STANFORD

Lecture 2

The AWGN

January 11, 2024

JOHN M. CIOFFI

Hitachi Professor Emeritus (recalled) of Engineering

Instructor EE379A – Winter 2024

Announcements & Agenda

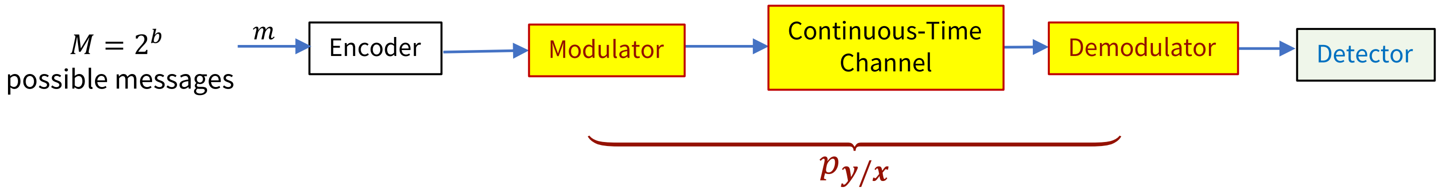
- Announcements
  - Homework – please start/look, due next Wed (1/17)
    - Submit by email to emliang@stanford.edu for now (he'll give to grader).
    - See Homework Helper documents (HWH) at website if needed.
    - Use off hours, help, emails, canvas notes, other students
    - EE379B pages significantly updated

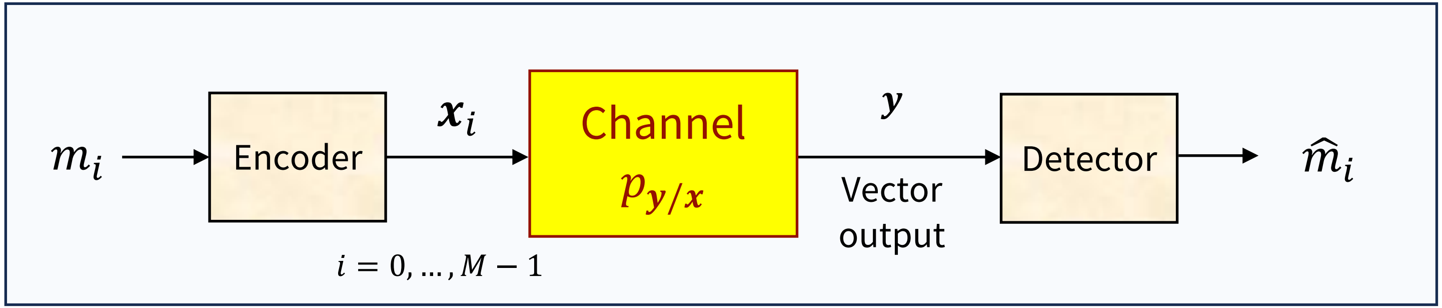
- Today
  - Review, Irrelevance, and Reversibility (1.1.5)
    - Bit-error probability (1.1.6)
  - AWGN Definition/Basics (1.3)
    - ML/MAP on AWGN (1.3.1-2)
  - Pe calculations and bounds (1.3.2)
  - Measures & Fair Comparisons (1.3.3)

Review, Irrelevance, & Reversibility

Section 1.1.5

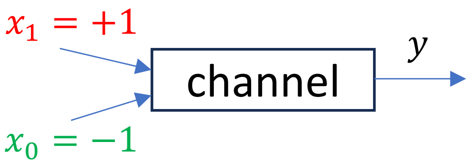
Review of “modem” and discrete channel

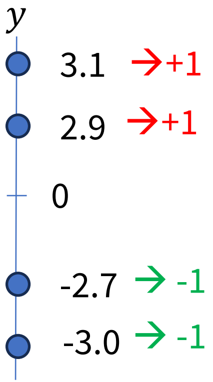




- The modulator/demodulator match to the “analog-world” channel.
  - Apply common analysis to all.
- This forms a strong basis for all modern digital transmission.

MAP Detector Example/Review





$p_{x_0} = p_{x_1}$=0.5 (MAP → ML)

$p_{x/y}$	3.1	2.9	-2.7	-3.0
+1	.9	.85	.1	.2
-1	.1	.15	.9	.8

$$P_{c,MAP} = \mathbb{E}_{x,y}[P_c(y)] = \sum_{i=0}^{M-1} \left\{ \sum_{v \in \mathcal{D}_i} P_{y/m_i}(v\,, m_i) \right\} \cdot \underbrace{p_{m_i}}_{1/2}$$

- This MAP detector minimizes P_e.

The following equations (after scaling each by 2) implement $p_x = \sum_y p_y \cdot p_{x/y}$

$$1 = 1.8 \cdot p_{3.1} + 1.7 \cdot p_{2.7} + .2 \cdot p_{-2.7} + .4 \cdot p_{-3}$$

$$1 = .2 \cdot p_{3.1} + .3 \cdot p_{2.7} + 1.8 \cdot p_{-2.7} + 1.6 \cdot p_{-3}$$

```
% First, compute py/x
>> right=[1.8 1.7 .2 .4 ; .2 .3 1.8 1.6]
   1.8000    1.7000    0.2000    0.4000
   0.2000    0.3000    1.8000    1.6000
>> left=[1 ; 1]
    1
    1
py=pinv(right)*left;
py' =          0.2409    0.2421    0.2597    0.2573
>> sum(py) =    1.0000

>> pygx1=py.*[1.8 ; 1.7 ; .2 ; .4]; % Bayes – move 1/2 to right side
Pygx1'=        0.4336    0.4115    0.0519    0.1029

>> pygxm1=py.*[.2 ; .3 ; 1.8 ; 1.6];
Pygxm1' =   0.0482    0.0726    0.4675    0.4118

>> pygx1*.5 +pygxm1*.5
   0.2409
   0.2421
   0.2597
   0.2573      % same, so far it checks!
```

Proof pinv –see Section 1.1.4, Appendix C on pinv

Example's Pe calculation in matlab

$$P_c = \sum_x \sum_{y \in \mathcal{D}_x} p_{y/x} \cdot p_x$$

```
>> pygx=[pygx1' ; pygxm1'] =
   0.4336   0.4115   0.0519   0.1029
   0.0482   0.0726   0.4675   0.4118
>> pxgy=[.9 .85 .1 .2 ; .1 .15 .9 .8] =
   0.9000   0.8500   0.1000   0.2000
   0.1000   0.1500   0.9000   0.8000
>> pxgy*py =
   0.5000
   0.5000  (checks)
>> pxy = pxgy.*[py' ; py'] =
   0.2168   0.2058   0.0260   0.0515
   0.0241   0.0363   0.2337   0.2059
>> sum(pxy) =
   0.2409   0.2421   0.2597   0.2573
>> sum(sum(pxy)) =    1 % checks
>> px=[0.5 ; 0.5];
>> pygx.*[px px px px] =
   0.2168   0.2058   0.0260   0.0515
   0.0241   0.0363   0.2337   0.2059  %checks again

>> Pc= (pygx(1,1)+pygx(1,2)+pygx(2,3)+pygx(2,4))*.5 =   0.8622
     % sum over only y in decision region (each x)
>> Pe=1-ans =   0.1378
```

