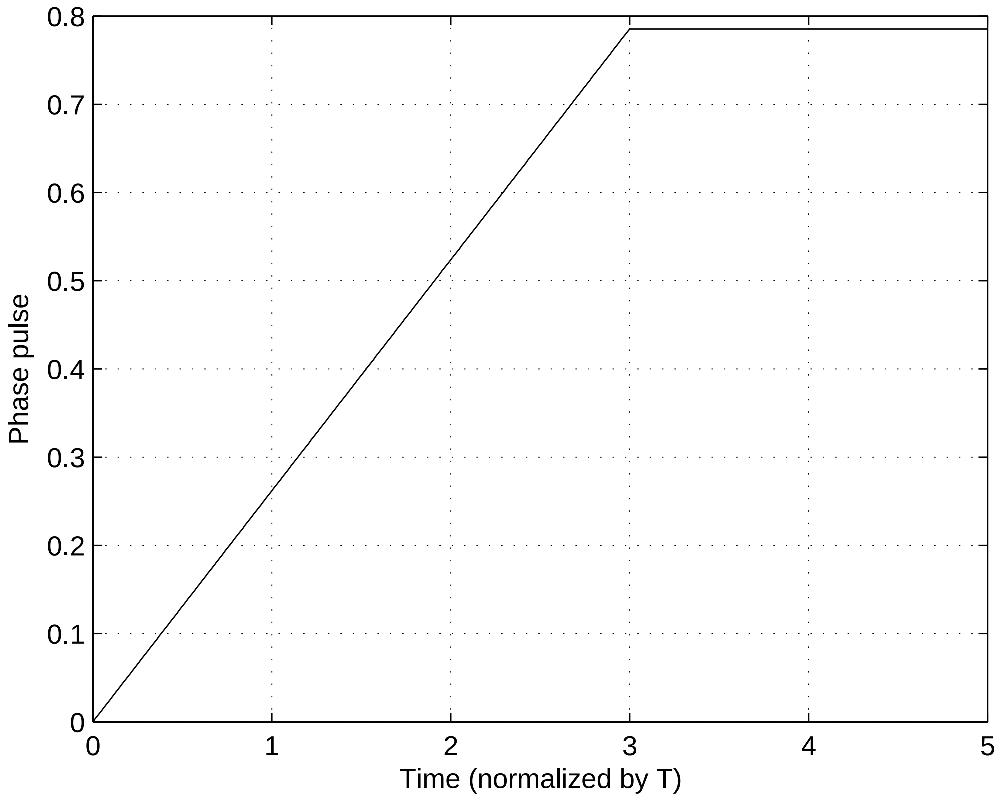



Figure 14: Figure for Problem 8.28 (a)

# Problem 8.29

## Part a

It can be easily shown that

$$\sum_{k=0}^{N-1} u(kT_s)\delta(t-kT_s) \leftrightarrow \frac{1}{T_s}\sum_{m=-\infty}^{\infty} U\left(f-\frac{m}{T_s}\right)$$

Assuming that $U(f)$ is bandlimited in $(-\frac{1}{2T_s}, \frac{1}{2T_s})$, R.H.S $\approx U(f)/T_s$ for all $f \in (-\frac{1}{2T_s}, \frac{1}{2T_s})$. This implies,

$$\int \sum_{k=0}^{N-1} u(kT_s)\delta(t-kT_s)e^{-j2\pi ft}dt \approx \frac{U(f)}{T_s}$$

Putting $f = \frac{k}{NT_s}$,

$$\sum_{k=0}^{N-1} u[k]e^{-j2\pi\left(\frac{k}{NT_s}\right)mT_s} \approx \frac{U\left(\frac{k}{NT_s}\right)}{T_s}$$

