STANFORD

Lecture 6

Coding Concepts & Dimensionality

January 25, 2024

JOHN M. CIOFFI

Hitachi Professor Emeritus (recalled) of Engineering

Instructor EE379A – Winter 2024

Announcements & Agenda

- Announcements
  - PS3 – problem 3.2 may take some runtime for matlab on P_e estimates, so give yourself time.
  - Again, recall HWH (HWH3 is at web site) if spending too much time.

- Today
  - Codewords, Symbols, and Redundancy
  - Random Coding: AWGN's Sphere Packing
  - DMC Codes: MDS' Ball Packing

EE379A Lectures – Winter 2024

Tu-Th 3:00 - 4:20 pm; Location Gates B1

Lecture #	Date	Topic	Reading	Hmwrk (out/in)
		Data-Transmission, Channels & Fundamentals		
1	1/9	Intro: Discrete Message Encoding/Decoding	1.1	1/-
2	1/11	White Gaussian Noise (AWGN) Channels	1.2	-/-
3	1/16	Modulation Types (PAM/QAM)	1.3	2/1
4	1/18	Complex and other Channels	1.4	-/-
5	1/23	MIMO and Statistical Channels	1.5, 1.6	3/2
		Codes and Decoding		
6	1/25	Coding Concepts & Dimensionality	2.1-2	-/-
7	1/30	Binary Codes	8.1,8.2	4/3
8	2/1	Viterbi-Sequence & MAP-Bit Decoding	7.1-3	-/-
9	2/6	Concatenated and Turbo Codes	8.3	-/4
--	2/8	**Midterm Exam** (open bk)		-/-
10	2/13	Constraints and LDPC Codes	7.4-6	5/-
11	2/15	Outer Hard-Code Concatenation	8.4,8.6	-/-
12	2/20	Guessing Decoders & Product Codes	7.6, 8.3.5	6/5

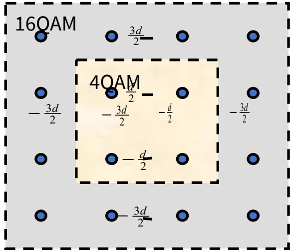
Codewords, Symbols, & Redundancy

Section 2.1

AWGN Summary/Review

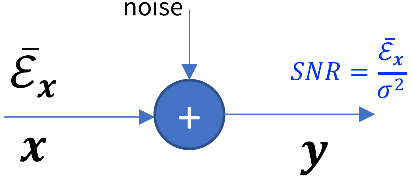
Detection problem first, every T seconds (symbol period)

symbol/codeword
(M=16, # of messages)

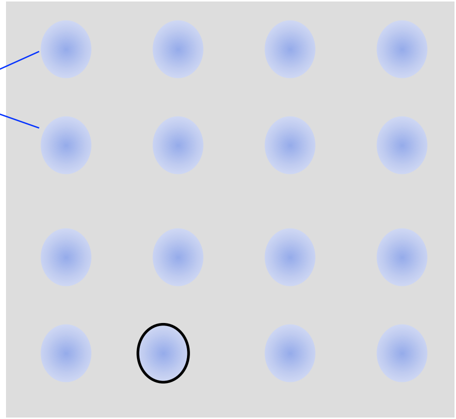


$n \sim \sigma^2 \left(=\frac{\mathcal{N}_0}{2}\right)$

noise



channel-output codeword



$\hat{x}$

- Min error probability P_e
- Pick closest point for AWGN
- Maximum Likelihood
  - Uniform input dist'n

$\mathcal{E}_x = \mathbb{E}[|x|^2]$ $\quad p_x$

$p_{y/x}$

$= p_n(y - x)$

$\max_{\hat{x}=x} p_{y/x}$

"SISO"
(scalar)
$N = 1 \quad \text{or} \quad 2$

- QAM → 2 dimensional
- Uniform input (usually) $p_x = \frac{1}{M}$
- $b = \log_2 M$ **bits/symbol**
- $R = \frac{b}{T}$ bits/second (**data rate**)

- Add noise
- Zero mean
- Variance σ^2 (= 2-sided PSD)

$$P_e = 4 \cdot \left(1 - \frac{1}{\sqrt{M}}\right) \cdot Q\left(\sqrt{\frac{3 \cdot SNR}{M-1}}\right)$$

Subsymbol if coded (Slide L6:6)

$x \to \tilde{x} \in \mathbb{R}\ ;\ x \to \tilde{x} \in \mathbb{C}$

x has N real dimensions in general,
and has $\bar{N}$ subsymbols, of dim $\tilde{N}$

SNR, QAM, PAM reminders

$$SNR \triangleq \frac{\bar{\mathcal{E}}_x}{\sigma^2} = \frac{\text{single} - \text{sided psd}}{\text{single} - \text{sided psd}} = \frac{\text{two} - \text{sided psd}}{\text{two} - \text{sided psd}}$$

- SNR must use the same number of dimensions in numerator (signal) and denominator (noise).
- Thus, also $SNR \triangleq \frac{\bar{\mathcal{E}}_x}{\sigma^2} = \frac{2 \cdot \bar{\mathcal{E}}_x}{\mathcal{N}_0} = \frac{\mathcal{E}_x}{N \cdot \sigma^2}$ where $\bar{\mathcal{E}}_x$ is energy/real-dimension.
- Energy/dimension generalizes power/Hz (= energy), so equivalent to a power-spectral density (psd).
  - 1-sided → power is integral over positive frequencies of psd.
  - 2-sided → power is integral over all frequencies of psd.
  - These two powers are the same.
  - So -40 dBm/Hz (one-sided) psd over 1 MHz is 20 dBm, or 100 mWatts of power.
- PAM is always real baseband. QAM is always complex baseband (2 real dimensions)
  - **When QAM** has only 1 bit (2 points) in constellation, it is called BPSK (not binary PAM).
  - PAM's positive-frequency bandwidth is [0, 1/2T) .
  - QAM's positive-frequency bandwidth is [-1/2T + f_c , 1/2T + f_c) .
  - The PAM system looks like it uses only 1/2 the bandwidth, but the QAM system is really transmitting two dimensions per symbol (so really like 2 PAM systems in parallel with symbol rate 1/T each), so then twice a single PAM's bandwidth.