Unshaded rows simply Check - unnecessary

- 14% errors is not good, so this channel's input encoding could improve.
- Perhaps the design spec is $P_e \leq 10^{-3}$?
  - Maybe repeat 7 times with majority vote, so need 4 errors?
  - (.1378)^4 $\cdot \binom{7}{4}$= 0.0126 << .1378 .
  - But data rate is $\bar{b} = 1/7$ << 1 ?
- Coding in Chapter 2
  - A good design can xmit at Pe → 0 on this channel with rate $\bar{b} = 0.998$, so almost 1 bit/dimension, with the use of more sophisticated codes.
- In general, a multilevel channel output (more levels than input, e.g. **redundancy**) often provides coding opportunity.

Irrelevant Channel Outputs

- Sometimes extra output components ($\boldsymbol{y}_2$) contribute nothing more than others already used ($\boldsymbol{y}_1$).

Theorem 1.1.3 **[Theorem on Irrelevance]** *If*

$$p_{\boldsymbol{x}/(\boldsymbol{y}_1,\boldsymbol{y}_2)} = p_{\boldsymbol{x}/\boldsymbol{y}_1}$$

or equivalently if the channel-related probability distribution

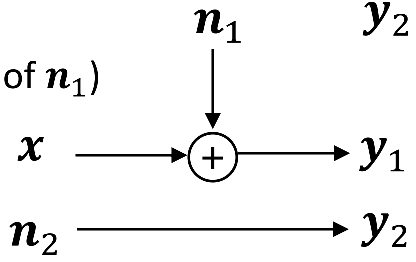
$$p_{\boldsymbol{y}_2/(\boldsymbol{y}_1,\boldsymbol{x})} = p_{\boldsymbol{y}_2/\boldsymbol{y}_1}$$

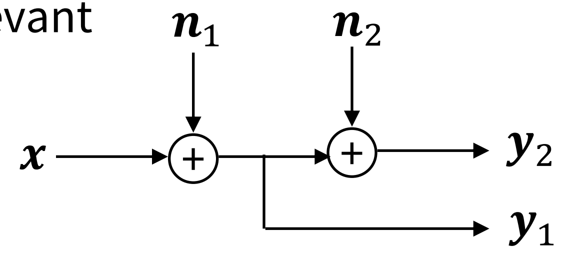
then $\boldsymbol{y}_2$ *is not needed in the optimum receiver, that is,* $\boldsymbol{y}_2$ *is* **irrelevant**.

- Receiver design may discard $\boldsymbol{y}_2$ without loss.

$\boldsymbol{y}_2$ is irrelevant

- Examples ($\boldsymbol{n}_2$ independent of $\boldsymbol{n}_1$)
  - Both independent of $\boldsymbol{x}$

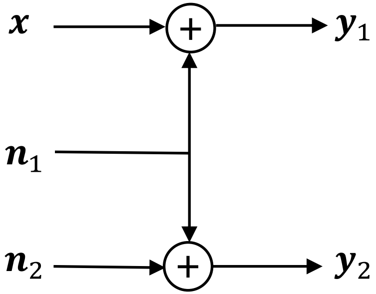




$$p_{y_2/[y_1,x]} = p_{y_2/y_1} = p_{\boldsymbol{y}_2} = p_{\boldsymbol{n}_2}$$

$$p_{y_2/[y_1,x]} = p_{y_2/y_1} = p_{\boldsymbol{n}_2}$$

Nondiscardable output example



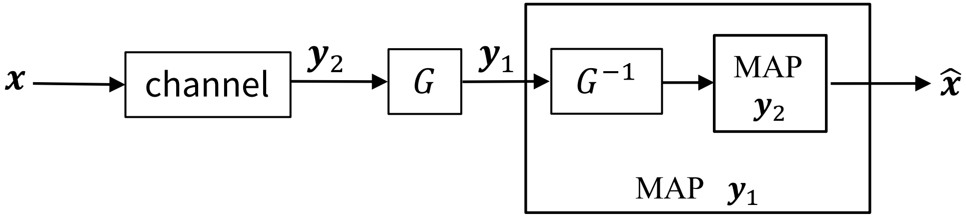
Hmmm .....
y_2 is all noise, must be useless, right?

- **Noise cancellation** - $\boldsymbol{y}_2$ cannot be discarded because it has an $\boldsymbol{n}_1$ component that could be (partially) used to cancel noise in output $\boldsymbol{y}_1$.

- Suppose $\boldsymbol{n}_2 = 0$, then $\boldsymbol{y}_2 = \boldsymbol{n}_1$, which means $\boldsymbol{n}_1$ could be subtracted (cancelled), $\boldsymbol{y}_1 = \boldsymbol{x}$.
  - MAP detector has no errors!

Reversible Transformations are OK

- Special case of irrelevance:



Theorem 1.1.4 [Reversibility Theorem] *The application of an invertible transformation to the channel output vector $\boldsymbol{y}$ does not affect the performance of the MAP detector.*

Proof: Using the Theorem on Irrelevance, if the channel output is $\boldsymbol{y}_2$ and the result of the invertible transformation is $\boldsymbol{y}_1 = G(\boldsymbol{y}_2)$, with inverse $\boldsymbol{y}_2 = G^{-1}(\boldsymbol{y}_1)$ then $[\boldsymbol{y}_1 \ \boldsymbol{y}_2] = [\boldsymbol{y}_1 \ \ G^{-1}(\boldsymbol{y}_1)]$. Then, $p_{\boldsymbol{x}/(\boldsymbol{y}_1,\boldsymbol{y}_2)} = p_{\boldsymbol{x}/\boldsymbol{y}_1}$, which is the definition of irrelevance. Thus, either of $\boldsymbol{y}_1$ or $\boldsymbol{y}_2$ is sufficient to detect $\boldsymbol{x}$ optimally and attain the same minimum error probability or equivalently the same optimum performance.**QED**.

- VERY useful - MAP receiver may simplify greatly after transformation.