This further implies

$$T_s U[k] \approx U\left(\frac{k}{NT_s}\right)$$

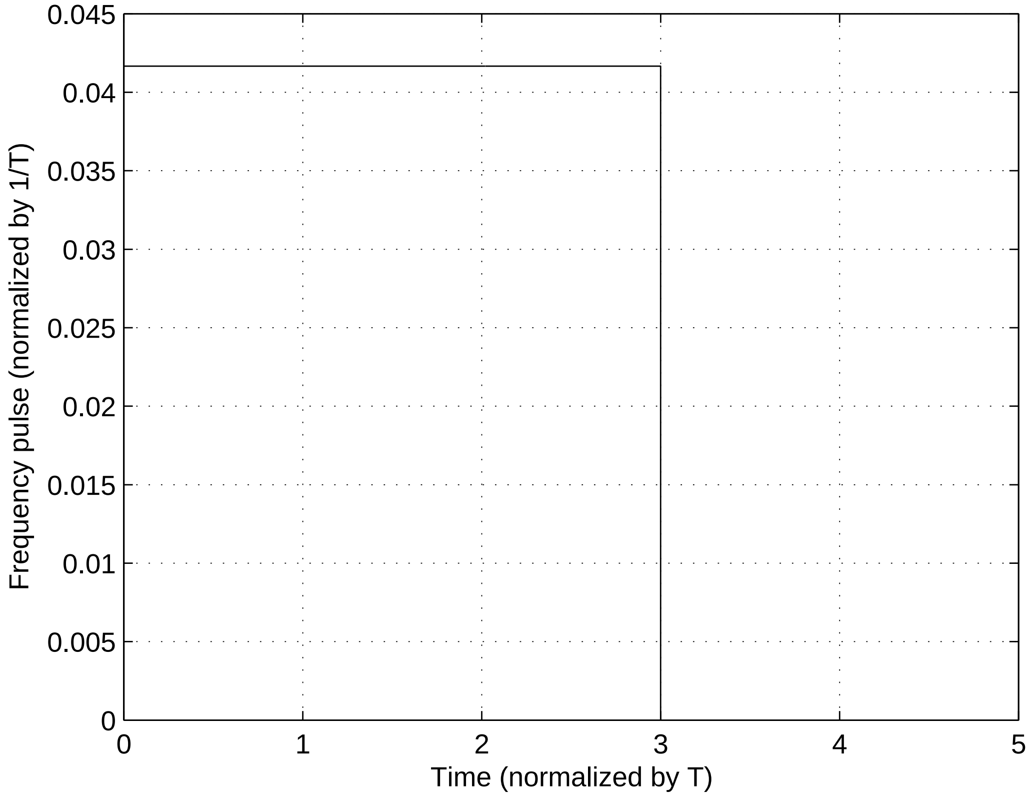



Figure 15: Figure for Problem 8.28 (a)

**Part b**

Given, $T_s = T/8$ and frequency resolution $1/(NT_s) = 1/(10T)$. This implies

$$N = 10\frac{T}{T_s} = 80 \Rightarrow \text{ Take } N = 128$$

$$T_o = NT_s = 16T$$

## Problem 8.30

The plots are shown in figures 17 and 18. Since $s_o(t)$ extends between $(0, 4T)$, each symbol is spread over 3 symbol periods. For the dominant linear system,

$$B_o[k] = e^{j\pi h \sum_{n=0}^{k} a[n]}$$

The summation $\sum_{n=0}^{k} a[n]$ can take any integral value for some $k$. So, $B_o[k]$ takes values $e^{jm\pi h}$ for $m \in Z$. This is equivalent to the set $\{1, -1, j, -j\}$.

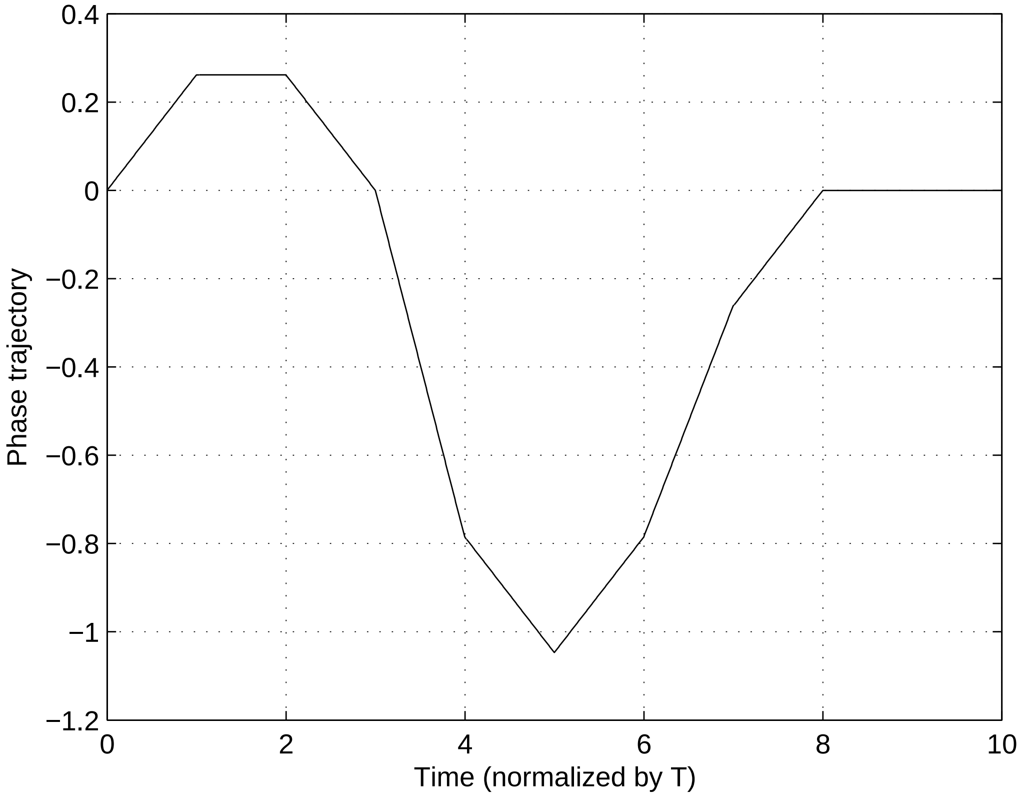



Figure 16: Figure for Problem 8.28 (b)

## Problem 8.31

### Part a

Let $\mathbf{Y} = \mathbf{Z} - \mathbf{m}$. This implies $\mathbf{Y} \sim CN(0, \mathbf{C})$.

$$\begin{aligned} h(\mathbf{Y}) &= -\int p_Y(\mathbf{y}) \log(p_Y(\mathbf{y})) d\mathbf{y} \\ &= -\int p(\mathbf{y} + \mathbf{m}) \log(p(\mathbf{y} + \mathbf{m})) d\mathbf{y} \\ &= -\int p(\mathbf{z}') \log(p(\mathbf{z}')) d\mathbf{z}' \\ &= h(\mathbf{Z}) \end{aligned}$$

### Part b

From Chapter 4,

$$p(\mathbf{z}) = \frac{1}{\pi^n det(\mathbf{C})} e^{-(\mathbf{z}-\mathbf{m})^* C^{-1} (\mathbf{z}-\mathbf{m})}$$

$$
\begin{aligned}
h(\mathbf{Z}) &= \mathbb{E}[-\log(p(\mathbf{z}))] \\
&= \log(det(\pi\mathbf{C})) + \log_2(e)\mathbb{E}[(\mathbf{z}-\mathbf{m})^* C^{-1}(\mathbf{z}-\mathbf{m})]
\end{aligned}
$$