Codewords constructed from "subsymbols"

$\tilde{\boldsymbol{x}}_1$	$\tilde{\boldsymbol{x}}_2$	$\tilde{\boldsymbol{x}}_3$	$\cdots$	$\tilde{\boldsymbol{x}}_{\bar{N}}$

codeword (symbol) $\boldsymbol{x}$

Good Code $\tilde{b} \to \mathcal{C}$ as $\bar{N} \to \infty$

subsymbols $N = \bar{N} \cdot \tilde{N} =$ # subsymbols x (dim/subsymbol)

Detector could also detect subsymbols
See PS3.3 (2.3)

bits/dim = $\bar{b} = {}^{b}/_{N}$; **bits/subsym =** $\tilde{b} = {}^{b}/_{N} = \tilde{N} \cdot \bar{b}$

Code construction

- QAM/PAM operates with given low P_e (10^{-6}) and at a "SNR gap" ($\Gamma = 8.8\ dB$ @10^{-6}) below capacity.
  - See basics in Section 1.3.4.

$$\tilde{b} = log_2\left(1 + \frac{SNR}{\Gamma}\right) \text{ bits/complex-subsymbol } \leq \mathcal{C}$$

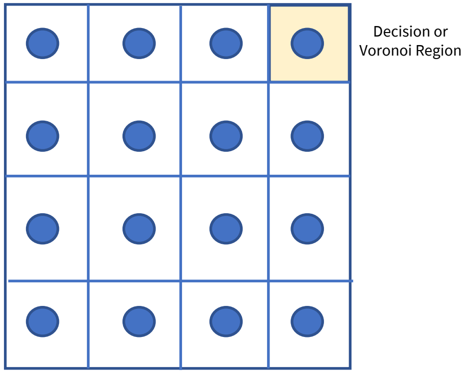
$\frac{3}{2^{\tilde{b}}-1} \cdot SNR = 13.5\ dB$ (from $P_e = 10^{-6}$ formula)

- For all $\tilde{b} > 1$, simple square QAM constellations have constant gap (= 8.8 dB at $P_e = 10^{-6}$).

- The subsymbols are QAM, but usually with more than $|C| > M = 2^{\tilde{b}}$ possible values (redundancy).

Trivial Coding

16 SQ QAM - UNCODED



$|C| = 16\,; M = 16$

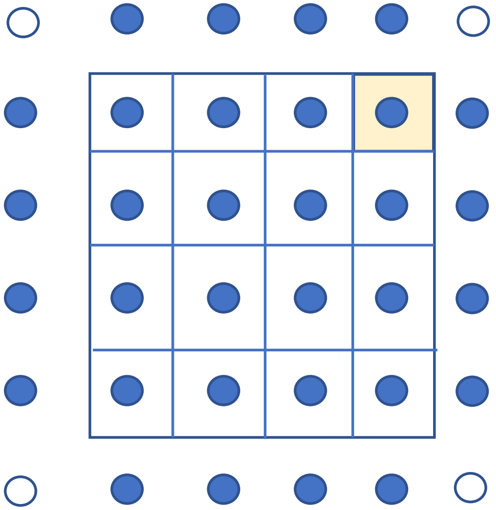
$b = 4\,;\ \bar{b} = 2\,;\ \tilde{b} = 4$

$N = \tilde{N} = 2\,;\ \bar{N} = 1$ (or swap $\tilde{N}$ and $\bar{N}$ → PAM)

32 CR QAM – UNCODED in 2D

$|C| = 32\,; b = 5\,;\ \bar{b} = 2.5$

$N = \tilde{N} = 2\,;\ \bar{N} = 1$



6 PAM x 6 PAM 36SQ CODED in 1D

$N = \bar{N} = 2\,;\ \tilde{N} = 1\,;\ |C| = 6\ = 2^{2.59}$

If $\bar{b}\ = 2.5\ , \bar{\rho} = 0.09$

(extra constellation points ~ redundancy)

$\bar{b} + \bar{\rho} = \log_2|C|$

- The subsymbol can have extra points, which means its coded.
  - More redundant points and/or more dimensions → better codes.
  - The subsymbol values may no longer be equally likely, but codewords are.

Redundancy and uncoded definition

Definition 2.1.2 (Code) *A* **code** *is any set of $M = 2^b$, N-dimensional* **codewords**

$$C_{\boldsymbol{x}} = \{\boldsymbol{x}_i\}_{i=0,\ldots,M-1} \tag{2.7}$$

where the N-dimensional codewords have $\overline{N}$, $\tilde{N}$-dimensional **subsymbols** *selected from an $\tilde{N}$-dimensional subsymbol constellation C with $|C|$ subsymbol values. The subscript $\boldsymbol{x}$ on $C_{\boldsymbol{x}}$ distinguishes $C_{\boldsymbol{x}}$ from the subsymbol constellation C . Thus,*

$$N = \underbrace{\tilde{N}}_{\text{subsymbol size}} \cdot \underbrace{\overline{N}}_{\#\text{ of subsymbols}} \quad . \tag{2.8}$$

- **Subsymbols** are basically now what our earlier efforts (and other texts) were calling QAM (or PAM) symbols.
  - Note $M^{1/\tilde{N}} \leq |C|$ - they are equal when "uncoded."
- The subsymbol dimensionality is $\tilde{N}$,and there are $\overline{N}$ such $\tilde{N}$ −dimensional subsymbols / symbol.
  - When $\tilde{\boldsymbol{x}}_n$ has $\tilde{N} = 2$, then $\boldsymbol{x} \in \mathbb{C}^N$; when $\tilde{\boldsymbol{x}}_n$ has $\tilde{N} = 1$, then $\boldsymbol{x} \in \mathbb{R}^N$.
- The symbols are codewords. The quantities d_{min} and P_e refer to this symbol/codeword error.
  - As before, more complicated decoders can focus on subsymbol-error or bit-error probabilities, as well as symbol errors.

More code definitions/relations

- Number of bits/subsymbol is $\tilde{b} = \frac{b}{\bar{N}} = \bar{b} \cdot \tilde{N} = b \cdot \frac{\tilde{N}}{N}$.

