STANFORD

Lecture 8

Multiple Access Channels

April 25, 2024

JOHN M. CIOFFI

Hitachi Professor Emeritus of Engineering

Instructor EE379B – Spring 2024

Announcements & Agenda

- Announcements
  - Problem Set #4 is due Tuesday April 30 (no late, so solutions can distribute).
  - Midterm is 5/2 in class.

- Agenda
  - General Capacity Region **(delayed from L7)**
  - MAC $\mathcal{C}(\boldsymbol{b})$ via partial rate sums
  - Scalar Gaussian MAC
  - Vector Gaussian MAC
    - mu_mac.m software
  - Back-up
    - Capacity Region for frequency-indexed channels

General MU Capacity Region and related optima

Section 2.6.4

3 General Search Steps

- Search 1: Find $\mathcal{I}_{\boldsymbol{min}}$ for given $\boldsymbol{\Pi}$ and $p_{\boldsymbol{xy}}$.
- Search 2: Generate these $\mathcal{I}_{\boldsymbol{min}}$ ‘s convex hull over all orders $\boldsymbol{\Pi}$ for the achievable region $\mathcal{A}(\boldsymbol{b}\,, p_{\boldsymbol{xy}})$.
- Search 3: Generate a 2nd Convex hull over all probability distributions $p_{\boldsymbol{x}}$ for $\mathcal{C}(\boldsymbol{b}\,)$.
- These searches can be complex for general case, but do simplify for Gaussian MAC, BC, and IC.

Order-and-Distribution-Dependent Region

- **Order Step** forms a first convex hull of all $\mathcal{I}_{min}$ vectors FOR EACH GIVEN ORDER and input distribution.

$$\mathcal{A}(\boldsymbol{b}\,,p_{xy}) = \bigcup_{\boldsymbol{\Pi}}^{conv} \mathcal{I}_{min}(\boldsymbol{\Pi}, p_{xy})$$

Achievable Region

- Any point outside $\mathcal{A}(\boldsymbol{b}, p_{\boldsymbol{x}})$ will, in the chain-rule sense, have large error probability for at least one receiver.
  - The orders are "dimension shared" across different designs (the convex hull / union) operation …. sub users.
  - Every order and all convex combinations thereof have been considered, so it it could have been decoded it was inside $\mathcal{A}(\boldsymbol{b}, p_{\boldsymbol{x}})$.

- **Distribution Step** forms hull over the allowed input distributions (a 2nd convex hull operation).

$$\mathcal{C}(\boldsymbol{b}\,) = \bigcup_{p_{\boldsymbol{x}}}^{conv} \mathcal{A}(\boldsymbol{b}\,,p_{\boldsymbol{xy}})$$

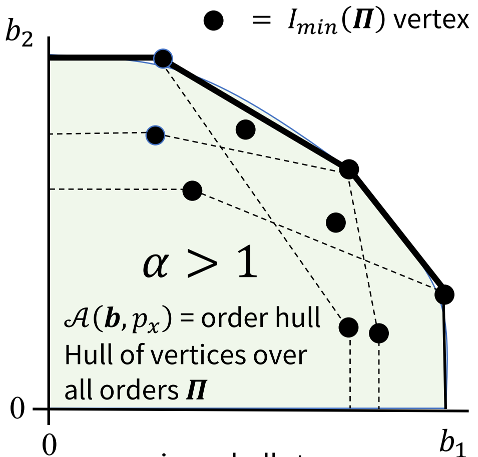
MU Capacity Region

many cases simplify

- The order search is "NP-hard."
- The distribution search can also be "NP-hard."
- **Admissibility:** Is $\boldsymbol{b} \in \mathcal{C}(\boldsymbol{b})$? (often easier fortunately)

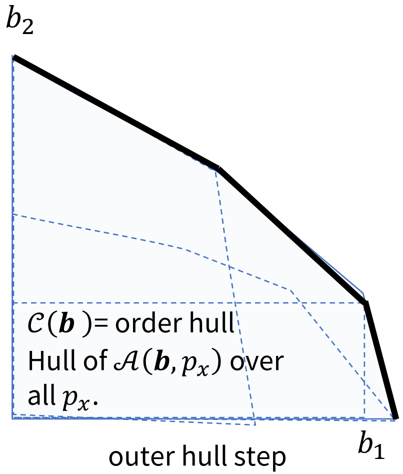
The two convex-hull steps

- The **order-vertices'** hull



inner hull step

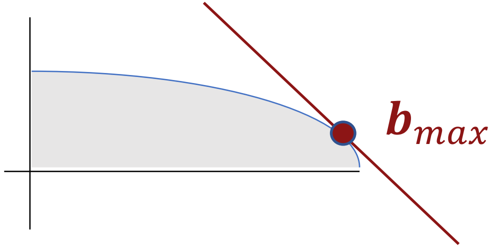
- The **input-distributions'** hull

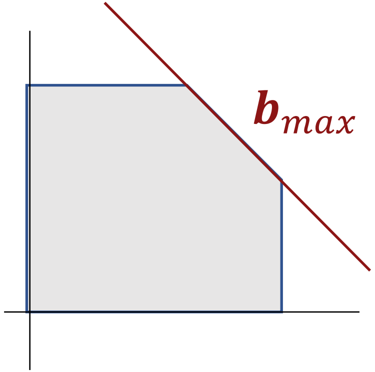


outer hull step

Maximum Rate Sum

- The **rate sum** is $\mathbf{1}^*\boldsymbol{b}$, or simply the sum of the user bits/symbol.
- This is a hyperplane in U-space.
- This plane with normal vector $\mathbf{1}$ will be tangent to $\mathcal{C}(\boldsymbol{b})$ at $\boldsymbol{b}_{max}$, where $\mathbf{1}^*\boldsymbol{b}_{max} = b_{max}$, the maximum sum rate.





MU Matrix AWGN Channels

- $\mathcal{C}(\boldsymbol{b})$ for a multi-user AWGN channel $\boldsymbol{y} = H \cdot \boldsymbol{x} + \boldsymbol{n}$ will have all users' input distributions as Gaussian at the region's (non-zero) boundary, $\mathscr{C}(\boldsymbol{b})$.
  - Each of these points is a mutual information that for each receiver/user $b_u = \mathbb{I}$ has a chain-rule decomposition.
  - For any subset of output dimensions $\boldsymbol{y}$ and any subset of inputs $\boldsymbol{x_u}$, $\mathbb{I}(\boldsymbol{x}; \boldsymbol{y}) = \mathbb{I}\left(\boldsymbol{x_u}; \boldsymbol{y} / \boldsymbol{x}_{\boldsymbol{U} \setminus \boldsymbol{u}}\right) + \mathbb{I}\left(\boldsymbol{x}_{\boldsymbol{U} \setminus \boldsymbol{u}}; \boldsymbol{y}\right)$.
    - With independent input messages, these are separable and can be separately maximized.
    - The second term is a "single-user," $\boldsymbol{U} \setminus \boldsymbol{u}$, channel, and this channel thus has optimum Gaussian input.
    - The uncancelled users' crosstalk may contribute in MMSE sense to noise, which then is sum of Gaussians that is also Gaussian.
  - (Proof by induction: last user is single-user channel, which has Gaussian; then next to last has Gaussian xtalk and noise, so it also is Gaussian …), the optimum $\boldsymbol{u}$ is also Gaussian. This also works for any user subset $\boldsymbol{u}$. **QED.**

In general, with user components, treat $U \to U'$.