Since $\mathbf{C} = \mathbf{C}^H$, $C$ can be diagonalized and $\mathbf{C} = U\Sigma U^H$ with $\Sigma$ as diagonal matrix of eigenvalues $\{\lambda_i\}$ and $U$ an unitary matrix, that is, $UU^H = U^H U = \mathbf{I}$. Consider $\mathbf{y} = U\mathbf{z}$. Then,

$$
\begin{aligned}
\mathbb{E}[(\mathbf{z}-\mathbf{m})^* C^{-1}(\mathbf{z}-\mathbf{m})] &= \mathbb{E}[(\mathbf{z}-\mathbf{m})^* U^{-H}\Sigma^{-1}U^{-1}(\mathbf{z}-\mathbf{m})] \\
&= \mathbb{E}[\mathbf{y}^H \Sigma^{-1}\mathbf{y} \\
&= \sum_i \frac{1}{\lambda_i}\mathbb{E}[|y_i|^2]
\end{aligned}
$$

Using $\mathbb{E}[\mathbf{y}\mathbf{y}^H] = U\mathbf{C}U^H = \Sigma$, it can be infered $\mathbb{E}[|y_i|^2] = \lambda_i$. Hence, L.H.S $= \sum 1 = n$. Thus, $h(\mathbf{Z}) = \log det(\pi e\mathbf{C})$.

Using the fact that $det(\mathbf{C}) = \prod \lambda_i$, the other result can be obtained.

### Part c

Since, $\mathbf{Z}$ is complex gaussian, the components of $\mathbf{Z}$ are complex gaussian. Hence, $Y_i$ is also complex gaussian. Assuming $\mathbf{m} = \mathbf{0}$, $\mathbb{E}[Y_i] = 0$. Also,

$$
\mathbb{E}[Y_i Y_j^*] = \mathbb{E}[\nu_i^* z z^* \nu_j] = \nu_i^* \mathbf{C}\nu_j = \lambda_j \nu_i^* \nu_j
$$

Hence, $\mathbb{E}[|Y_i|^2] = \lambda_i$ and for $i \neq j$, $\mathbb{E}[Y_i Y_j^*] = 0$ as $\nu_j$ and $\nu_i$ are orthonormal. Thus, $\{Y_i\}$ are uncorrelated and hence independent.

Since $\mathbf{C} = \mathbf{C}^*$, any vector can be written as linear combination of eigenvectors of $\mathbf{C}$. Hence, $\mathbf{Z} = \sum Y_i \nu_i$. Here, the components of $\mathbf{Z}$ are given by $\{Y_i\}$. Using the fact that $\{Y_i\}$ are independent,

$$
h(\mathbf{Z}) = h(Y_1, \cdots, Y_n) = \sum h(Y_i) = \sum \log(\pi e \lambda_i)
$$

## Problem 8.32

### Part a

$$
C_a = \frac{1}{N_T}\sum_{l-1}^{N_T} \log(1 + |h_l|^2 \mathrm{SNR})
$$

Consider a random variable taking the values $\{|h_l|^2\mathrm{SNR}\}$ each with probability $1/N_T$. By Jensen's inequality,

$$
\mathbb{E}[\log(1+X)] \leq \log(1+\mathbb{E}[X]) = \log\left(1 + \frac{||\mathbf{h}||^2}{N_T}\mathrm{SNR}\right)
$$

### Part b

Since, $\{h_l\}$ are i.i.d, $\{\log(1+|h_l|^2\mathrm{SNR})\}$ are also i.i.d and hence the antenna-hopping capacity $C$ can be approximated as a Gaussian random variable. Also, as $h_l \sim CN(0,1)$, $|h_l|^2$ is exponential