Bit-Error Probability Calculation

- Message has b bits $u_j \in \{0\,,1\}$; $j = 1, \ldots, b$.
- Receiver can (instead) use a MAP/ML detector that minimizes bit-error probability.
  - For a MAP that maximizes bit-a-posteriori probability, $p_{u_j \in \{0\,,1\}/y}$.
  - Each bit could have different decision region $\mathcal{D}_j(\boldsymbol{y})$ ➔ more complex , or could just use common $\mathcal{D}\,(\boldsymbol{y})$ for P_e .

$$\begin{aligned}\bar{P}_{b,j} &= \Pr\{\hat{u}_j \neq u_j\} = 1 - \sum_{u_j=0}^{1}\left[\sum_{\boldsymbol{v}\in\mathcal{D}_j} P_{\boldsymbol{y}/u_j}(\hat{u}_j = u_j, \boldsymbol{v})\right]\cdot p_{u_j} \\ &= 1 - \sum_{u_j=0}^{1}\left[\sum_{\boldsymbol{v}\in\mathcal{D}_j} p_{u_j|\boldsymbol{y}}(u_j, \boldsymbol{v})\cdot p_{\boldsymbol{y}}(\boldsymbol{v})\right] \quad .\end{aligned}$$

- $\bar{P}_{b,j}$ can vary with j.
- LLR's often used for MAP of each bit:

Definition 1.1.5 [Log Likelihood Ratio (LLR)] *A log likelihood ratio for a bit u_j is the logarithm of probability ratio that bit takes the values 0 and 1. Often convention considers the bit value 0 as correct and the bit value 1 as incorrect, without loss of generality.*

$$\begin{aligned} LLR_{u_j}(\boldsymbol{v}) &\triangleq \ln\left(\frac{P_{u_j=0}(\boldsymbol{v})}{P_{u_j=1}(\boldsymbol{v})}\right) \\ &= \ln\left(\frac{\sum_{\boldsymbol{u}\setminus u_j} p_{\boldsymbol{y}|\boldsymbol{u}}(\boldsymbol{v}, \boldsymbol{u}\,|_{u_j=0})\cdot p_{\boldsymbol{u}}(\boldsymbol{u}\,|_{u_j=0})}{\sum_{\boldsymbol{u}\setminus u_j} p_{\boldsymbol{y}|\boldsymbol{u}}(\boldsymbol{v}, \boldsymbol{u}\,|_{u_j=1})\cdot p_{\boldsymbol{u}}(\boldsymbol{u}\,|_{u_j=1})}\right) \quad .\end{aligned}$$

When $u_j \to \hat{u}_j$ above for a decoder with average bit-error rate $\bar{P}_b$, then

$$LLR_{\hat{u}_j} = \ln\frac{1-\bar{P}_{b,j}}{\bar{P}_{b,j}} \qquad (1.28)$$

- Examples later in course

Bhattacharya Bound

- **B-Bound** is based on the MAP detector's specific **error events** (needs to be averaged over all such events).

$$P\{\varepsilon_{m\tilde{m}}\} \triangleq Pr\{\boldsymbol{x}_m \rightarrow \boldsymbol{x}_{\tilde{m}}\}$$

$$P\{\varepsilon_{m\tilde{m}}\} \leq \sum_{\boldsymbol{v}} \sqrt{p_{\boldsymbol{y}/\boldsymbol{x}}(\boldsymbol{v}, \boldsymbol{x}_{\tilde{m}}) \cdot p_{\boldsymbol{y}/\boldsymbol{x}}(\boldsymbol{v}, \boldsymbol{x}_m)}$$

- Simple proof – see text, Section 1.1.7 ; most often used for symbols that are groups of bits (codewords).

- When $\bar{P}_{b,j} \equiv p$, the error event's B-Bound takes the simple, often-encountered, form where $\boldsymbol{x}_m$ and $\boldsymbol{x}_{\tilde{m}}$ differ in only **Hamming distance** d_H positions:

$$P\{\varepsilon_{m\tilde{m}}\} \leq [4 \cdot p \cdot (1-p)]^{d_H/2}$$

- Suggestive of an **inner channel** where a first decision is made with $\bar{P}_{b,j} \equiv p$, and then a second **outer** bit-level code is present.
  - Symbols are bit vectors carefully chosen to have separation (two MAP decoders).

- For binary channels where the error probability interchanges for 1 and 0 as the only two messages, Griot, Weng, and **Wesel** (GWW) tightened the B-Bound to $P\{\varepsilon_{m\tilde{m}}\} \leq \frac{1}{2}[4 \cdot p \cdot (1-p)]^{d_H/2}$.
  - The extra factor of 1/2 can overly burden some analyses an so original B-Bound form is often used.

Return to Example on slide 5

- $\bar{P}_{b,j} \equiv p$ on this channel was equal to $P_{e,inner}$ =.1378 .
- Simple outer code with $\bar{b} = 1/7$ (so all zeros or all ones) would have:

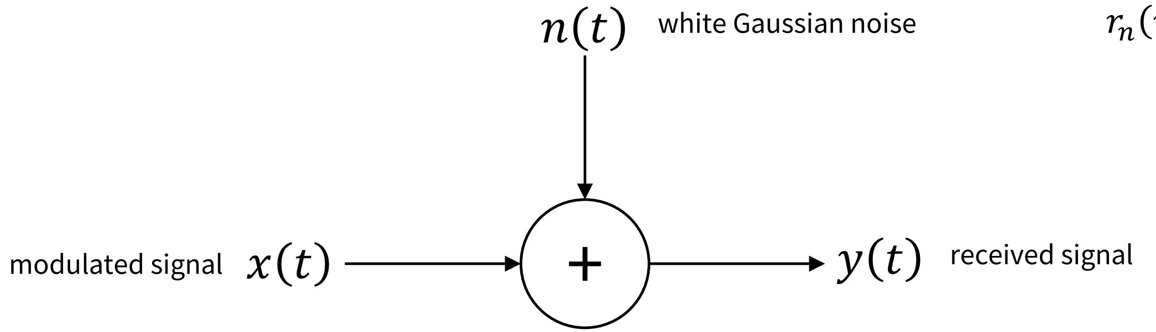
$$P_{e,outer} \leq 1/2 \cdot [4 \cdot .1378 \cdot .8622]^{7/2} = 0.0012$$

(pretty close to value on slide L2:6)