- The code's minimum distance remains (for codword spacing):

AWGN $$d_{min}(C_{\boldsymbol{x}}) \triangleq d_{min} = \min_{\boldsymbol{x}_i \neq \boldsymbol{x}_j} \|\boldsymbol{x}_i - \boldsymbol{x}_j\|$$

BSC $$d_{free} = \min_{\boldsymbol{v}_i \neq \boldsymbol{v}_j} d_H(\boldsymbol{v}_i - \boldsymbol{v}_j)$$

Uncoded definition

Definition 2.1.3 (Uncoded and Coded) Uncoded data transmission *has subsymbol constellation C with zero redundancy $\tilde{\rho} = 0$. Necessarily, then uncoded transmission also has $\rho = 0$ and $\bar{\rho} = 0$. Usually in uncoded transmission, the codeword and the subsymbol are trivially the same. If the redundancy is greater than zero, $\tilde{\rho} > 0$, then data transmission is* **coded.**

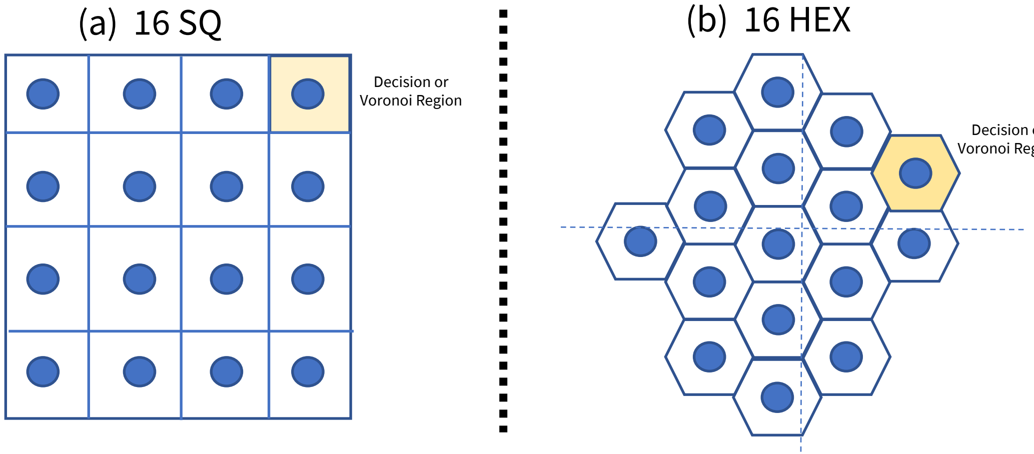
- So QAM and PAM are uncoded when all constellation values are equally likely.
- **SQ QAM**, equivalently PAM, becomes the reference system for coding gain (with same number of bits/subsymbol).

$$\gamma \triangleq \frac{\left(d^2_{\min}(\boldsymbol{x})/\bar{\mathcal{E}}_{\boldsymbol{x}}\right)}{\left(d^2_{\min}(\breve{\boldsymbol{x}})/\bar{\mathcal{E}}_{\breve{\boldsymbol{x}}}\right)} = \underbrace{\frac{\left(\frac{d^2_{\min}(\boldsymbol{x})}{V^{2/N}(\Lambda)}\right)}{\left(\frac{d^2_{\min}(\breve{\boldsymbol{x}})}{V^{2/N}(\breve{\Lambda})}\right)}}_{\substack{\gamma_f \\ \text{fundamental} \\ \text{gain}}} \cdot \underbrace{\frac{\left(\frac{V^{2/N}(\Lambda)}{\bar{\mathcal{E}}_{\boldsymbol{x}}}\right)}{\left(\frac{V^{2/N}(\breve{\Lambda})}{\bar{\mathcal{E}}_{\breve{\boldsymbol{x}}}}\right)}}_{\substack{\gamma_s \\ \text{shaping} \\ \text{gain}}}$$

$$\gamma_f = \frac{d^2_{min}(C_x)}{V^{2/N}(\Lambda)}$$

$$\gamma_s = \frac{V^{2/N}(\Lambda)\cdot\left(2^{2\bar{b}}-1\right)}{\tilde{\mathcal{E}}_x \cdot 6 \cdot \widetilde{N}}$$

Hexagonal constellations, 2D



$$\gamma_s = \frac{V^{2/N}(\Lambda)\cdot\left(2^{2\bar{b}}-1\right)}{\tilde{\mathcal{E}}_x \cdot 6 \cdot \tilde{N}}$$

- Hexagonal lattice has $V(A_2) = 6(\frac{1}{2})(\frac{d}{2})(\frac{d}{\sqrt{3}}) = d^2\frac{\sqrt{3}}{2}$ $\quad \gamma_f = \frac{d^2}{\frac{\sqrt{3}d^2}{2}} = \frac{2}{\sqrt{3}} = .625 \text{ dB}$

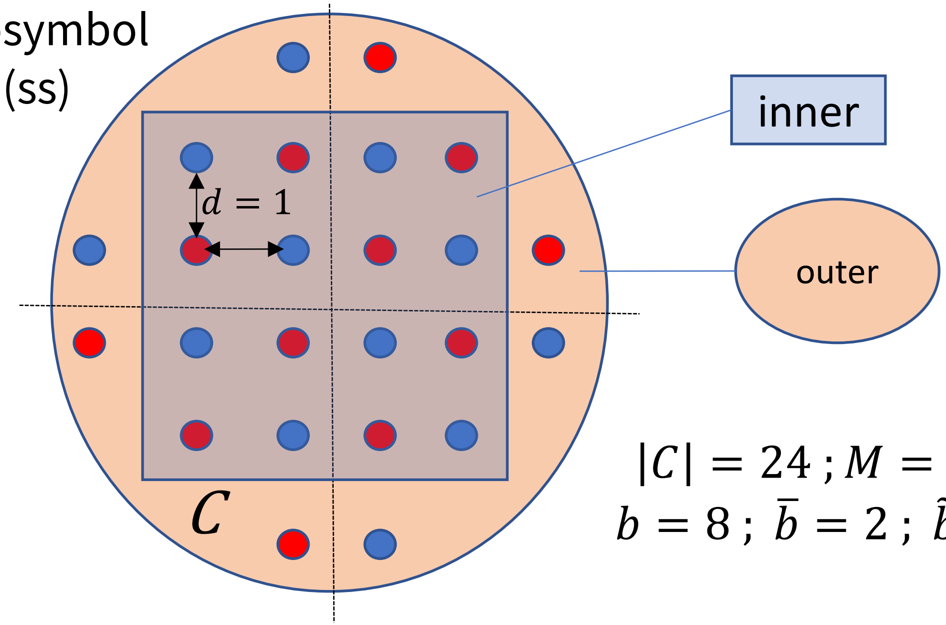
- Overall gain is **+.49 dB** (16HEX/16QAM)
  - Recall Homework problems PS 2.2 and 2.4.