Degraded-Matrix AWGN

Definition 2.6.7 [(Subsymbol) Degraded multiuser Gaussian Channel] *A* **(subsymbol)-degraded AWGN multiuser channel** *has matrix ranks for H and/or $R_{\boldsymbol{xx}}$ that are ϱ_H and $\varrho_{R_{\boldsymbol{xx}}}$ respectively, such that*

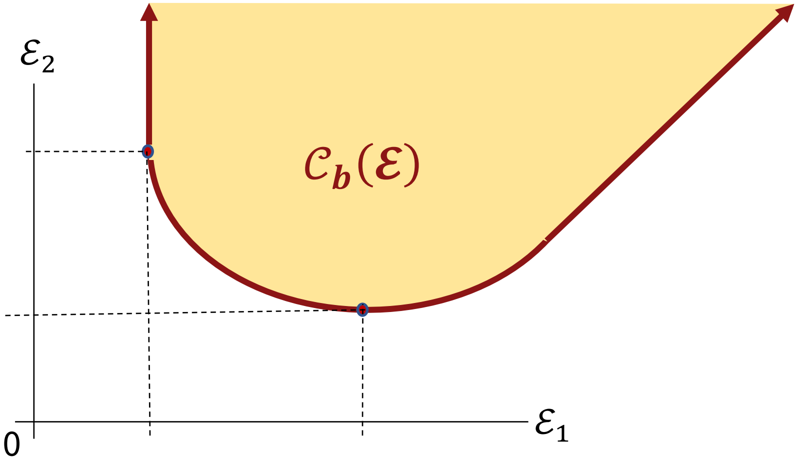
$$\min\left\{\varrho_{R_{\boldsymbol{xx}}}, \varrho_H\right\} < U \quad . \tag{2.284}$$

Otherwise, the channel is **non-degraded**. *The literature often omits the word "subsymbol," but it is tacit in degraded-channel definitions.*

This degraded definition depends on channel AND input.

- What "degraded" means physically is that there are not enough dimensions to carry all users independently.
  - There are other chain-rule conditional-probability definitions, but they appear equivalent.
- If all users energize, some must co-exist on the available (subsymbol) dimensions.
  - A name is NOMA (new name for old subject) – Non-Orthogonal Multiple Access (associated with IoT where U can be very large).
- Non-degraded channels (Massive MIMO is an example) have a surplus of dimensions (less likely to be degraded).
- $R_{\boldsymbol{nn}}$ is never singular on real channels, so noise whitening should not reduce the rank.
  - however, we will see a special case where design will assume a fictitious singular noise, so we'll need care on this when used.

Capacity-Energy Region (AWGN only)



- Essentially redraws the capacity regions for different energy vectors with fixed $\boldsymbol{b}$.
  - Trivially, any point within is reliably achievable, while points outside have insufficient energy.

- If a given $\boldsymbol{\mathcal{E}}_x \in \mathcal{C}_{\boldsymbol{b}}(\boldsymbol{\mathcal{E}})$, then $\boldsymbol{b}$ is **admissible** when also $\boldsymbol{b}_{\boldsymbol{\mathcal{E}}_x} \in \mathcal{C}(\boldsymbol{b})$.

Ergodic Capacity Region

- Design averages the capacity region over the variable-channel's parameter (joint if multiparameters) distribution.
  - This assumes messages are independent of parameters.

- Example: The **ergodic capacity region** is $\langle \mathcal{C}(\boldsymbol{b}) \rangle = \mathbb{E}_{\boldsymbol{H}}[\mathcal{C}(\boldsymbol{b})]$ for the matrix AWGN:

  - *interesting result* – The distribution $p_{\boldsymbol{x}}$ that maximizes the ergodic capacity when H is Raleigh (any user) fading is a discrete distribution (so then not Gaussian); extends well-known result for single user.

  - The AEP results don't hold because they assume the INPUT distribution is ergodic – and that is not necessarily true if the channel is varying (the reversal of input/channel limits for large blocklength may not hold and Rayleigh is example).

  - This presumably extends to multiuser case; however most channel variation for wideband (e.g. modern wireless) have codeword lengths/delays for good codes that are less than the coherence time, so Gaussian good codes remain in wide use. Thus, might as well go with Gaussian/known-good-codes for "quasi-stationary" assumption.

- **Outage Capacity Region?**
  - There is some work on "zero-outage" capacity region (depending on definition may not be same as $\langle \mathcal{C}(\boldsymbol{b}) \rangle$).

  - Not necessarily just $(1 - P_{out}) \cdot \langle \mathcal{C}(\boldsymbol{b}) \rangle$, like single-user case because of "which user outage?" question, although it probably is a decent measure anyway.

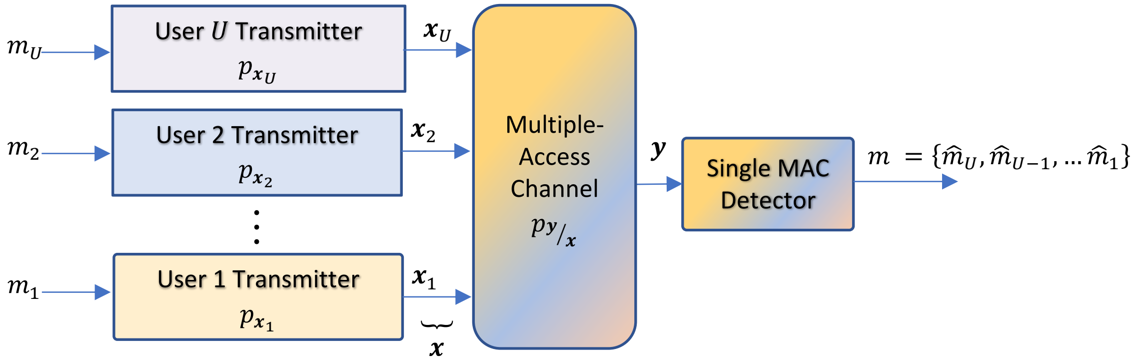
  - It is probably more important to look at user input-rate variation (and contention for which point in $\mathcal{C}(\boldsymbol{b})$) and layer 2/3 buffer overflow outages, etc. (see back up slides for L7)

MAC $C(\boldsymbol{b})$ via partial rate sums

PS4.3 - 2.23

The MAC's partial rate sums

$p_{\boldsymbol{x}} = \prod_{u=1}^{U} p_{\boldsymbol{x}_u}$ independent user inputs



- User u has maximum bit rate, when all other users are given (cancelled):

$$b_u \leq \mathbb{I}\left(\boldsymbol{x}_u; \boldsymbol{y} / \boldsymbol{x}_{U \setminus u}\right)$$

- The single receiver can process any user subset $\boldsymbol{u} \subseteq \boldsymbol{U}$.
  - This has a single-macro-user interpretation with summed bits/subsymbol:
    - $b_{\boldsymbol{u}} = \sum_{u \in \boldsymbol{u}} b_u \leq \mathbb{I}\left(\boldsymbol{x}_{\boldsymbol{u}}; \boldsymbol{y} / \boldsymbol{x}_{U \setminus \boldsymbol{u}}\right)$.
  - This defines a hyperplane with $|\boldsymbol{u}| - 1$ dimensions $\left(\in \mathbb{R}^{|\boldsymbol{u}|}\right)$.
- MAC order simplifies (receiver) to $\boldsymbol{\Pi} = \boldsymbol{\pi}_1$.
  - The user order within $\boldsymbol{u}$ does not change the sum $\mathbb{I}\left(\boldsymbol{x}_{\boldsymbol{u}}; \boldsymbol{y} / \boldsymbol{x}_{U \setminus \boldsymbol{u}}\right)$, nor does the order within $\boldsymbol{U} \setminus \boldsymbol{u}$.
  - The number of planes (lines … hyperplanes) to search decreases substantially to $2^U - 1$ (null set excluded) << $(U!)^U$ (large U).