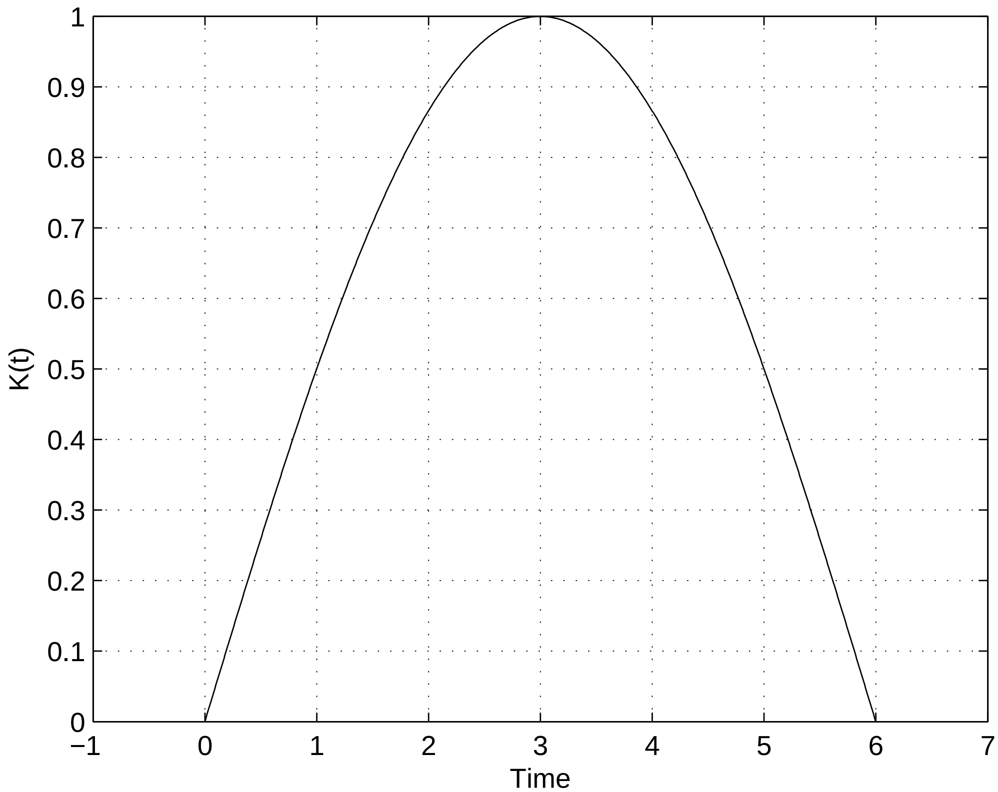



Figure 17: Figure for Problem 8.30

with mean 1. The mean of $C$ can be calculated as

$$\begin{aligned} \mu = \mathbb{E}[C] &= \mathbb{E}[\log(1 + |h_l|^2 \mathrm{SNR})] \\ &= \int_0^\infty \log(1 + r\mathrm{SNR}) e^{-r} dr \qquad r = |h_l|^2 \end{aligned}$$

For SNR=10, $\mathbb{E}[C] \approx 2.91$. The variance can similarly be calculated as

$$\begin{aligned} \sigma^2 = \mathrm{Var}[C] &= \frac{1}{N_T} \mathrm{Var}[\log(1 + |h_l|^2 \mathrm{SNR})] \\ &= \frac{1}{N_T} \left( \int_0^\infty (\log(1 + r\mathrm{SNR}))^2 e^{-r} dr - (\mathbb{E}[C])^2 \right) \end{aligned}$$

For SNR=10 and $N_T = 6$, $\sigma^2 = 0.29$.

**Part c**

Let $R$ be the outage capacity. Then, $\Pr(C < R) = 0.01$. This implies,

$$Q\left(\frac{R - \mu}{\sigma}\right) = 0.99$$

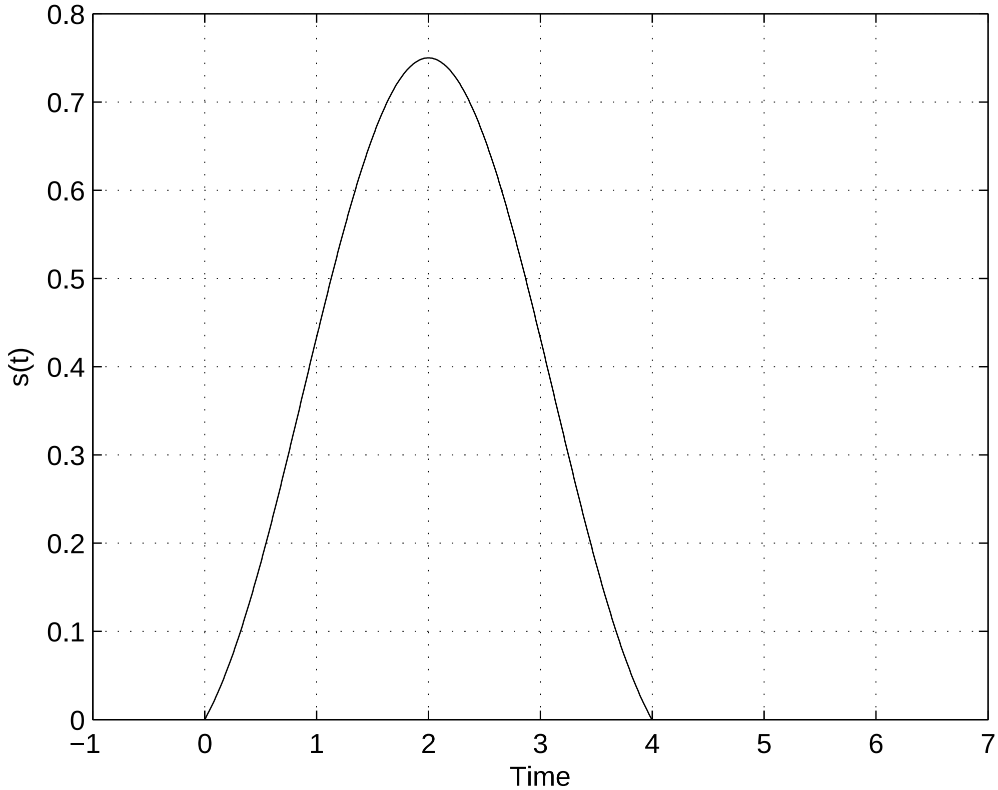



Figure 18: Figure for Problem 8.30

Hence, $R = 1.65$. The capacities for AWGN and Rayleigh channels can be found as

$$C_{\text{awgn}} = \log(1 + \text{SNR}) = 3.46$$

$$C_{\text{rayleigh}} = \int_0^\infty \log(1 + r\text{SNR})e^{-r}dr = 2.91$$

It can be seen that outage capacity is smaller than the ergodic capacity of Rayleigh channel. This is due to the fluctuations in the channel gains.

**Part d**

Use $\log(ab) = \log a + \log b$.