$$\gamma_s = \frac{1\cdot(2^4-1)}{\tilde{\mathcal{E}}_x \cdot 12} = \gamma \text{-} \gamma_f = -.135 \text{ dB}$$

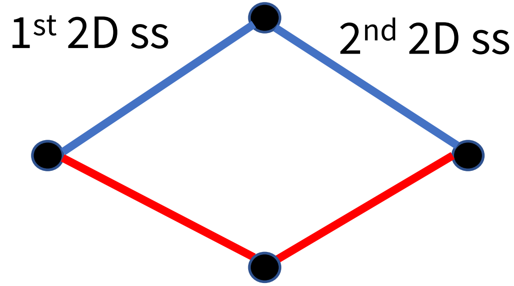
16 points in 2D, even with zero mean Is "lopsided" – 20 would be better - Hex is more ``trit oriented'' or be clever with time-varying constellation design, as in PS2.4.

Simple 4D lattice code example

Subsymbol (ss)



$$|C| = 24\,;M = 256$$
$$b = 8\,;\ \bar{b} = 2\,;\ \tilde{b} = 4$$



Outer can occur only once per codeword.

How would you decode?

- Not all points are equally likely in the 2D subsymbol
- How did we get 256? (8 bits)
  - Blue path (8x8 + 4x8 + 8 x4) = 128
  - Red path similarly 128
  - 128+128 = 256 !

code has $d_{min}^2 = 2$ with 2D ave energy $\tilde{\mathcal{E}}_x = 7$

16 QAM has $d_{min}^2 = 1$ with 2D ave energy $\tilde{\mathcal{E}}_x = 5$

$\gamma = \dfrac{2/7}{1/5}$ = 10/7 = **1.55 dB !** *(forget A_2)*

Random Codes: AWGN's Sphere Packing

Section 2.1

Law of Large Numbers & Random Coding

- Fundamental coding gain can be infinite (which means $P_e \to 0$).

$$\hat{\boldsymbol{x}} \triangleq \frac{1}{N} \cdot \sum_{n=1}^{N} \boldsymbol{x}_n \qquad \hat{f}(\boldsymbol{x}) \triangleq \frac{1}{N} \cdot \sum_{n=1}^{N} f(\boldsymbol{x}_n)$$

Theorem 2.1.1 (Law of Large Numbers (LLN)) *The LLN observes that a stationary random variable z's sample average over its observations $\{z_n\}_{n=1,\ldots,N}$ converges to its mean with large N such that*

$$\lim_{N\to\infty} Pr\left\{\left|\left(\frac{1}{N}\sum_{n=1}^{N} z_n\right) - \mathbb{E}[z]\right| > \epsilon\right\} \to 0 \quad \textbf{weak form} \qquad (2.14)$$

$$\lim_{N\to\infty} Pr\left\{\frac{1}{N}\sum_{n=1}^{N} z_n = \mathbb{E}[z]\right\} = 1 \quad \textbf{strong form}\ . \qquad (2.15)$$

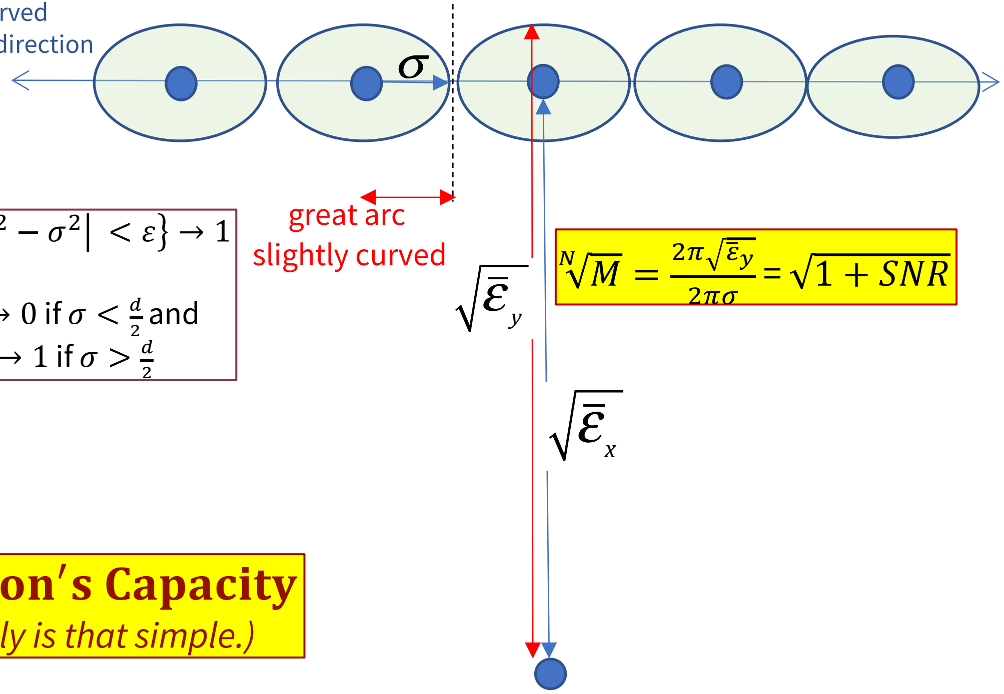
Proof: *See Appendix A.* **QED**.

- Think of $\boldsymbol{x}_n$ here as subsymbol – so randomly pick $\widetilde{N}$ values of $\boldsymbol{x}_n$ from some dist (cont or discrete) to form an N-D codeword.
- Do it again for another codeword, M times.
- That's one code. – Do it again for another code. Average results. This is a **"random-code" design process**.
- Energy via LLN: $f(\boldsymbol{x}) = \|\boldsymbol{x}\|^2$ → all a hypersphere's energy (points) are at its surface (well known in geometry).