Chain-Rule Reminder Lemma 2.3.4

$$\mathbb{I}(\boldsymbol{x};\boldsymbol{y}) = \mathbb{I}\left(\boldsymbol{x}_{\boldsymbol{u}};\boldsymbol{y}\,/\boldsymbol{x}_{\boldsymbol{U}\setminus\boldsymbol{u}}\right)+\mathbb{I}\left(\boldsymbol{x}_{\boldsymbol{U}\setminus\boldsymbol{u}}\,;\boldsymbol{y}\,\right)$$

User (set) $\boldsymbol{u}$ is detected with all other users $\boldsymbol{x}_{\boldsymbol{U}\setminus\boldsymbol{u}}$ given (cancelled).

Other-user (set) $\boldsymbol{U}\setminus\boldsymbol{u}$ is detected with users $\boldsymbol{x}_{\boldsymbol{u}}$ as noise.

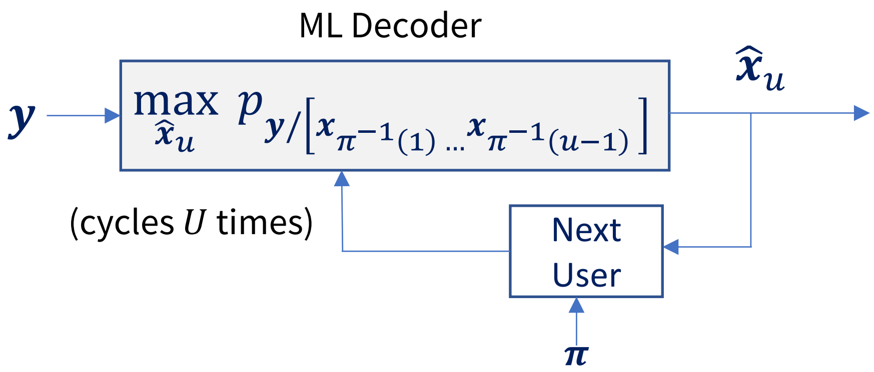
2^{U} possible choices of $\boldsymbol{u}$

- $b \le \mathbb{I}(\boldsymbol{x};\boldsymbol{y})$ - This rate sum corresponds to the choice $\boldsymbol{u}$= $\boldsymbol{U}$.
- A (hyperplane) **face**: $b_1 + b_2 + \cdots + b_{|\boldsymbol{u}|} \le \mathbb{I}\left(\boldsymbol{x}_{\boldsymbol{u}};\boldsymbol{y}\,/\boldsymbol{x}_{\boldsymbol{U}\setminus\boldsymbol{u}}\right)$ - defines $(2^{|\boldsymbol{u}|}-1)$ partial rate sums.
  - There are also U trivial faces for positive bits/subsymbol $b_u \ge 0$, so really 2^{u} -1+U faces that bound $\mathcal{A}(\boldsymbol{b}, p_{\boldsymbol{x}})$.
- A **vertex** corresponds to a specific $\boldsymbol{b} = \mathbb{I}$ for a specific order $\boldsymbol{\pi}$; *examples include for $U = 2$:*

$$\begin{bmatrix}\mathbb{I}(\boldsymbol{x}_2;\boldsymbol{y}\,/\boldsymbol{x}_1)\\ \mathbb{I}(\boldsymbol{x}_1;\boldsymbol{y}\,)\end{bmatrix}\quad\begin{bmatrix}\mathbb{I}(\boldsymbol{x}_1;\boldsymbol{y}\,/\boldsymbol{x}_2)\\ \mathbb{I}(\boldsymbol{x}_2;\boldsymbol{y})\end{bmatrix}.$$

In general, $\exists\ U!$ vertices for a specific $p_{\boldsymbol{x}}$.

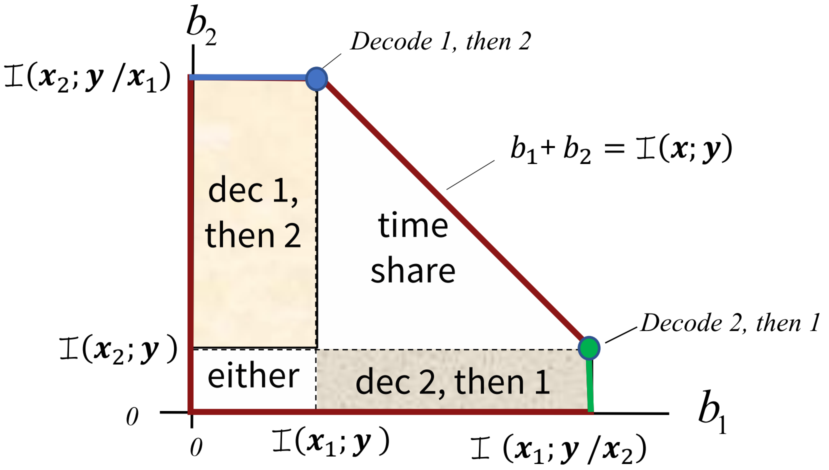
Chain-Rule Decoder



Successive Decoding or …
Generalized Decision Feedback Eq
(or "NOMA")

- For the given order, decode all the lower-indexed users first and then current user.
- Since there is only one order, relabel users and avoid all the $\pi^{-1}(\cdot)$ notation.
- There is no loss of generality.

A 2-user MAC rate region

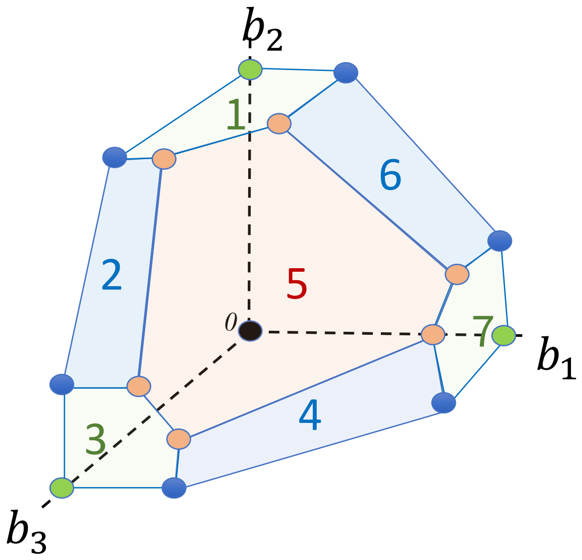


Specific to a p_{xy}

- Pentagon – 5 vertices and 5 faces
  - $2^U - 1 + U$ Faces are the $\mathbb{I}(x_u ; y / x_{U \setminus u})$ & $b_u \geq 0$
  - $U! = 2$ vertices are the both-user order points π
    - 2 more are single-user points, one for each user
    - 1 more is the origin
    - 5 total

- b_2 vertex (short blue line) decodes 1 first (given), then 2 as if 1 is "cancelled."
  - Similar statement holds for b_1 vertex (and short green) line.
- Line with slope -1 is **time-share or really vertex-share**; it also is constant maximum rate sum (for this p_{xy}).
  - There are two codes for each user (4 codes); This is example of user components (or subusers, sometimes called "**rate splitting**")