### Part e

The approximation, $\log_2(1+x) \approx \log_2(e)x$, can be used to derive the result as follows.

$$\begin{aligned}
\mathrm{Var}[\log_2(1+aX)] &= \mathrm{Var}\left[\log_2\left(1+\frac{a}{1+a}(X-1)\right)\right] \\
&= \mathrm{Var}\left[\log_2(e)\frac{a}{1+a}(X-1)\right] \\
&= (\log_2(e))^2\frac{a^2}{(1+a)^2}\mathrm{Var}[X]
\end{aligned}$$

### Part f

Since $X_l$ are i.i.d,

$$\mathrm{Var}[C] = \frac{1}{N_T}\mathrm{Var}[\log_2(1+aX_1)]$$

Put $X_1 = |h_1|^2$ and $a = \mathrm{SNR}$. Then, since $X_1$ is exponential with mean 1. Using the approximation in (e),

$$\begin{aligned}
&= \frac{(\log_2(e))^2}{N_T}\left(\frac{\mathrm{SNR}}{1+\mathrm{SNR}}\right)^2\mathrm{Var}[|h_1|^2] \\
&= \frac{(\log_2(e))^2}{N_T}\left(\frac{\mathrm{SNR}}{1+\mathrm{SNR}}\right)^2
\end{aligned}$$

### Part g

For SNR =10 and $N_T = 6$, $\sigma = \sqrt{\mathrm{Var}[C]}$ can be calculated using part (f) as 0.2867. This is very close to the value obtained by simulation in part (b). For 1% outage,

$$Q\left(\frac{R-\mu}{\sigma}\right) = 0.99$$

This implies $R = 1.66$. This value is very close to the one obtained in (c).

## Problem 8.33

### Part a

$|h_l|^2$ is exponential with mean 1 and all $\{|h_l|^2\}$ are i.i.d. Using problem 8.5 (e), for $Y = N_T G$,

$$p_Y(y) = \frac{y^{N_T-1}}{(N_T-1)!}e^{-y}, \qquad y \geq 0$$

The pdf of $G$ can be written as

$$p_G(y) = N_T\frac{(N_T y)^{N_T-1}}{(N_T-1)!}e^{-N_T y}, \qquad y \geq 0$$

### Part b

The outage probability is given as

$$\begin{aligned}
\Pr(C < R) &= \Pr(\log_2(1 + G \cdot \mathrm{SNR}) < R) \\
&= \Pr(G < \frac{2^R - 1}{\mathrm{SNR}}) \\
&= \int_0^{\frac{2^R-1}{\mathrm{SNR}}} \frac{(6y)^5}{5!} e^{-6y} dy
\end{aligned}$$

The above integral equals 0.01 for $R \approx 1.99$. Compared with Prob. 8.32, this result is higher. This is expected as averaged channel gain has less fluctuations and this increases the outage capacity.

### Part c

Using $G \sim N(1, 1/N_T)$, the outage probability is given as

$$\Pr(G < \frac{2^R - 1}{\mathrm{SNR}}) = 1 - Q\left[\sqrt{N_T}\left(\frac{(2^R - 1)}{\mathrm{SNR}} - 1\right)\right]$$

For outage probabilty of 0.01, $R$ can be calculated as 0.59. This is quite different from result of (b). This is due to the inaccuracy of the approximation of the distribution of $G$ near 0 by the Gaussian tail. $p_G(y)$ is zero for $y < 0$ while Gaussian has an infinite tail as shown in fig. 19.

## Problem 8.34

### Part a

The SVD of $H$ yields $H = UDV^H$. The eigenmodes are given as

$$v_1 = [-0.92, -0.26], u_1 = [-0.26 - 0.46j, -0.66j, 0.26 - 0.46j], \sqrt{\lambda_1} = 2.56$$

$$v_2 = [0.26(-1 + j), 0.66(1 - j)], u_2 = [-0.66 + 0.18j, 0.26j, 0.66 + 0.18j], \sqrt{\lambda_2} = 1.56$$

### Part b

Assume the noise power to be 1 on each of the scalar channels (eigenmodes). Let $P_1$ and $P_2$ represent the power transmitted on the 2 eigenmodes. Using the waterfilling solution to maximize the capacity,

$$P_1 = \max\left(0, a - \frac{1}{\lambda_1}\right)$$

$$P_2 = \max\left(0, a - \frac{1}{\lambda_2}\right)$$

Here, $a$ is chosen so that $P_1 + P_2 = P$. The values of $P_1$ and $P_2$ as a function of $P$ are plotted in fig. 21.