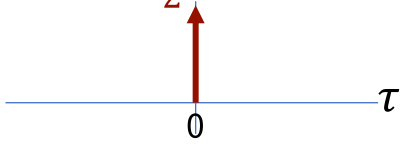
The AWGN

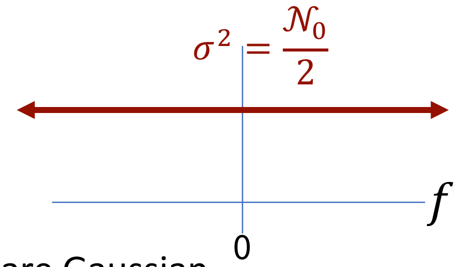
Section 1.3

Additive White Gaussian Noise (AWGN)



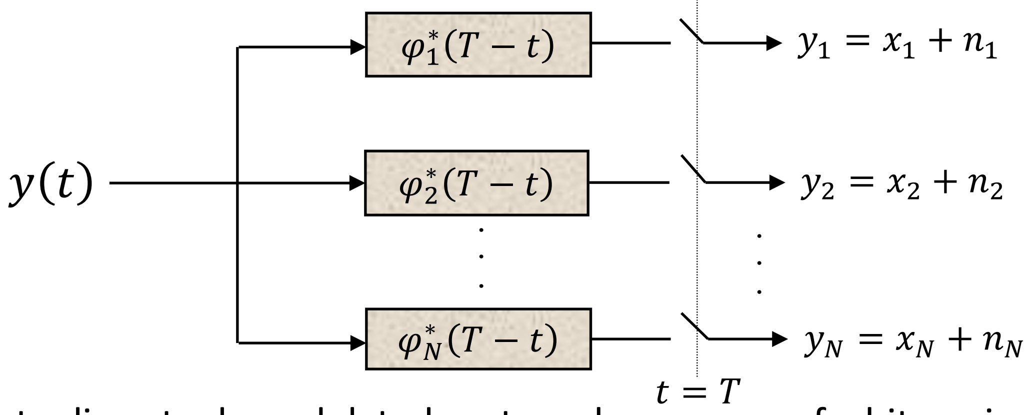
$$r_n(\tau) \triangleq \mathbb{E}[n(\tau) \cdot n\,(t-\tau)] = \frac{\mathcal{N}_0}{2} \cdot \delta(\tau)$$





- This channel is the most common in all wireless/wireline communication.
  - Most variants use it as foundation.
- Central Limit Theorem (see A.1.8) – large number of noise events added together are Gaussian.
  - White means "all frequencies equal" power spectra density, or $S_n(f) = \frac{\mathcal{N}_0}{2} = \mathfrak{F}\{r_n(\tau)\}$ (Fourier Transform).
  - That is "flat."
  - Flat has the same energy σ^2 on any basis function (on any dimension).
- Thermal Noise ($k_{boltz} \cdot Temp$) = -174 dBm/Hz at room temperature) is an example, but also:
- analogy amplifiers front-end noise, ADC quantization noise (with enough bits quantizing), or even
- many crosstalking interference signals from other sources (especially if they use good "Gaussian" codes).

AWGN discrete channel loses nothing!



$$R_{\boldsymbol{nn}} \triangleq \mathbb{E}[\boldsymbol{n} \cdot \boldsymbol{n}^t] = \frac{\mathcal{N}_0}{2} \cdot I$$

Autocorrelation matrix,
See Appendix D.1

$$R_{\boldsymbol{yy}} = R_{\boldsymbol{xx}} + R_{\boldsymbol{nn}}$$

- This conversion to discrete demodulated vector $\boldsymbol{y}$ loses some of white noise, but that part is irrelevant.
- See proof in Section 1.3.1.

Rule 1.3.1 **[The Vector AWGN Channel]** *The vector AWGN channel is given by*

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{n} \tag{1.85}$$

and is equivalent to the channel illustrated in Figure 1.26. The noise vector $\boldsymbol{n}$ *is an* N*-dimensional Gaussian random vector with zero mean, equal-variance, uncorrelated components in each dimension. The noise distribution is*

$$p_{\boldsymbol{n}}(\boldsymbol{u}) = (\pi\mathcal{N}_0)^{-\frac{N}{2}} \cdot e^{-\frac{1}{\mathcal{N}_0}\|\boldsymbol{u}\|^2} = \left(2\pi\sigma^2\right)^{-\frac{N}{2}} \cdot e^{-\frac{1}{2\sigma^2}\|\boldsymbol{u}\|^2} \quad . \tag{1.86}$$

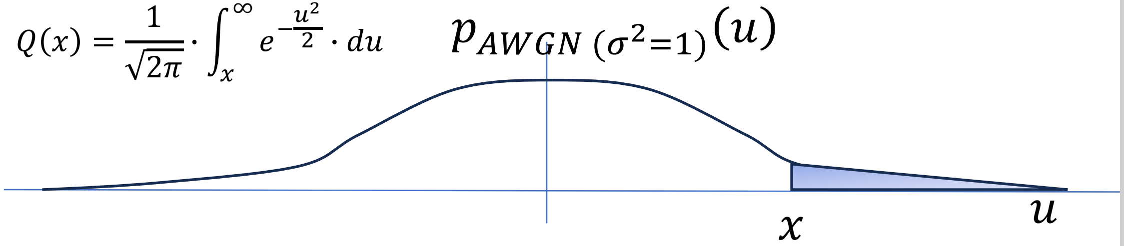
- MAP for $\boldsymbol{y}$ still minimizes P_e.
- For AWGN Channel, analysis need not know the modulator type