A 3-user rate region



- Decahedron – 10 faces
  - $2^U - 1 + U$ Faces are the $\mathbb{I}(\boldsymbol{x_u} \,; \boldsymbol{y} \,/\boldsymbol{x_{U\setminus u}})$
  - $U! = 6$ vertices (rose) are the 3-user order points $\boldsymbol{\pi}$

(rose)	= vertex for 3 users $U!$ $(= 6)$
(blue)	= vertex for 2 users $U!$ $(= 6)$
(green)	= vertex for 1 user U $(= 3)$
(black)	= vertex for 0 users $(= 1)$

$\sum_{k=0}^{U} \binom{U}{k} \cdot k! =$16 vertices

- b_2 horizontal plane (pentagon) decodes 1 and 3 first (given), then 2 as if 1 and/or 3 are "cancelled."
  - 1 and 3 form a two-user horizontal pentagon region.
  - Similar statements hold for b_1 vertical-plane pentagon and b_3 facial-plane pentagon.
- Rose plane normal to $\mathbf{1} = [1 \quad 1 \quad 1]^*$ is dimension-share of rose vertices; it has constant maximum rate sum (for this $p_{\boldsymbol{xy}}$).
  - There could be as many as 3 codes/components for each user on a time-share of vertices.
- The blue and green planes may also dimension-share vertices.
- $\mathcal{A}(\boldsymbol{b}, p_{\boldsymbol{x}})$ is the entire interior plus faces and vertices. Any point outside violates at least one single-user mutual-information bound.

MAC Capacity Region

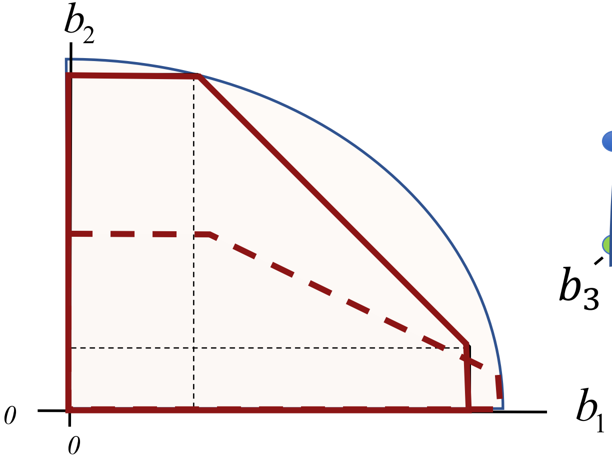
- More formally, the MAC's achievable region is bounded by hyperplanar regions

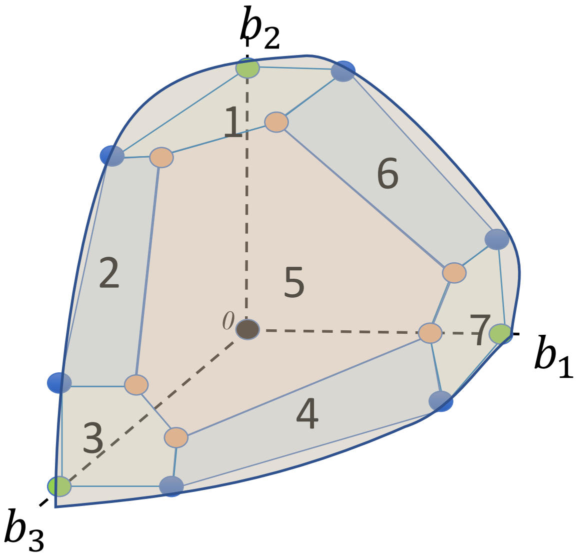
$$\mathcal{A}(\boldsymbol{b}, p_{\boldsymbol{x}}) = \bigcap_{\boldsymbol{u} \subseteq \boldsymbol{U}} \left\{ \boldsymbol{b} \,\middle|\, 0 \leq \sum_{i \in \{\boldsymbol{u}\}} b_i \leq \mathcal{I}\left(\boldsymbol{x}_i ; \boldsymbol{y} / \boldsymbol{x}_{\boldsymbol{u} \setminus i}\right) \right\} .$$

- The vertices are where hyperplanes intersect at a point.
  - Or, lines (smaller dimensional hyperplanes) may also bound.

- Convex hull over all multi-user input probability distributions $p_{\boldsymbol{x}}$ is

$$\mathcal{C}_{MAC}(\boldsymbol{b}) = \bigcup_{\boldsymbol{u} \subseteq p_{\boldsymbol{x}}}^{conv} \mathcal{A}(\boldsymbol{b}, p_{\boldsymbol{x}}).$$



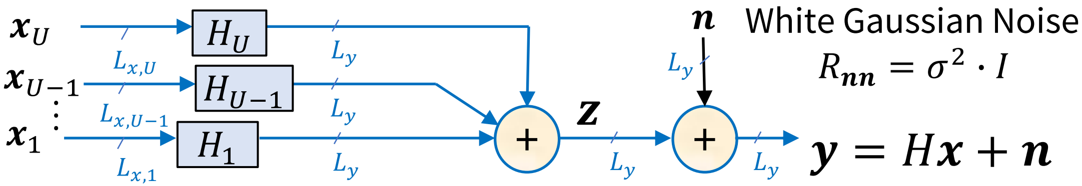


Scalar Gaussian MAC

PS4.3 - 2.25 Time-Division Multiplexing region

Section 2.7.2

General Gaussian MAC



$$y = [H_U \quad H_{U-1} \quad \cdots \quad H_1] \cdot \begin{bmatrix} x_U \\ x_{U-1} \\ \vdots \\ x_1 \end{bmatrix} + n$$

$L_y \times 1$ — H: $L_y \times \mathfrak{L}_x$ — x: $\mathfrak{L}_x \times 1$

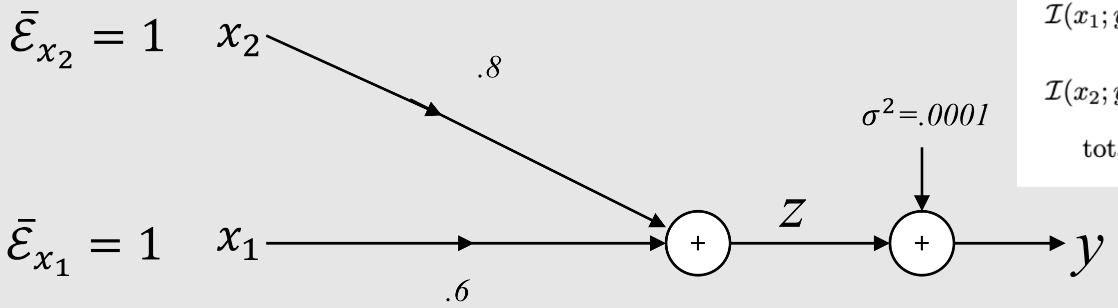
More generally, variable-dim inputs have

$$\mathfrak{L}_x = \sum_{u=1}^{U} L_{x,u} \sim U \cdot L_x$$

- Inputs are independent.
  - R_{xx} is block diagonal.
  - Only 1 output and 1 noise.
- One Receiver will estimate all inputs.
  - It can do so in any order.
  - "Given an input" x_u means cancel it from y.
  - This does not necessary mean subtract $H_u \cdot x_u$ from y
    - Unless $L_y = L_{x,u} = 1$; or H_u is diagonal and noise is white.