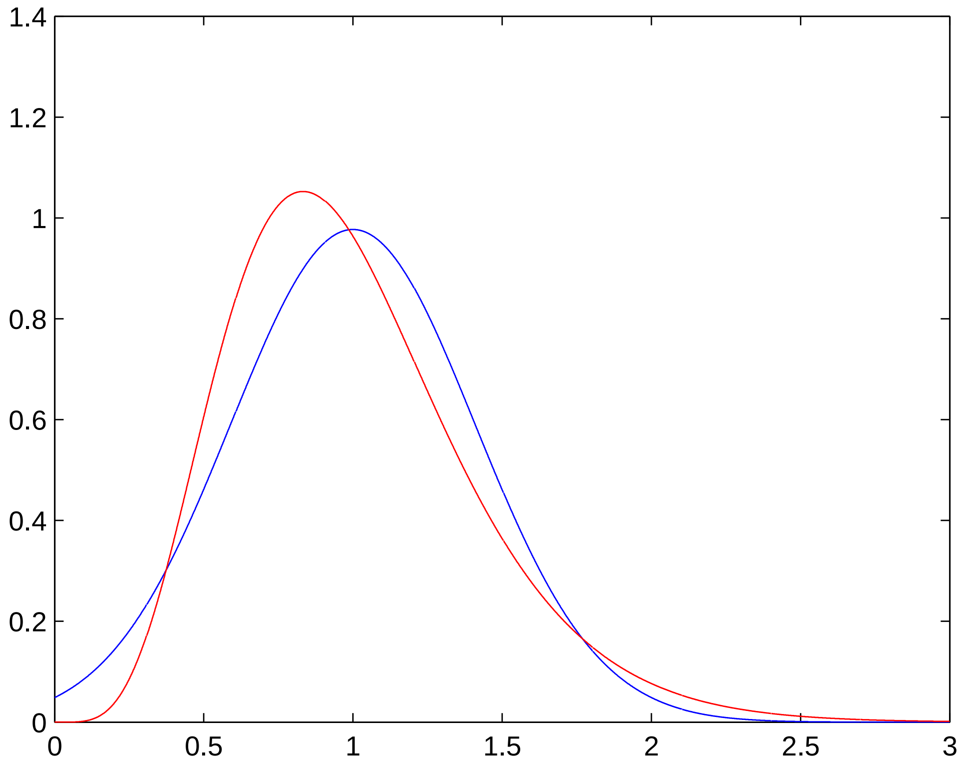



Figure 19: Figure for Problem 8.33

The capacity is given as

$$C = \log(1 + \lambda_1 P_1) + \log(1 + \lambda_2 P_2)$$

$C$ is plotted as a function of $P$ in fig. 20. It can be seen that reduction of transmit antennas reduces the capacity.

**Part c**

Part (i)
Since only 1 transmit element is used,

$$C = \log(1 + \lambda_1 P)$$

Plot is given in fig. 20.

Part (ii)
Modified $H$ is given as

$$H = \begin{pmatrix} 1+j & j \\ \sqrt{2}j & \frac{1+j}{\sqrt{2}} \end{pmatrix}$$

The eigenmodes are calculated as

$$v_1 = [-0.85, 0.53], u_1 = [-0.37 - 0.60j, -0.16 - 0.69j], \sqrt{\lambda_1} = 2.29$$

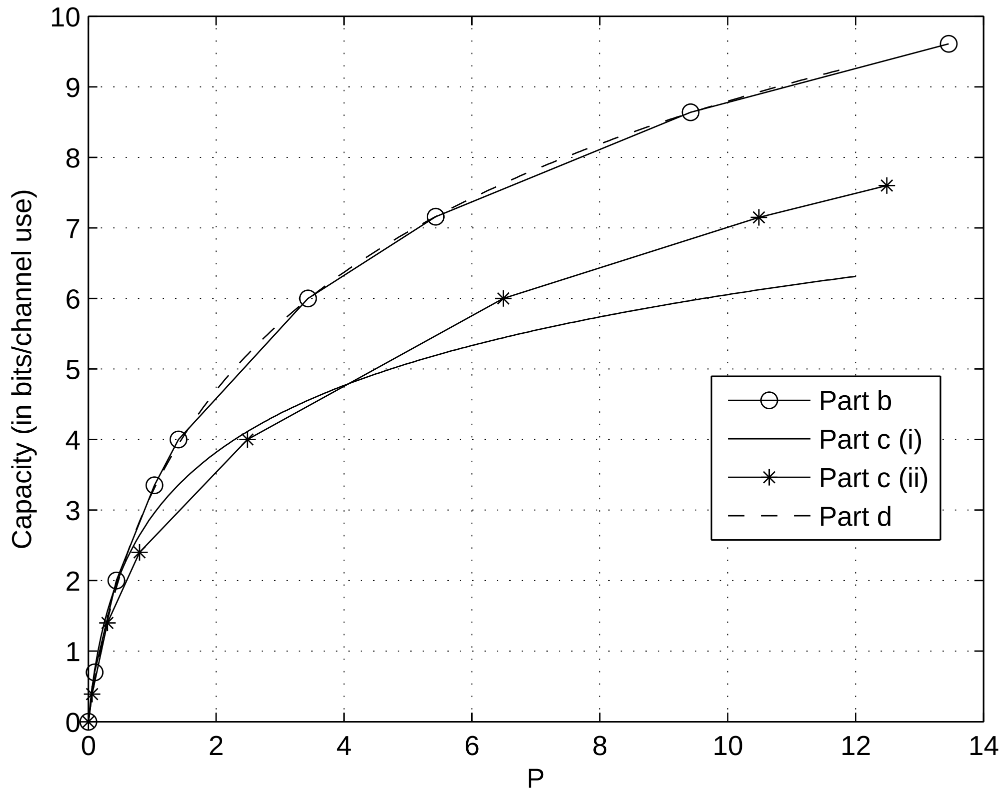



Figure 20: Figure for Problem 8.34 (b)

$$v_2 = [-0.53, -0.85], u_2 = [0.60 - 0.37j, -0.69 + 0.16j], \sqrt{\lambda_2} = 0.87$$

Values of transmit powers of both the transmitters $P_1$ and $P_2$ are plotted as functions of $P$ in fig. **??**. $C$ is plotted as a function of $P$ in fig. 20. It can be seen that reduction of receive antennas also reduces the capacity.

**Part d**

Equal powers on both transmit antennas implies equal powers on both the eigenmodes. This implies,

$$C = \log(1 + \lambda_1 P/2) + \log(1 + \lambda_2 P/2)$$

From fig. 20, it can be seen that the capacity is close to the optimal waterfilling solution. The knowledge about the channel is not of much importance in this situation where both eigenmodes have comparable powers.

## Problem 8.35

The scalar channel gains can be found using SVD of $H$ as $\sqrt{\lambda_1} = 2.29, \sqrt{\lambda_1} = 0.87$. Using the waterfilling solution as in Prob. 8.34 for $P_1 + P_2 = 1$ , $C$ is maximized when $P_1 = 1$ and $P_2 = 0$.