Sphere Packing and AWGN Capacity

- For given energy, what is most efficient hypershape? (Think shaping gain.)
  - A Hypersphere!
- Clearly from examples, code design would like to "pack" as many nicely uniformly spaced (for good inter-codeword d_{min}) in a volume as possible.
  - Each codeword has a decision region around it.
- Gaussian noise decision region:
  - LLN implies that noise $\tilde{\boldsymbol{n}}$ must have average variance with prob 1 on shell of its own little hypershere.
- Marginal $\tilde{\boldsymbol{x}}$ distribution is Gaussian, so that is best distn for picking the random codewords.

With Pr→ 1, all symbols are at the surface and along some great arc, where a good code equally spaces them.



$$Pr\left\{\left|\|\hat{\boldsymbol{n}}\|^2 - \sigma^2\right| < \varepsilon\right\} \rightarrow 1$$

So $P_e \rightarrow 0$ if $\sigma < \frac{d}{2}$ and
$P_e \rightarrow 1$ if $\sigma > \frac{d}{2}$

$$\sqrt[N]{M} = \frac{2\pi\sqrt{\bar{\varepsilon}_y}}{2\pi\sigma} = \sqrt{1 + SNR}$$

Shannon's Capacity
(It really is that simple.)

AWGN Capacity

$$\tilde{b} = log_2\left(1 + \frac{SNR}{\Gamma \cdot \gamma_m}\right) \text{ bits/complex-subsymbol } \leq \tilde{\mathcal{C}}$$

- A good code (e.g., one chosen at random) will have $P_e \to 0$ if $\tilde{b} < \tilde{\mathcal{C}}$.
  - Only issue is the very large N.

- And if $\tilde{b} > \tilde{\mathcal{C}}$, even slightly, some decision regions will necessarily have 2 or more codeword points in them, which means "flip a coin" to decide - rapidly the $P_e > 0$. So, it gets bad in a hurry if $\tilde{b} > \tilde{\mathcal{C}}$.

Margin

$$\tilde{b} = log_2\left(1 + \frac{SNR}{\Gamma \cdot \gamma_m}\right) \text{ bits/complex-subsymbol } \leq \mathcal{C}$$

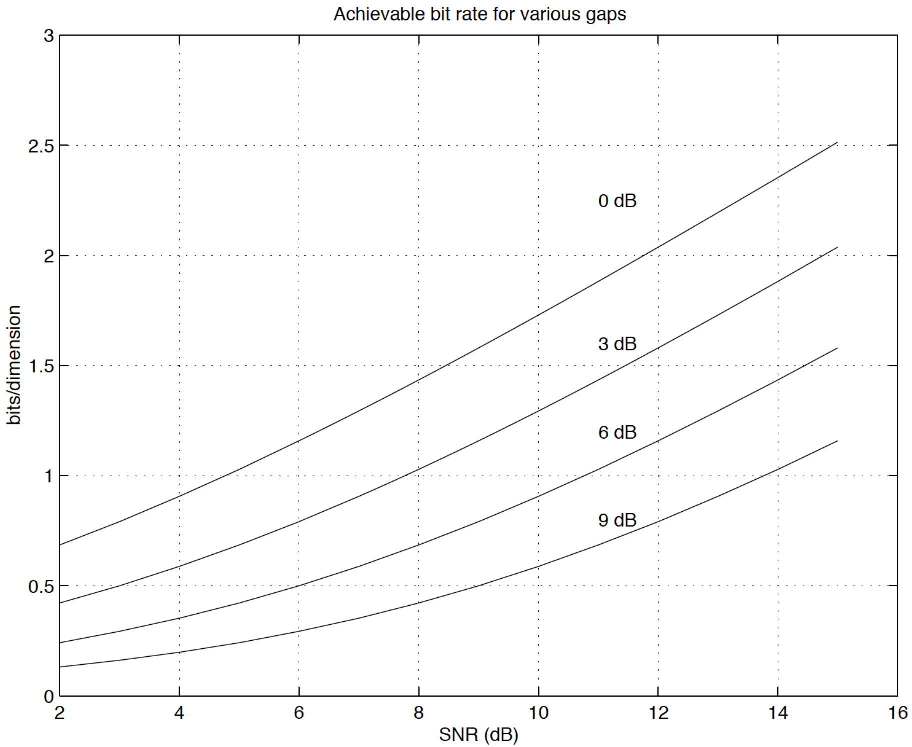
- The designer wants a little “margin” protection against possible noise-power increase.
- **MARGIN** γ_m is this protection (usually in dB), $\gamma_m = \frac{(SNR/\Gamma)}{2^{\tilde{b}}-1}$.

Positive margin – means performing well ; **Negative margin** – means not meeting design goals.