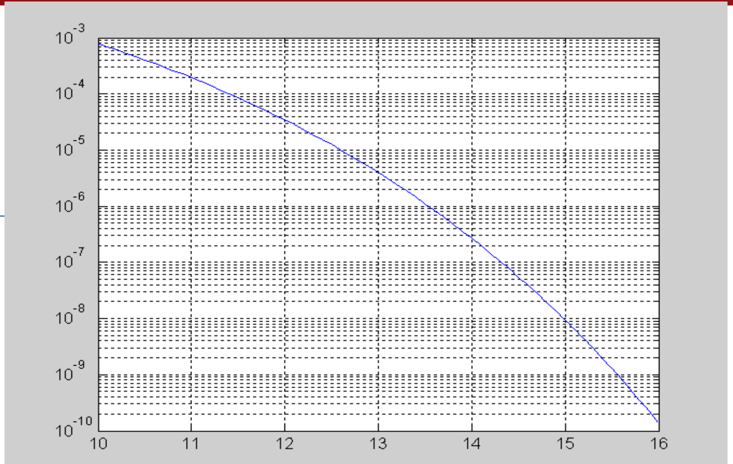
Irrelevant White-Noise Concept

- Clearly independent noise where there is no signal does not help estimate x

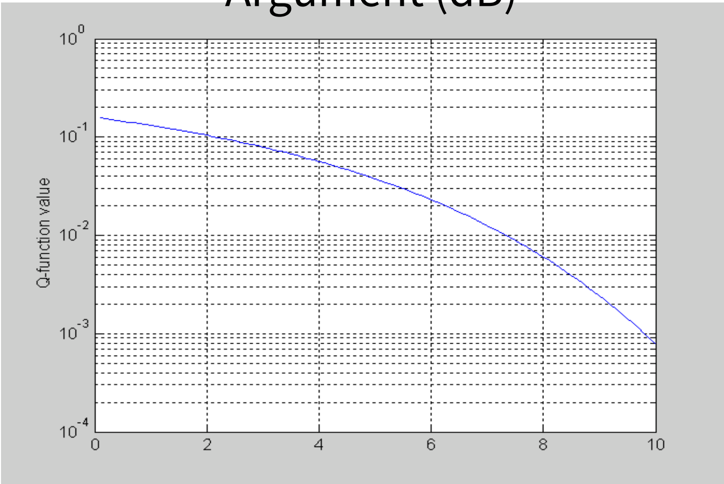
The Q-function (A.1.7)

$$Q(x) = \frac{1}{\sqrt{2\pi}} \cdot \int_x^{\infty} e^{-\frac{u^2}{2}} \cdot du$$





Argument (dB)



- $Q(x)$=Pr$\left\{n \geq \frac{x}{\sigma}\right\}$
- Measures probability that AWGN noise exceeds a certain level (relative to standard deviation)
- Matlab

```
function qfunc(x);
 Computes the q function
 qfunc(x) = .5 erfc(x/sqrt(2))

>> qfunc(1) =   0.1587
>> qfunc(3) =   0.0013
>> qfunc(5) =  2.8665e-07
>> qfunc(10^(13.5/20)) =  1.1143e-06  (13.5 dB → 10^-6)

>> 20*log(3) =  21.9722 dB  (sample amplitude)
>> 10*log(3^2) =  21.9722 dB (energy)
```

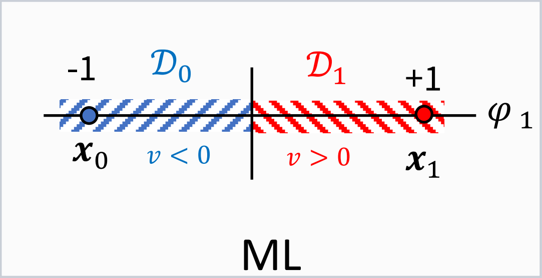
MAP/ML on the AWGN

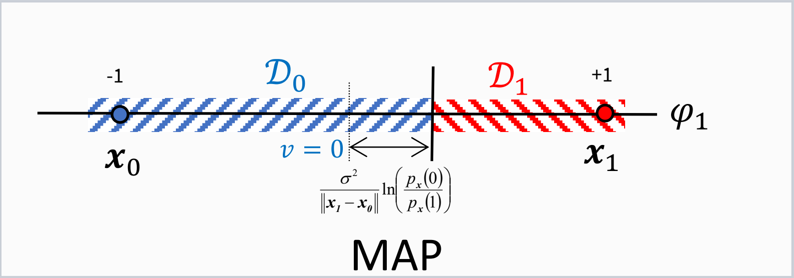
- ML AWGN detector (if the inputs are not equally likely, your source can be improved):

Rule 1.3.3 [AWGN ML Detection Rule]

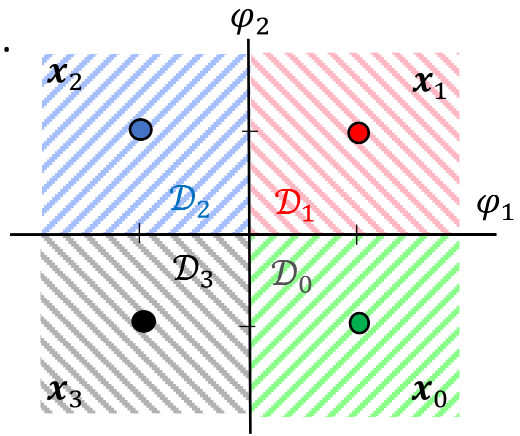
$$\hat{m} \Rightarrow m_i \;\; if \;\; \|\boldsymbol{v} - \boldsymbol{x}_i\|^2 \leq \|\boldsymbol{v} - \boldsymbol{x}_j\|^2 \;\; \forall \;\; j \neq i \quad .$$

- ML is simple (conceptually) with AWGN – pick the closest symbol value (to $\boldsymbol{y}$).



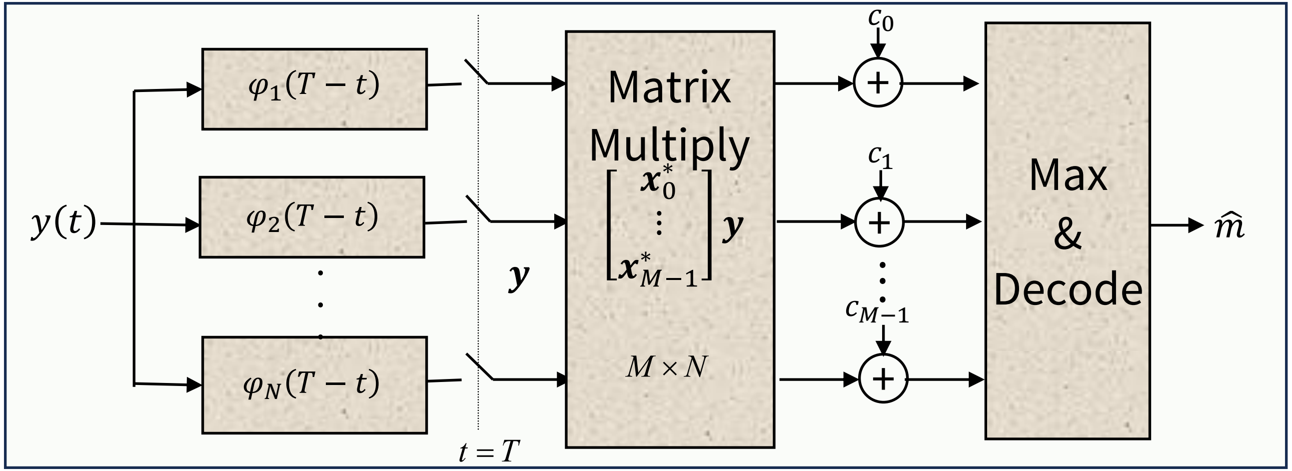


4SQ (QPSK)