ϱ_H is the matrix H's **rank:**
= number of linearly independent rows (or columns)
= # of non-zero singular values.

Example



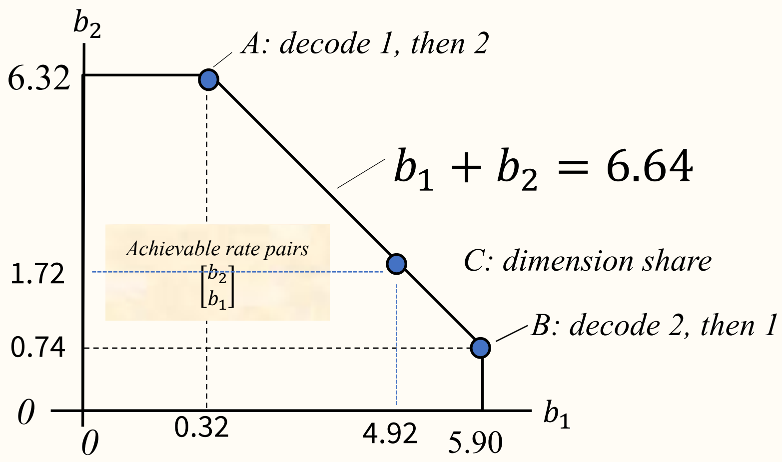
$$\mathcal{I}(x_1; y) = \frac{1}{2}\log_2\left(1 + \frac{.36 \cdot 1}{.0001 + .64}\right) = .32 \text{ bits/dimension}$$

$$\mathcal{I}(x_2; y) = \frac{1}{2}\log_2\left(1 + \frac{.64 \cdot 1}{.0001 + .36}\right) = .74 \text{ bits/dimension}$$

$$\text{total} = 1.06 \text{ bits/dimension} \quad ,$$

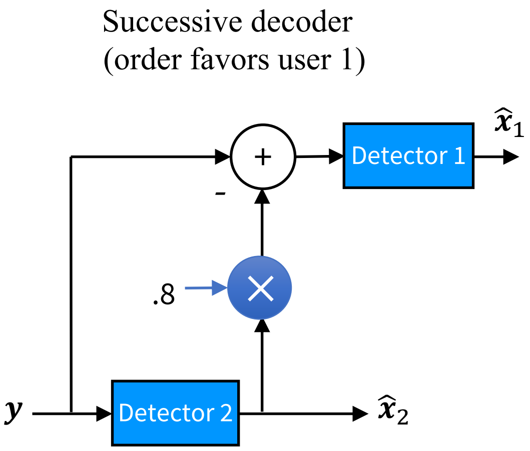
$$\mathcal{I}(x_2; y/x_1) = \frac{1}{2}\log_2\left(1 + \text{SNR}_2\right) = \frac{1}{2}\log_2\left(1 + \frac{.64 \cdot 1}{.0001}\right) = 6.32 \text{ bits/dimensio}$$

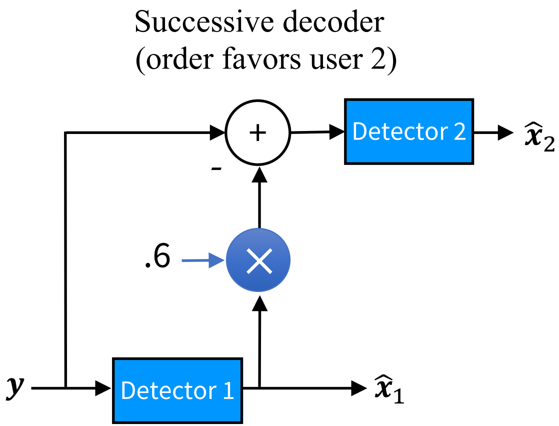
$$\mathcal{I}(x_1; y/x_2) = \frac{1}{2}\log_2\left(1 + \text{SNR}_1\right) = \frac{1}{2}\log_2\left(1 + \frac{.36 \cdot 1}{.0001}\right) = 5.90 \text{ bits/dimension}$$



- Point C is ¼ share B and ¾ share A.

Successive decoding for scalar example





- Only 2 orders are possible for 2 users.
- $\exists$ $U!$ in general (corresponding to each possible order).
- The last user is “favored” in decoding (first accepts other as noise).

2 – User Scalar $L_x = L_y = 1$

General formula
Scalar MAC

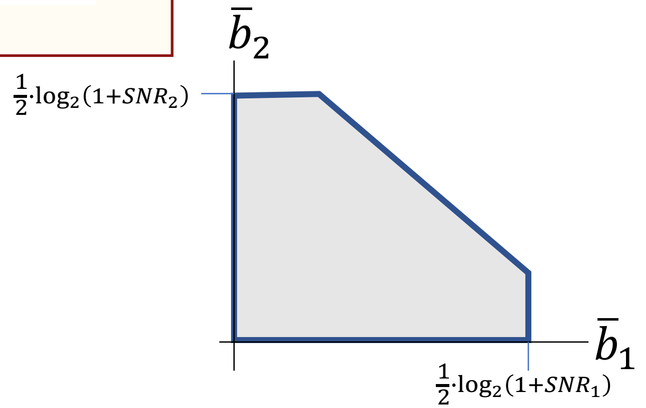
$$\bar{I}(x_{\boldsymbol{u}}; y / x_{\boldsymbol{U} \setminus \boldsymbol{u}}) = \frac{1}{2} \log_2 \left(1 + \frac{\sum_{i \in \boldsymbol{u}} \bar{\mathcal{E}}_i \cdot |H_i|^2}{\sigma^2} \right)$$

2 users

$$\begin{aligned} \mathrm{SNR}_1 &= \frac{\mathcal{E}_1 \cdot |h_1|^2}{\sigma^2} \\ \mathrm{SNR}_2 &= \frac{\mathcal{E}_2 \cdot |h_2|^2}{\sigma^2} \\ \mathrm{SNR} &= \frac{\mathcal{E}_1 \cdot |h_1|^2 + \mathcal{E}_2 \cdot |h_2|^2}{\sigma^2} \end{aligned}$$

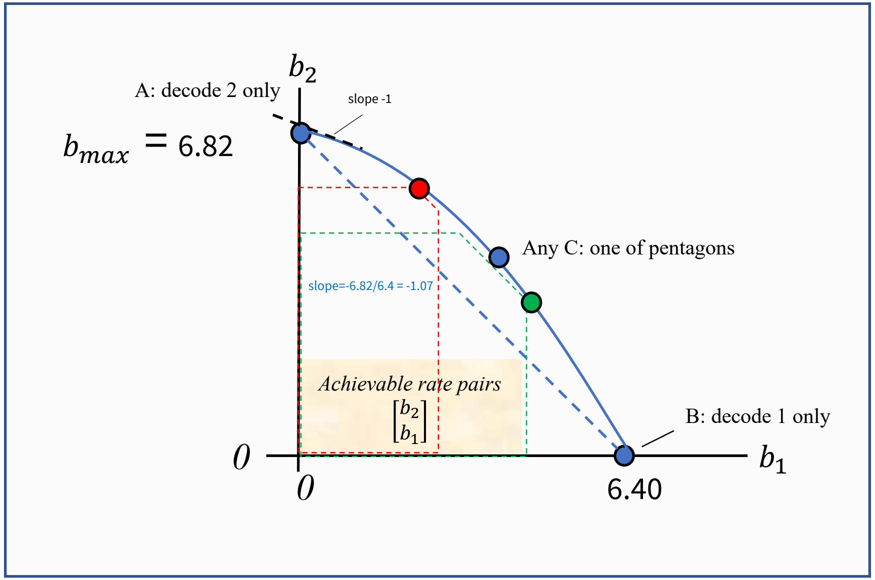
$$\begin{aligned} \bar{b}_1 &\leq \frac{1}{2} \log_2 (1 + \mathrm{SNR}_1) \\ \bar{b}_2 &\leq \frac{1}{2} \log_2 (1 + \mathrm{SNR}_2) \\ \bar{b}_1 + \bar{b}_2 &\leq \frac{1}{2} \log_2 (1 + \mathrm{SNR}) \end{aligned}$$