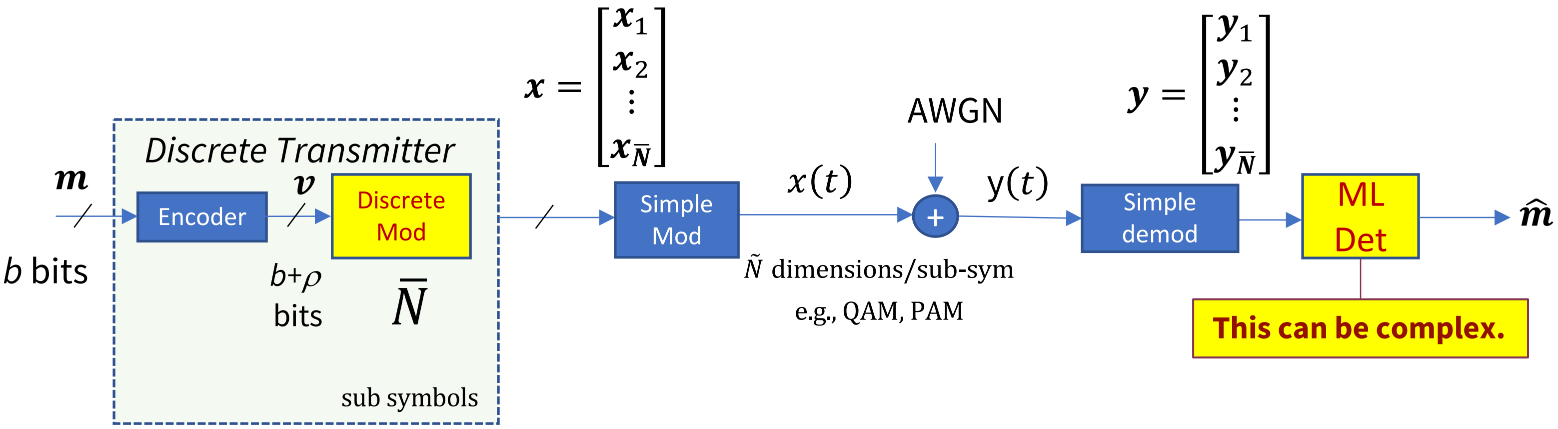
- An AWGN with SNR = 20.5 dB has $\tilde{C} = log_2\,(1 + 10^{2.05})$ = 7 bits/subsymbol.
- Suppose that 16-QAM ($\tilde{b} = 4$) is transmitted @ $P_e = 10^{-6}$ ($\Gamma = 8.8$ dB), then $\gamma_m = \frac{10^{2.05-.88}}{2^4-1} = 0$ dB .
- Suppose instead QAM with $\tilde{b}$ =5 bits/complex-subsymbol with a code and gain 7 dB of gain (Γ→ 8.8-7=1.8 dB)?
  - $\gamma_m = \frac{10^{2.05-.18}}{2^5-1} = 3.8$ dB .
- 6 bits/subsymbol with same code? → 0.8 dB margin – just barely below the desired P_e ; $\bar{P}_e = {}^{P_e}/_{N}$.

Gap Plot & Example

- The **gap Γ is constant**, independent of the bits/dimension $\bar{b}$ – greatly simplifies "loading" (adapting $\bar{b}$ to a specific channel).



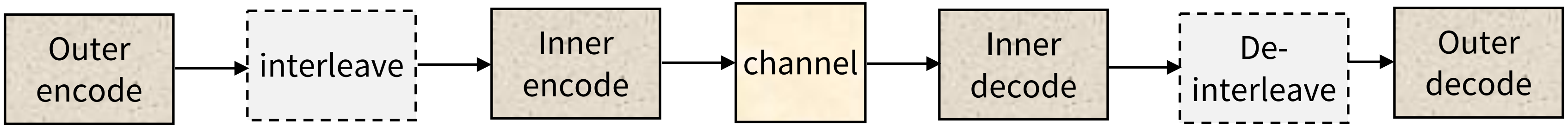
More complete coding illustrated for AWGN



- Simple modulator is QAM (or PAM) typically.
- Demodulator produces $\boldsymbol{y}$, which feeds the ML detector for the code (which applies through encoder).
  - Overall is simple concept.
  - ML detector might have to check large number of codewords.
  - There are many very good codes → having a simple detector becomes the objective.

Can the random part make it complex?

- Put simply, yes, really complex.
- Unless we abandon pure ML decoding …. or do random “educated” guessing.



Iterative decoding

- One way or another, good codes essentially randomize by interleaving (perhaps more than 1 inner code).
- MAP (or approximately so) decoders for each inner code, which then need to help each other.

DMC Codes: Maximum Distance Separable

Section 2.2

Good Ball Packing – Singleton Bound

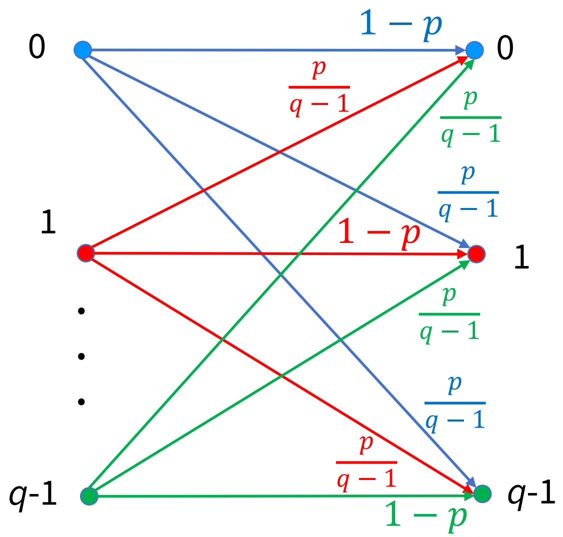
Lemma 2.2.1 (Singleton Bound) *If a designer knows the blocklength n and the d_{free} necessary for performance, then a binary block code's rate must be less than*

$$\begin{aligned} k = \log_2(M) &\leq n - d_{free} + 1 \\ r &\leq 1 - \frac{d_{free} - 1}{n} \quad . \end{aligned}$$

$${}^{2^n}/_{2^k} \sim 2^{d_{free}-1}$$

- Codes that meet the SB are basically most dense ball packing – **Maximum Distance Separable (MDS).**
- For $q > 2$ there are classes of good codes that are **MDS.**
  - For binary unfortunately, these are trivial.
- For instance, linear cyclic codes, Reed Solomon, nonbinary BCH – (L11 and also see EE 387).
- For binary, very good codes often require some degree of randomness (and large n) to even approach SB.
  - See the interleaver on Slide 19 for the random version. Best codes often mix multiple subcodes.