AWGN ML Detector

- Implement closest point? $\hat{m} \Rightarrow m_i \;\; if \;\; \langle \boldsymbol{y}, \boldsymbol{x}_i \rangle + c_i \geq \langle \boldsymbol{y}, \boldsymbol{x}_j \rangle + c_j \; \forall j \neq i$

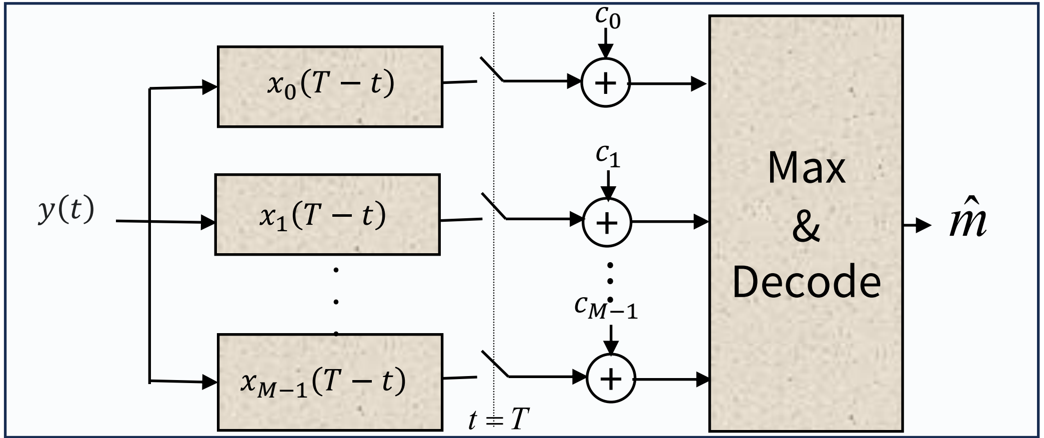


$$c_i \triangleq \frac{\mathcal{N}_0}{2} \cdot \ln\{p_{\boldsymbol{x}}(i)\} - \frac{\|\boldsymbol{x}_i\|^2}{2}$$

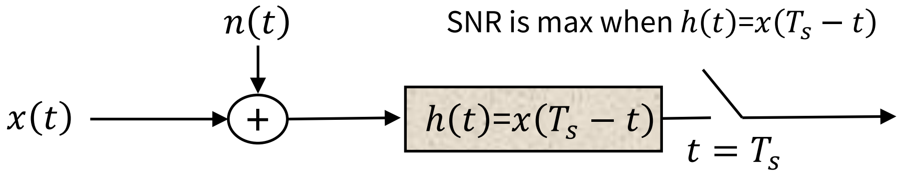
"Machine Leaners" – recognize this?

It is a ReLU (rectified linear unit, 1.5.2; where we already know the coefficients, bias terms, and use hard nonlinearity.

- Or perhaps simpler?



Matched Filter SNR Maximization

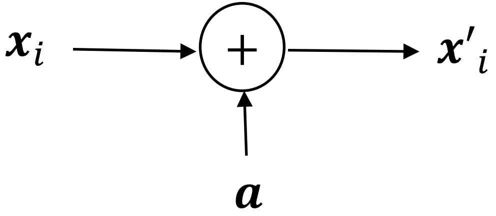


- Convolution reverses and multiplies – when the two convolved signals align perfectly, the SNR is largest.
- Matched filter essentially aligns filter with the pre-noise signal ("matched") to align/boost maximally.
- MF is fundamental in many detection/receiver strategies.
- See Section 1.3.1 (proof is there).

Pe Calculations and Bounds

(Section 1.3.2)

Translational Invariance (AWGN)



- This does not change the distances between constellation points. (It is a reversible transformation.)

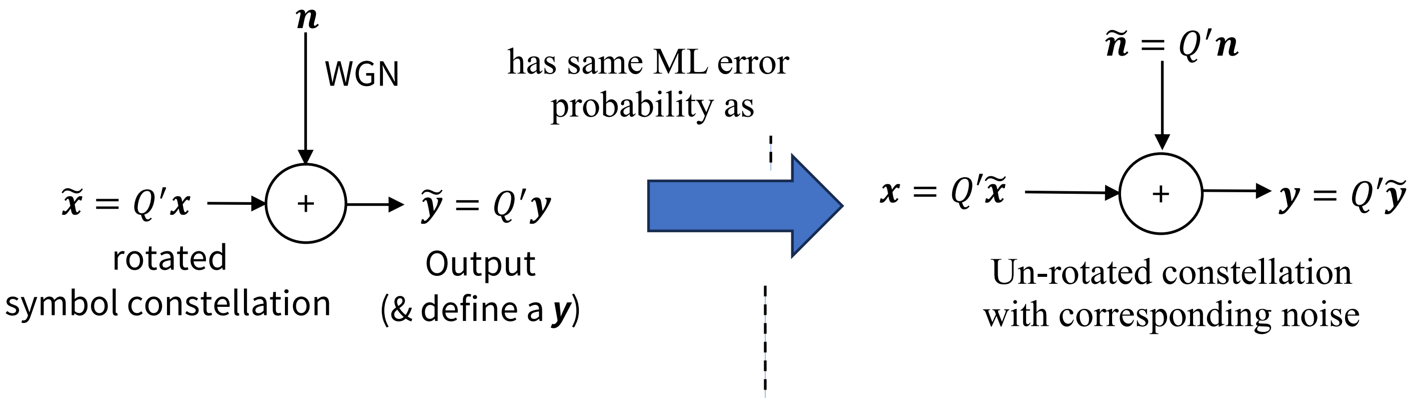
- Thus, the detector simply subtracts the known $\boldsymbol{a}$ so $\mathbf{y} \rightarrow \mathbf{y} - \boldsymbol{a}$ and proceeds the same way (note this is 1-to-1 so reversibility applies).

- **Minimum Energy translate?**

> **Definition 1.3.1 [Minimum-Energy Translate]** *A constellation's* **minimum energy translate** *is obtained by subtracting the constant vector $E\left[\boldsymbol{x}\right]$ from each data symbol.*

- Section 1.3.2.1 – saves energy, no performance loss (AWGN) – PS1.5's "tilt."

Rotational Invariance (AWGN)



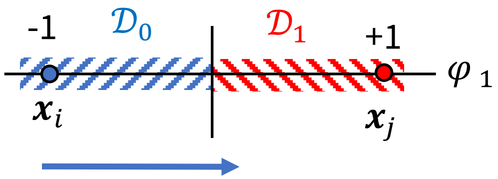
- The unitary matrix $Q \cdot Q^* = I = Q^* \cdot Q$ is 1-to-1 (reversible).
- Also, the energy does not change $\mathbb{E}[\llbracket \boldsymbol{x} \rrbracket^2] = \mathbb{E}[\llbracket Q \cdot \boldsymbol{x} \rrbracket^2]$.