- 3 planes (lines) ~ 3 SNR's
- 1 sum rate
- Nonzero individual rates

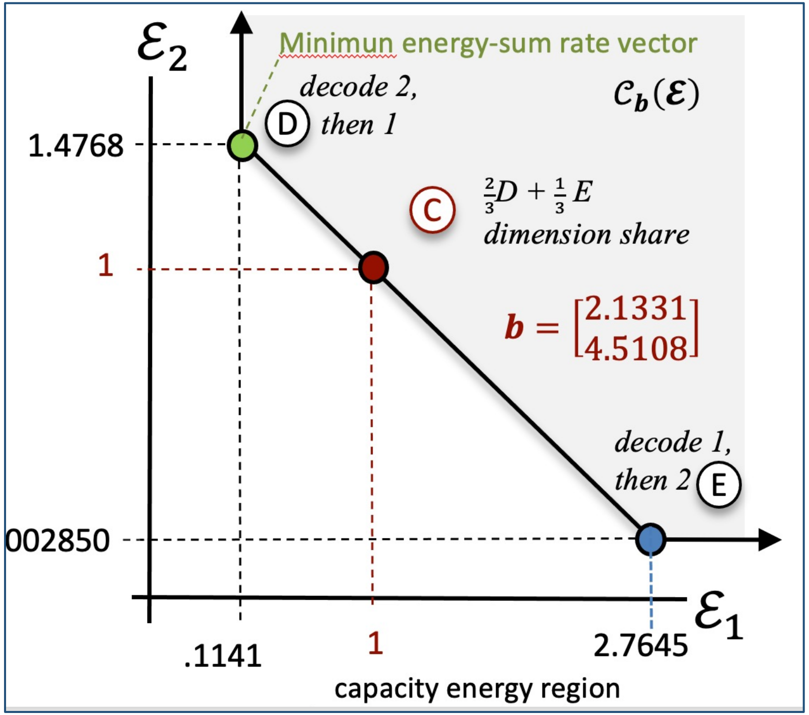


Energy-Sum MAC

- Single energy constraint $\mathcal{E}_1 + \mathcal{E}_2 \leq \mathcal{E}_x$ (instead of 2 constraints)
- Capacity region becomes union of pentagons (and 1 triangle),
  - one for each combination of energies that add to total.



- Or view Energy-Capacity Region
  - one for each bit vector $\boldsymbol{b}$



Time-Sharing Conundrum (Coding Theorist's Fallacy in disguise)

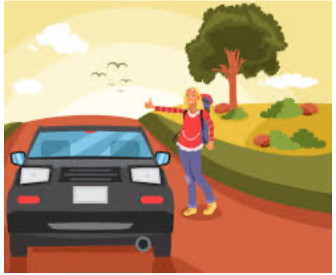
- What is meaning of time-sharing? ("convex hull")
  - The different codes correspond to user components, each used for its respective fraction of "time" (dimensions).
- With time-sharing, what does $\mathcal{E}_u$ mean?
  - Energy constant at $\mathcal{E}_u$: Is this then for every symbol/subsymbol in the sharing?
  - Or the average over the "time-shared" subsymbols?
- The second instance of averaging often enlarges the capacity region.
- So, "time-sharing" is somewhat ill-defined.
  - Despite most info/com texts on MAC using it.
- Lecture 4's Separation Theorem actually allows different mutual information $\mathcal{I}_A$ and $\mathcal{I}_B$ to be represented by their average information – *for the same user.*
  - $\mathcal{I} = \alpha \cdot \mathcal{I}_A + (1-\alpha) \cdot \mathcal{I}_B$.
  - ST uses same constellation with average $\boldsymbol{b}$ for each symbol, possible very large$|C|$).
  - If the shared same-user codes correspond to vertices with different orders, this creates issues for Separation Thm application.
    - But it is still possible, although the successive decoding needs to become "iterative-user" successive decoding.
    - Of course, each user can use subusers; each user has subcode for A and for B, but then constellation varies.

Primary and Secondary Components (E-sum MAC)

Primary-user component: has nonzero energy for E-sum MAC's maximum rate.

Secondary-user component: has zero energy for E-sum MAC's maximum rate.

- Primary components dominate with largest pass-space gains (dimensions used for component).
- Secondary users "free load" on these primary-component dimensions.



Previous example (.8 and .6):

The pass-space is just one dimension ($L_y = 1$).

user 2 is all primary (.8) ; user 1 is all secondary (.6).

max sum is 6.82 (all energy on user 2).

Rate-sum decreases if secondary user components energize (see slide L8:15).

How Many Primary Components (E-sum MAC)?

- The MAC has no more than $U^o \leq \wp_H$ primary components, to find them first do U SVD's:

$$\tilde{H}_u = R_{noise}^{-1/2}(u) \cdot H_u = F_u \cdot \Lambda_u \cdot M_u^* \quad \text{with} \quad |\tilde{H}_u| \triangleq \prod_{l=1}^{\wp_{H_u}} \lambda_{u,l} > 0\,.$$

- Each user can excite up to $\wp_{H_u}$ possible independent dimensions per subsymbol.
  - The $R_{noise}(u)$ includes all other user components' crosstalk for whatever energies they use (knows all $R_{\boldsymbol{xx}}(u)$'s).
  - Each user can have vector-coding modulator without loss, or some linear combination of the pass-space dimensions.

- For the channel gains in the VC,

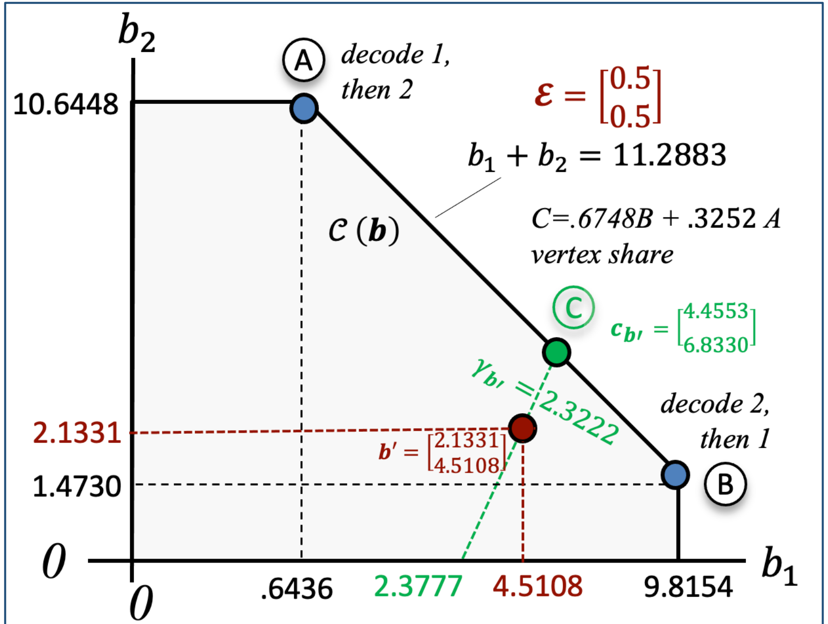
$$g_u = |\tilde{H}_u|^2 = \prod_{l=1}^{\wp_{H_u}} \lambda_{u,l} \quad .$$

- The **primary-user components** correspond to those energized in achieving max rate sum on the E-sum MAC. All others are **secondary-user components**.