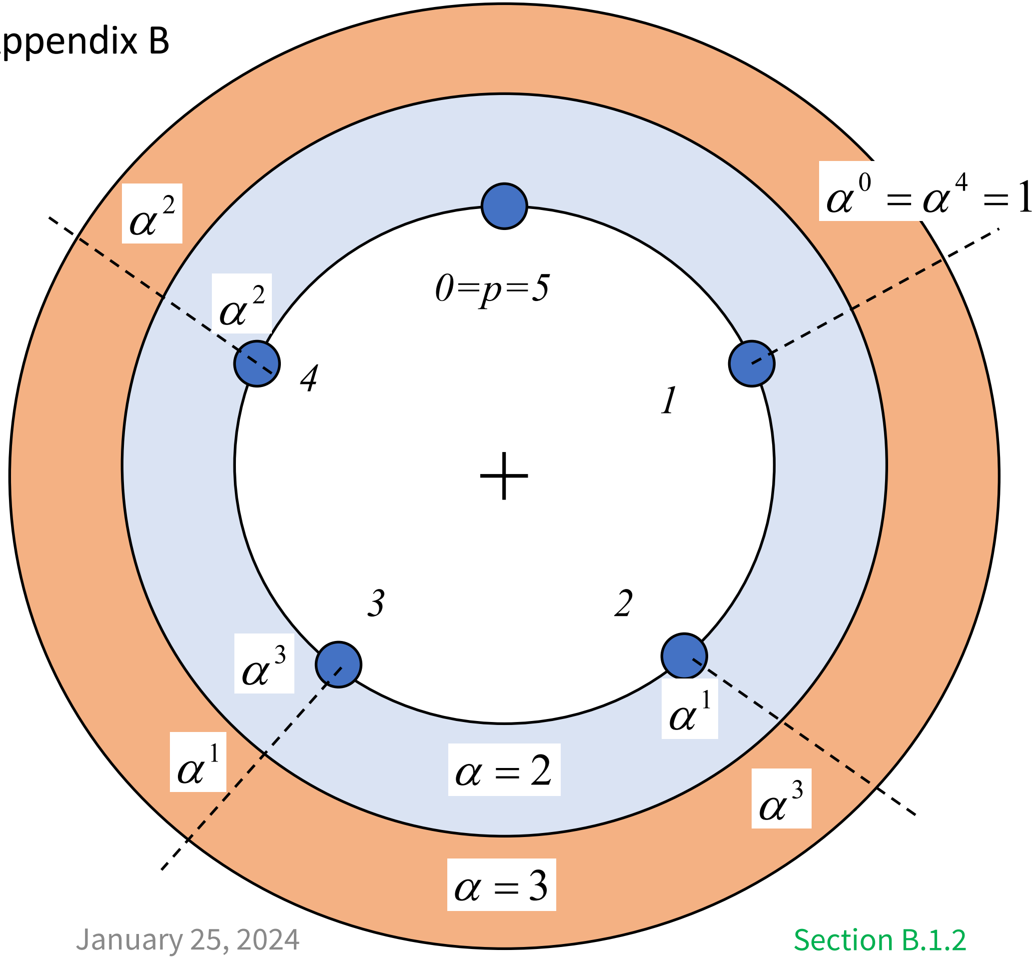
The q-ary Symmetric Channel



- SDMC is typically used with "bytes" (blocks) of inner-channel detected bits.
  - Codes can be much more powerful than best binary codes.
- This channel may have erasures in various modifications also.
- Typically models an "inner channel" for application of outer cyclic codes over finite field (e.g. "Reed Solomon" codes, see EE387, winter 2025, also L11).

Galois Fields: $GF(5) = \{0\ \ 1\ \ \alpha\ \ \alpha^2\ \ \alpha^3\}$

- Appendix B



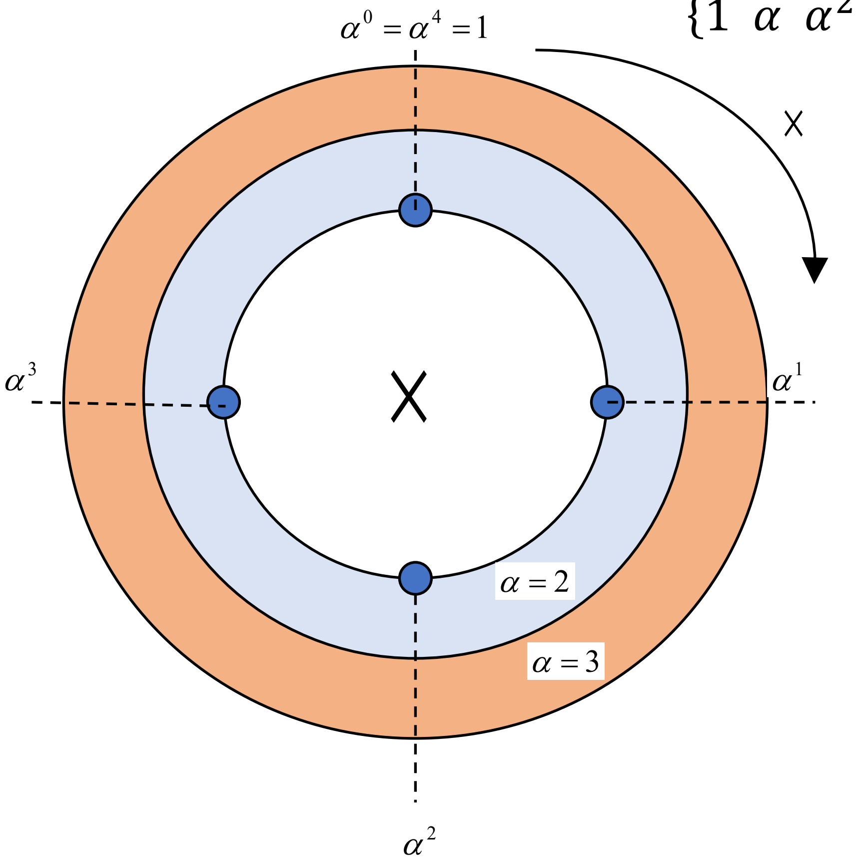
×	1	2	3	4
1	1	2	3	4
2	2	4	1	3
3	3	1	4	2
4	4	3	2	1

α	α^2	α^3	α^4
2	4	3	1
3	4	2	1

Codes for the symmetric DMC

- Finite Field $GF(q)$ has q as prime or product of primes (See Appendix B), and is:
  - closed under addition, with 0 element, &
  - closed under multiplication, with identity, and under division except by 0.
- Codewords are constructed from subsymbol elements of $GF(q)$.
- Same random coding argument leads to uniform over finite field "ball" (consequently uniform in each subsymbol slice) if $q \to \infty$ and $N \to \infty$.