AWGN ML-Detect performance is invariant to rotation and/or translation.

Minimum Distance & AWGN's Union Bound

Definition 1.3.2 [Minimum Distance, $d_{\min}$] *The* **minimum distance** *for a constellation with symbol vectors* $\boldsymbol{x} \triangleq \{\boldsymbol{x}_i\}_{i=0,\ldots,M-1}$, *is* $d_{\min}(\boldsymbol{x})$ *and measures the smallest separation between any two different constellation symbol (or "codeword") values. The argument* $(\boldsymbol{x})$ *is often dropped when the specific signal constellation is obvious from the context, thus leaving*

$$d_{\min} \triangleq \min_{i \neq j} \|\boldsymbol{x}_i - \boldsymbol{x}_j\| \quad \forall\, i,j \quad . \tag{1.122}$$



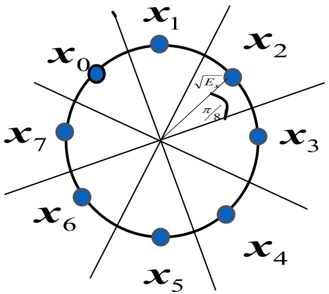
$$P_e \leq \underbrace{(M-1)}_{\text{no more than } M-1 \text{ events}} \cdot \underbrace{Q\left(\frac{d_{min}}{2\sigma}\right)}_{\text{noise big enough}}$$

$\boldsymbol{x}_i + n$ this big causes an error, or $|n| > \frac{d_{min}}{2}$

- **Union Bound** is thus $P_e \leq (M-1) \cdot Q\left(\frac{d_{min}}{2\sigma}\right)$
- At worst, any constellation point can be confused for any of the $M-1$ other points, which are at least at minimum distance.

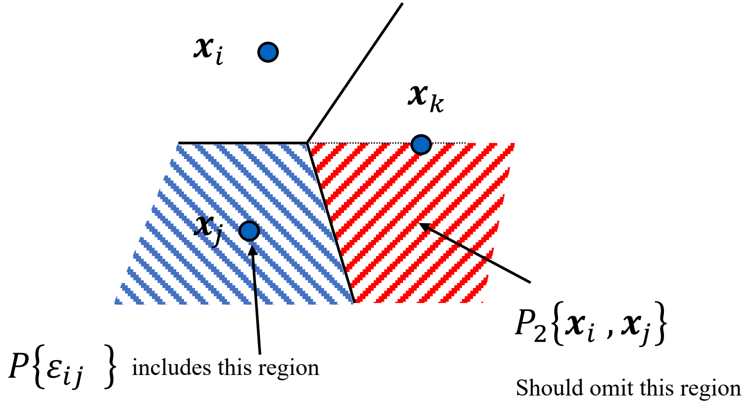
Example 8 PSK

- 8 messages $(M = 8\,, b = 3\,; N = 2)$ - same basis functions as BPSK (and QPSK)



$$P_e < 7 \cdot Q\left(\frac{d_{min}}{2\sigma}\right)$$

- Overkill? Well, yes, especially if M is large.
- Double counts the red-shaded region for ε_{ij}
- Can you do better?
- How about $P_e < 2 \cdot Q\left(\frac{d_{min}}{2\sigma}\right)$?



Nearest Neighbor Union Bound

- $P_e < N_e \cdot Q\left(\frac{d_{min}}{2\sigma}\right)$.

- Where $N_e \triangleq$ the **number of nearest neighbors .**

Definition 1.3.3 [Average Number of Nearest Neighbors] *A constellation's average number of nearest neighbors, N_e is*

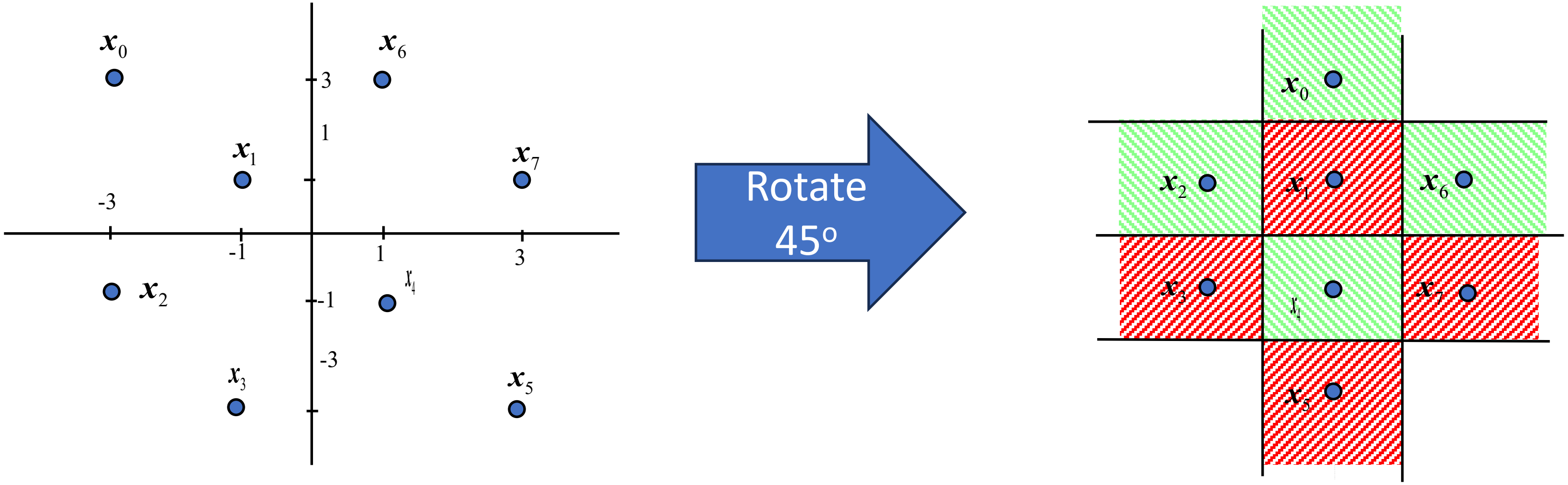
$$N_e \triangleq \sum_{i=0}^{M-1} N_i \cdot p_{\boldsymbol{x}}(i) \quad , \tag{1.147}$$

where N_i is the symbol $\boldsymbol{x}_i$'s number of neighboring constellation symbols, that is the number of other symbol vectors sharing a common decision region boundary[28] *with $\boldsymbol{x}_i$.*

- Count only those at distance d_{min} (often easier to do) $N_e \cong \sum_{i=0}^{M-1} \breve{N}_i \cdot p_{\boldsymbol{x}}(i)$; $\breve{N}_i$ counts only @ dmin .