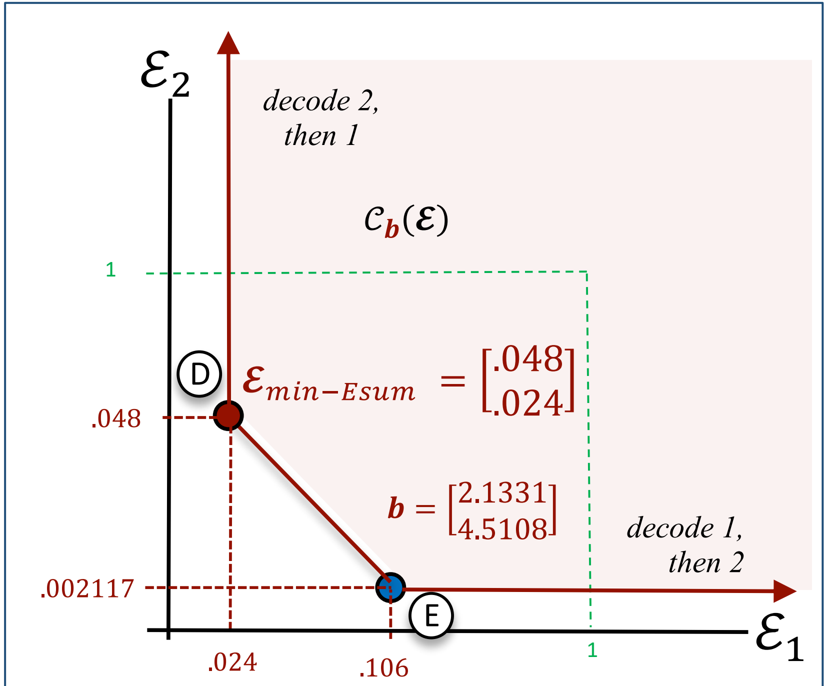
- The "components" idea is helpful when individual users' transmitters have >1 dimension (MIMO), via
  - time-sharing, DMT, and/or multiple antennas.

Conundrum: double-sampling-rate Example

[80 60] channel again at twice sampling rate



a). capacity rate region



b). capacity energy region

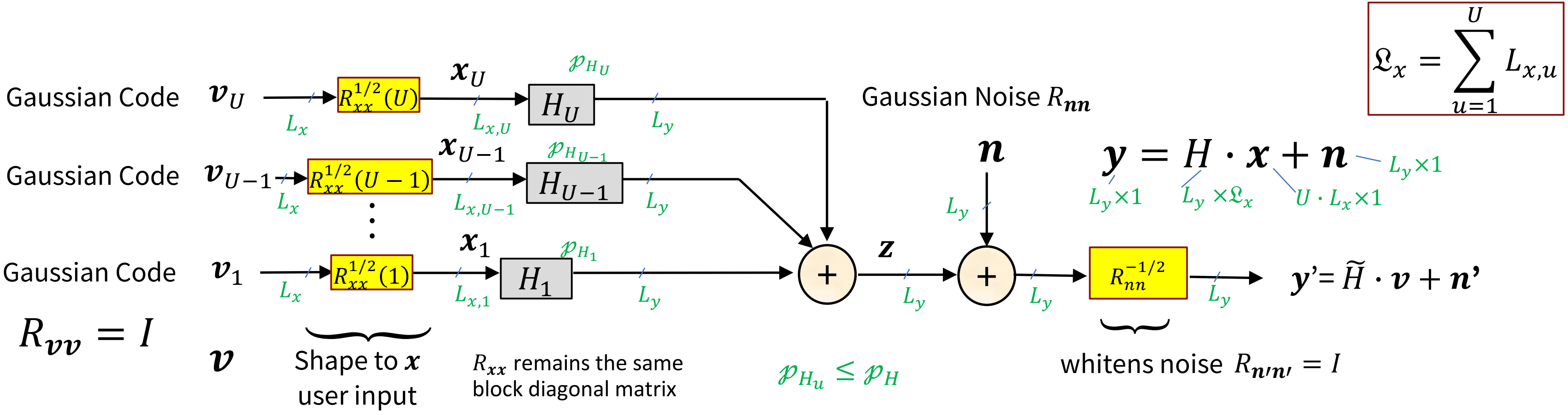
- The vector $\boldsymbol{b}$ is now in the interior of the region, although is it the same channel?
  - The time-sharing needs to occur at the same sampling rate, meaning the symbol period increases, for the original $\mathcal{C}(\boldsymbol{b})$ to apply.

Vector Gaussian MAC

PS4.4 - 2.24 MAC regions

Section 2.7.2.2

MAC ~ single channel with white input



- This normalizes (redefines, not $R_{noise}(u)$ here) individual user MAC channels to $\tilde{H}_u \triangleq R_{\boldsymbol{nn}}^{-1/2} \cdot H_u \cdot R_{\boldsymbol{xx}}^{1/2}(u)$.
- **Normalized MAC** is now $\boldsymbol{y}' = \tilde{H} \cdot \boldsymbol{v} + \boldsymbol{n}'$, where:
  - New input(s) is (are) "white" , $R_{\boldsymbol{vv}} = I$.
  - New noise is "white" , $R_{\boldsymbol{n'n'}} = I$.
  - We drop the primes going forward; $\boldsymbol{y} = \tilde{H} \cdot \boldsymbol{v} + \boldsymbol{n}$ → $\tilde{H}$ ' s dimensions carry the information (secondary may freeload).

Cholesky Factorization

- This is related to MMSE linear-prediction (see Appendix D).

- Positive definite Hermitian symmetric matrix factors as $R = G^* \cdot S \cdot G$, where
  - G is upper triangular monic (1's on diagonal), &
  - S is positive real diagonal matrix (even if R is complex).