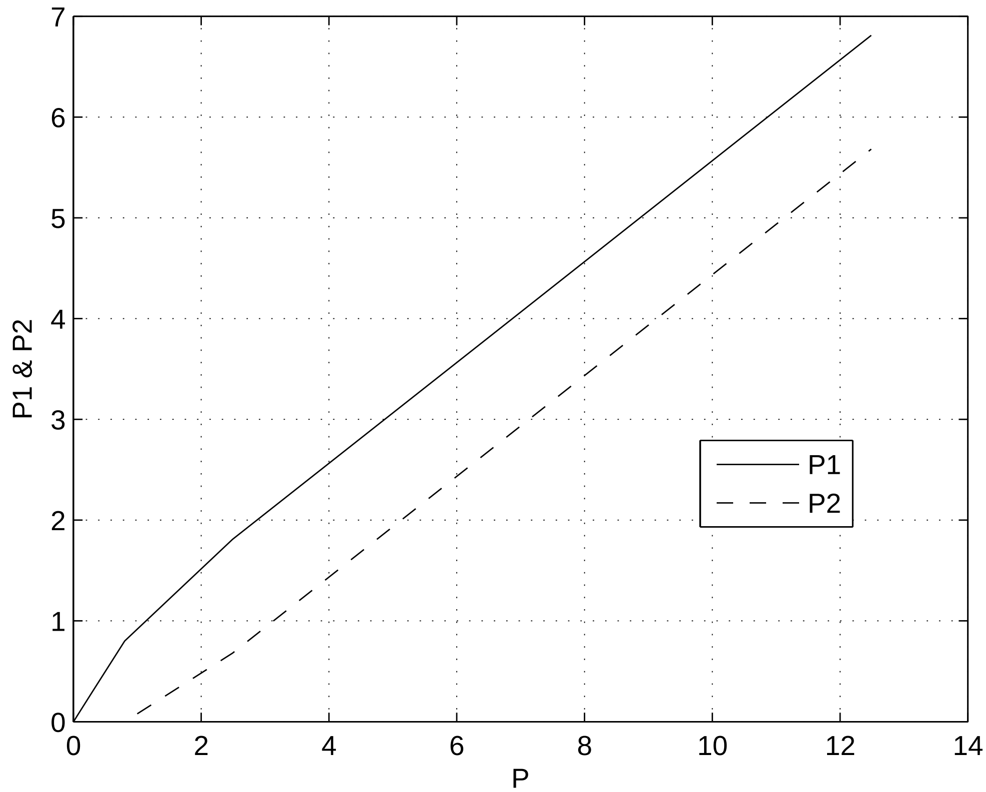



Figure 21: Figure for Problem 8.34 (c)

The achievable capacity is 2.65 bits/channel use. The transmit strategy is to transmit only along the dominant eigenmode. The corresponding eigenvector is given as $(0.85, 0.53)$. The symbols are scaled by 0.85 and 0.53 before sending them on the transmit antennas 1 and 2 respectively.

## Problem 8.36

### Part a

Several realizations of the random matrix $H$ are generated for both 2x2 and 3x3 systems. Ergodic capacity is found using 8.129. Outage capacity $R$ is found as that value of $R$ which satisfies

$$\Pr\left(\sum_i \log\left(1+\frac{\text{SNR}}{N_T}\lambda_i\right) < R\right) = 0.1$$

$\lambda_i$ are obtained using the SVD of the random matrix $H$ and this probability is computed empirically by counting number of instances when the capacity is less than $R$ and dividing it by the total number of instances. The plot of ergodic and outage capacities is shown in fig. 22. It can be seen that 3x3 system achieves better ergodic as well as outage capacities.

### Part b

The plot is shown in fig. 23. It can be seen that $C(n)/n$ is almost independent of $n$ and depends only on the total transmit power $P$. This suggests that $C$ scales as $n$ for all transmit powers considered.

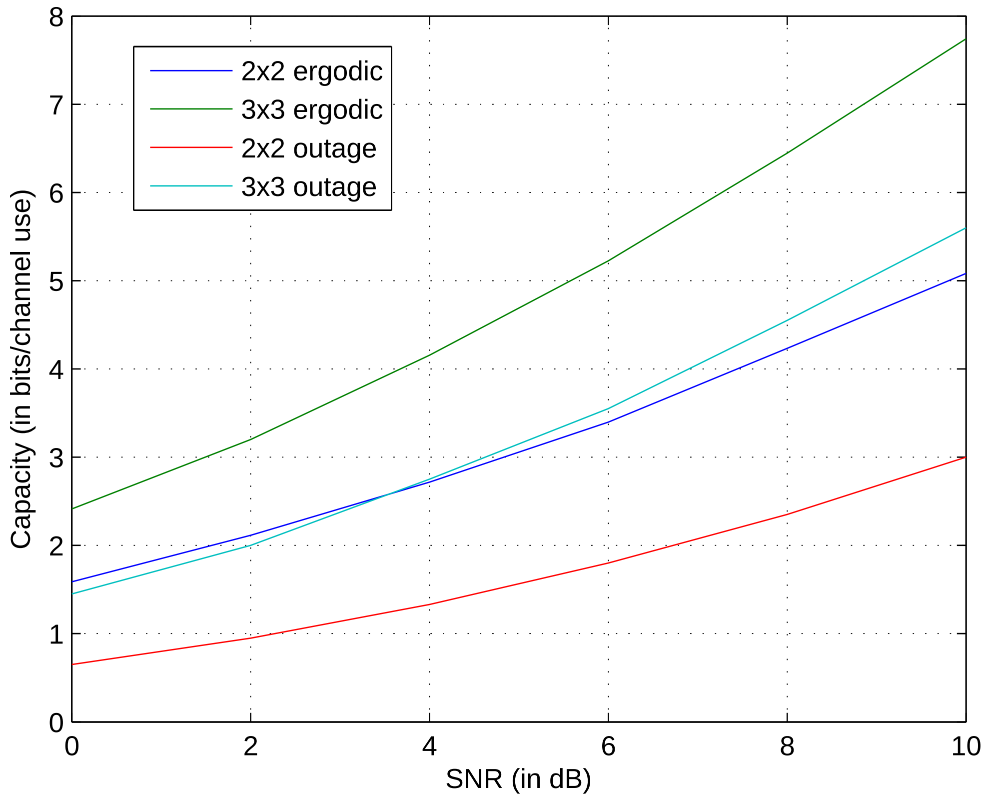



Figure 22: Figure for Problem 8.36 (a)