- $P_e \cong N_e \cdot Q\left(\frac{d_{min}}{2\sigma}\right)$ **this is the nearest-neighbor union bound** (usually very tight, often used).

NNUB Example 8SQ



- This analysis counts only (so approximates) neighbors at minimum distance.
  - Others have smaller contribution.
- $P_e < 3.25 \cdot Q$

$$
\begin{aligned}
P_c &= \sum_{i=0}^{7} P_{c/i} \cdot p_{\boldsymbol{x}}(i) = \sum_{i \neq 1,4} P_{c/i} \cdot \frac{1}{8} + \sum_{i=1,4} P_{c/i} \cdot \frac{1}{8} \\
&> \frac{6}{8}(1-Q)(1-2Q) + \frac{2}{8}(1-2Q)^2 \\
&= \frac{3}{4}\left(1-3Q+2Q^2\right) + \frac{1}{4}\left(1-4Q+4Q^2\right) \\
&= 1 - 3.25Q + 2.5Q^2 \quad .
\end{aligned}
$$

Measures & Fair Comparisons

See 1.3.2.4

Calculate ave bit error prob (symbol decoder)

Definition 1.3.4 [**Average Bit Error Rate**] *The* **average bit error rate** *is*

$$P_b \triangleq \sum_{i=0}^{M-1} \sum_{\substack{j \\ j \neq i}} p_{\boldsymbol{x}}(i) \cdot P\{\varepsilon_{ij}\} \cdot n_b(i,j) \quad (1.152)$$

where $n_b(i,j)$ is the number of bit errors for the particular choice of encoder when symbol i is erroneously detected as symbol j. This quantity, despite the label using P, is not strictly a probability. It also related to the average total number of bit errors per error event.

- Average Total Bit Errors per Error Event

$$N_b = \sum_{i=0}^{M-1} p_x(i) \cdot n_b(i) \quad \text{where} \quad n_b(i) = \sum_{j=0}^{N_i} n_b(i,j)$$

- Then

$$P_b \approx N_b \cdot Q\left[\frac{d_{\min}}{2\sigma}\right]$$

$$\begin{aligned}
P_b &\approx \sum_{i=0}^{M-1} \sum_{j=1}^{N_i} p_{\boldsymbol{x}}(i) \cdot P\{\varepsilon_{ij}\} \cdot n_b(i,j) \\
&\leq Q\left[\frac{d_{\min}}{2\sigma}\right] \cdot \sum_{i=0}^{M-1} p_{\boldsymbol{x}}(i) \sum_{j=1}^{N_i} n_b(i,j) \\
&\lesssim Q\left[\frac{d_{\min}}{2\sigma}\right] \cdot \sum_{i=0}^{M-1} p_{\boldsymbol{x}}(i) \cdot n_b(i) \\
&\lesssim N_b \cdot Q\left[\frac{d_{\min}}{2\sigma}\right]
\end{aligned}$$

$$n_b(i) \triangleq \sum_{j=1}^{N_i} n_b(i,j) \quad ,$$

Lemma 1.3.2 [**Average bit-error probability $\bar{P}_b$.**] *The* **average probability of bit error** *is defined by*

$$\bar{P}_b = \frac{P_b}{b} \quad . \quad (1.155)$$

The corresponding average total number of bit errors per bit is

$$\bar{N}_b \triangleq \frac{N_b}{b} \quad . \quad (1.156)$$

Normalization to Dimensionality

$$\bar{A} \triangleq \frac{A}{N}$$

- $\bar{P}_e$, $\bar{N}_e$, $\overline{\mathcal{E}_x}$, $\bar{b}$, ...

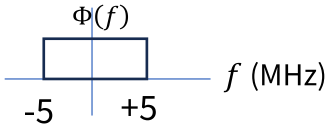
- The measure, indicated by a bar, normalizes to the "resources" - the number of dimensions.

Fair Comparisons

Many engineers, including some really famous ones, have erred on comparisons

- Fix 4 of these 5 – compare last
  - Data rate $R = \frac{b}{T}$
  - Power $P_x = \frac{\mathcal{E}_x}{T}$
  - System bandwidth W
  - Total transmission time or symbol period T
  - Error probability P_e

System 1: $R = 10$ Mbps, $N = 1, \pm 1\ (M = 2), \frac{1}{T}$=10 MHz

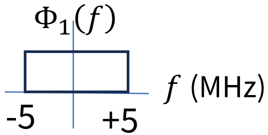


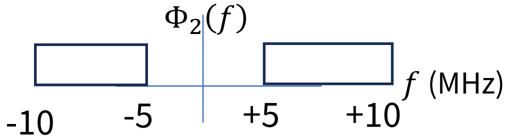
$W = 5$ MHz

$d_{min} = 2$

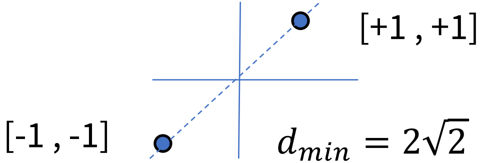
- Or fix 2 of these 3 – compare the last
  - Bits/dim $\bar{b} = \frac{b}{N}$
  - Energy/dim $\bar{\mathcal{E}}_x = \frac{\mathcal{E}_x}{N}$
  - Error Prob/dim $\bar{P}_e = \frac{P_e}{N}$

System 2: $R = 20$ Mbps, $N = 2, \pm 1\ (M = 2), \frac{1}{T}$=10 MHz





$W = 10$ MHz



$d_{min} = 2\sqrt{2}$

Bigger d_{min} and twice rate ?
Not fair, both $\bar{b}$ and $\bar{P}_e$ differ
(these two are really the same)
set $\bar{b}$ =1 and becomes QPSK, smaller d_{min}

Packet Error Rate

- **Block error rate**, $P_{e,block}$, is the average probability that a packet or "block" of B bits contains at least one erred bit.

$$P_{e,block} \approx B \cdot \bar{P}_b$$

- More accurately, $P_{e,block}$ counts the ways that bit errors can occur:

$$\sum_{i=1}^{B} \binom{B}{i} (1 - \bar{P}_b)^{B-i} \bar{P}_b^i$$

- Other examples
  - An **erred second** is second in which at least one (uncorrectable) bit error occurred.
  - **Code violations** – usually measured in 15min intervals -- and count the number of erred packets in that interval.

End Lecture 2

- (Reversible) Transformations to simplify detectors
- AWGN channel simplifications for ML detectors.
- Pe calculations
- Fair analysis