$\{1\ \ \alpha\ \ \alpha^2\ \ \alpha^3\}$



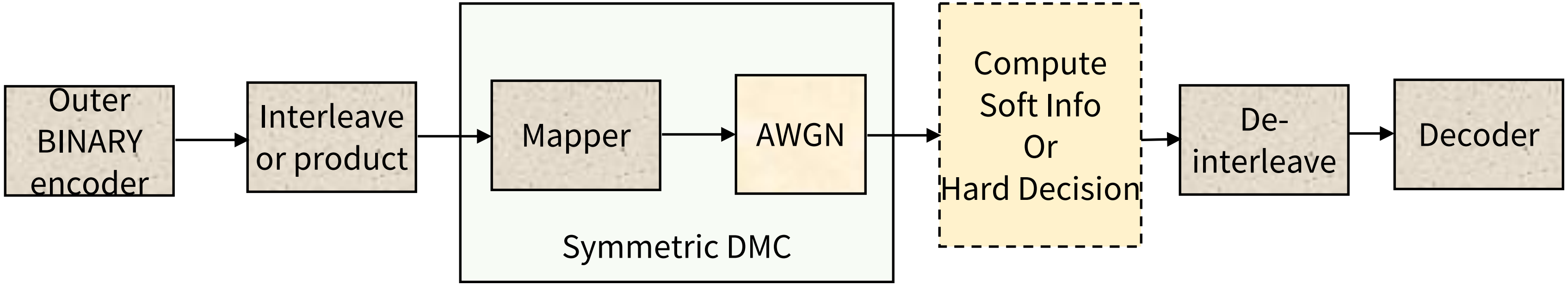
MDS – maximum distance separable

- These are the good "ball packers" for finite-length codewords.
- Cyclic Codes (Reed Solomon, BCH, etc) are See EE387, L11.
- Basically, these codes achieve best ball packing for finite $q = p^n$ and $N \leq q^n - 1$.
- They are cyclic in the finite field (all codewords are circular shifts of one another).
- Their encoders are linear (in $GF(q)$) easy implementation; relatively easy ML decoders.

- Unfortunately, binary MDS are all trivial (like repeat N times – so very low rate or add one parity bit so not very powerful).
  - So nontrivial binary codes are not MDS.
- Really good binary codes will have some "randomness" and long block length, but they exist (Lectures 9-12).
  - Turbo
  - Low-Density Parity Check (LDPC)
  - Polar
  - Product

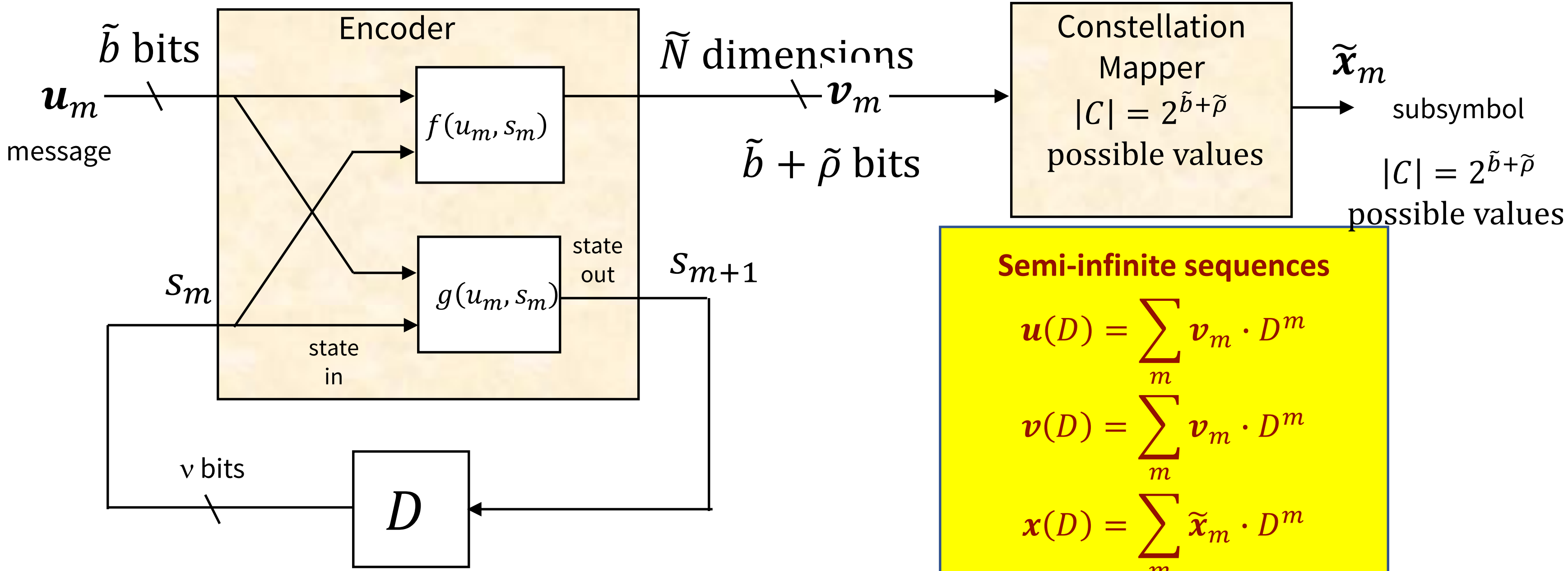
Modern Powerful codes

- γ_f is large, equivalently can be reliably decoded (low Pe).
- Can be based on good long-length binary codes:
  - With binary-to-$|C|$ "mapper" for larger QAM constellations
  - Leave shaping (γ_s) to the constellation boundary design (< 1.53 dB).



Generalization: Sequential Encoder & Mapper

- **Trellis** or **Convolutional** Codes (see feedback below) have model:



Semi-infinite sequences

$$\boldsymbol{u}(D) = \sum_m \boldsymbol{v}_m \cdot D^m$$

$$\boldsymbol{v}(D) = \sum_m \boldsymbol{v}_m \cdot D^m$$

$$\boldsymbol{x}(D) = \sum_m \tilde{\boldsymbol{x}}_m \cdot D^m$$

- Inputs have $\tilde{b}$ – usually bits.
- Outputs are $\tilde{N}$ –dimensional.
  - When $\tilde{\boldsymbol{x}} \in \mathbb{C}^{\tilde{N}}$ → Trellis Code.
  - When $\tilde{\boldsymbol{x}} = \boldsymbol{v} \in GF(2)^{\tilde{N}}$ → Binary convolutional code.

Tries to ``fake''
larger block length
with finite real-time complexity/delay

STANFORD

End Lecture 6