## Problem 8.37

### Part a

Phase difference between the transmit and receive antennas is given as

$$\phi = \frac{2\pi d \sin\theta}{\lambda} = \pi \sin\theta$$

$C$ is computed using 8.119 as

$$C = \frac{6}{\pi} \int_{-\pi/12}^{\pi/12} \mathbf{a}(\theta)\mathbf{a}^H(\theta) d\theta$$

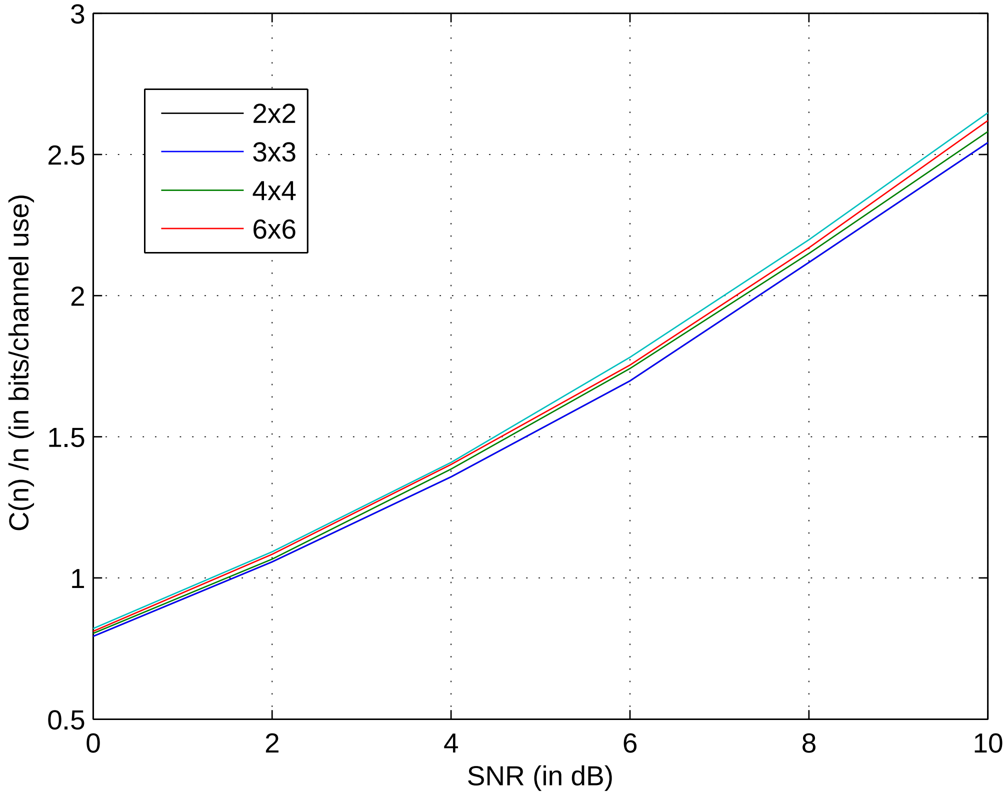



Figure 23: Figure for Problem 8.36 (b)

with $\mathbf{a}(\theta)$ as in 8.117. $C$ is given as

$$C = \begin{pmatrix} 1 & 0.89 & 0.61 & 0.26 & -0.04 & -0.20 \\ 0.89 & 1 & 0.89 & 0.61 & 0.26 & -0.04 \\ 0.61 & 0.89 & 1 & 0.89 & 0.61 & 0.26 \\ 0.26 & 0.61 & 0.89 & 1 & 0.89 & 0.61 \\ 0.61 & 0.26 & 0.61 & 0.89 & 1 & 0.89 \\ 0.89 & 0.61 & 0.26 & 0.61 & 0.89 & 1 \end{pmatrix}$$

The eigenvalues of $C$ are found as $\{3.63, 2.04, 0.32, 0.01, 0.0002, 0\}$. The dominant eigenvector is given as $\mathbf{v}_1 = \{0.28, 0.42, 0.50, 0.50, 0.42, 0.28\}$. This is close to the $\mathbf{a}(0)$ which has all its entries as $1/\sqrt{6}$. So, the transmitter can send most of its power along $\theta = 0$.

**Part b**

The channel can be written using 8.120 as $\mathbf{h} = \sum_i \sqrt{\lambda_i} h_i \mathbf{v}_i$, with $h_i$ as $CN(0,1)$. Combining with the transmit power gain $P_i$, the effective channel is $\mathbf{h} = \sum_i \sqrt{P_i \lambda_i} h_i \mathbf{v}_i$. The ergodic capacity is given as

$$C \;=\; \mathbb{E}[\log(1 + ||\mathbf{h}||^2)]$$

Using the orthonormality of $\mathbf{v}_i$ and the fact that $\{|h_i|^2\}$ are exponential with mean 1 gives the result.