- Matlab command is "**chol**" for *lower* × *upper* (lower is upper*) – **produces upper.**
  - Gtemp=chol (R)
  - G= inv(diag(diag(Gtemp)))*Gtemp
  - S= diag(diag(Gtemp))*diag(diag(Gtemp))

- See website's lohc.m program for *lower* × *upper* .

```
>> R=[2 1
1 2];
>> Gtemp=chol(R) % =
   1.4142   0.7071
        0   1.2247
>> G= inv(diag(diag(Gtemp)))*Gtemp %=
   1.0000   0.5000
        0   1.0000
 S= diag(diag(Gtemp))*diag(diag(Gtemp)) % =
   2.0000        0
        0   1.5000
>> G'*S*G % =
   2.0000   1.0000
   1.0000   2.0000
```

Forward and Backward Canonical Channels

- **Forward Canonical** Channel is
  - the output of matched-filter matrix.

$$\boldsymbol{y}' = \underbrace{\widetilde{H}^* \cdot \widetilde{H}}_{R_f} \cdot \boldsymbol{\nu} + \underbrace{\widetilde{H}^* \cdot \boldsymbol{n}}_{\boldsymbol{n'}},$$

- **MMSE Estimator** for backward channel

$$R_{\boldsymbol{\nu}\boldsymbol{y}'} \cdot R_{\boldsymbol{y}'\boldsymbol{y}'}^{-1} = R_f \cdot [R_f \cdot R_f + R_f]^{-1} = [R_f + I]^{-1} = R_b$$

- **Backward Canonical** Channel

$$\boldsymbol{\nu} = R_b \cdot \boldsymbol{y}' + \boldsymbol{e} \qquad R_{\boldsymbol{ee}} = R_b$$

- Use **Cholesky** on backward-channel inverse $R_b^{-1} = R_f + I = G^* \cdot S_0 \cdot G$

$$\boldsymbol{y}'' = S_0^{-1} \cdot G^{-*} \boldsymbol{y}' \qquad \text{(algebra)}$$

$$\boxed{\boldsymbol{y}'' = G \cdot \boldsymbol{\nu} - \boldsymbol{e}'}$$

where $R_{\boldsymbol{e}'\boldsymbol{e}'} = S_0^{-1}$.

Back Substitution

- Not quite ML/MAP, but **successive decoding,**
  - but **canonical –** achieves $\mathcal{I}$ reliably, each user**,**
  - **if decisions are correct** (asymptotic MMSE = MAP again).
  - If $\Gamma > 0$ dB, then iterative decoding that →ML may be needed.
- Each of these is MMSE based,
  - which is related to conditional $\mathcal{I}$.
- The decoder is much simpler decoder ("GDFE").
- SNR (biased) for each decision/dimension is $S_{0,u,l}$.
- But also

$$G = \begin{bmatrix} 1 & g_{U,U-1} & \cdots & g_{U,2} & g_{U,1} \\ 0 & 1 & \cdots & g_{U-1,2} & g_{U-1,1} \\ \vdots & \ddots & \cdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & g_{2,1} \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix}$$

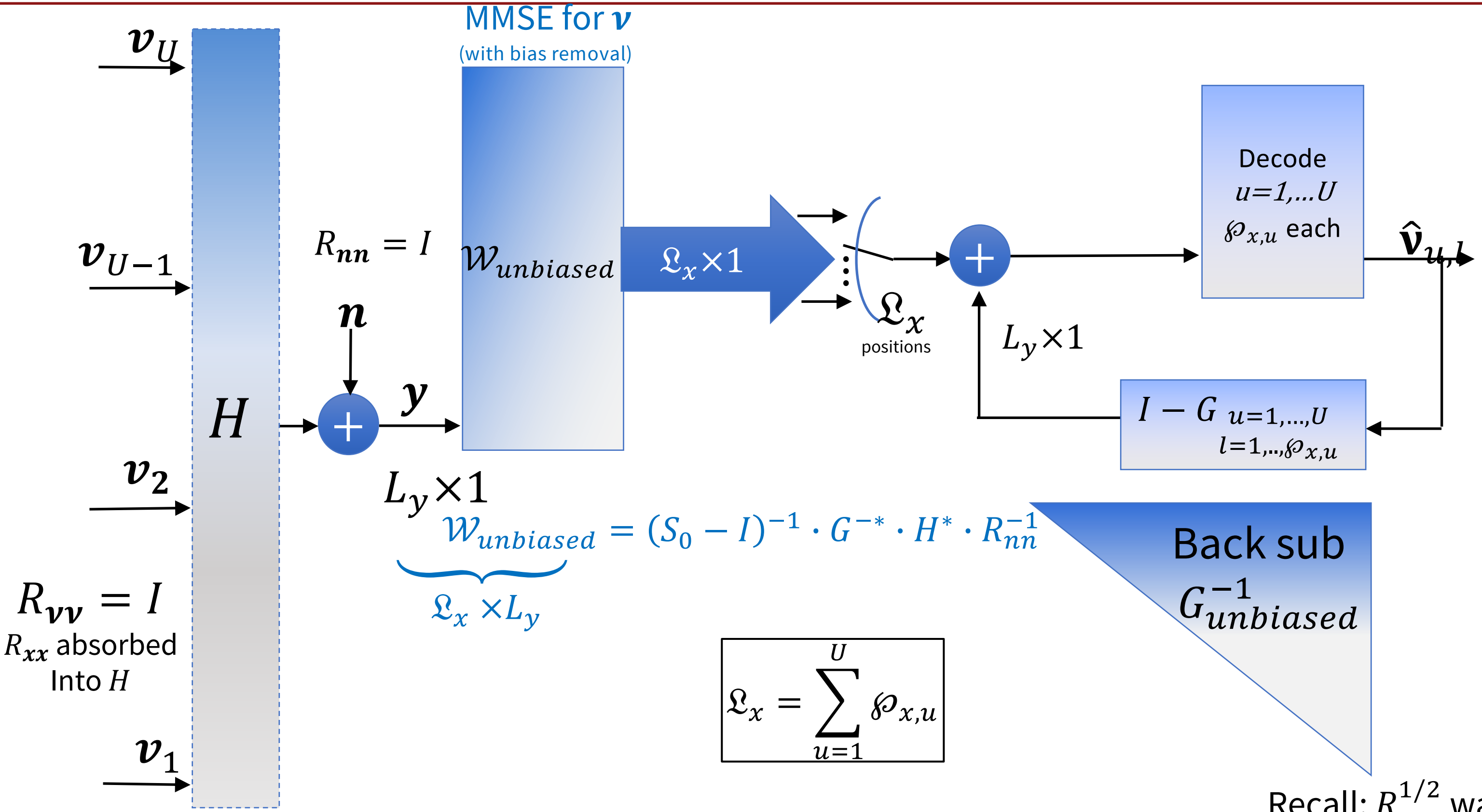
$$\begin{aligned} \hat{\boldsymbol{\nu}}_1 &= \text{decision}\left(\boldsymbol{y}_1''\right) \\ \hat{\boldsymbol{\nu}}_2 &= \text{decision}\left(\boldsymbol{y}_2'' - g_{2,1}\cdot\hat{\boldsymbol{\nu}}_1\right) \\ \vdots & \quad \vdots \quad \vdots \\ \hat{\boldsymbol{\nu}}_{U'} &= \text{decision}\left(\boldsymbol{y}_U'' - \sum_{i=1}^{U-1} g_{U,i}\cdot\hat{\boldsymbol{\nu}}_i\right) \end{aligned}$$

$$\mathcal{I}(\boldsymbol{x};\boldsymbol{y}) = \log_2(|\underbrace{\tilde{H}^*\tilde{H}+I}_{R_b^{-1}}|) = \log_2|S_0| = \log_2\left\{\prod_{u=1}^{U'}\prod_{\ell=1}^{L_{x,u}} SNR_{mmse,u,\ell}\right\} \text{bits / complex symbol} .$$

New parallel "independent" subchannels

CANONICAL RECEIVER (any R_{xx})

Vector MAC Receiver



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

$\underbrace{W_{unbiased}}_{\mathfrak{L}_x \times L_y}$

$$\mathfrak{L}_x = \sum_{u=1}^{U} \wp_{x,u}$$

Recall: $R_{\boldsymbol{xx}}^{1/2}$ was absorbed into $\widetilde{H}$

- Each user/decoder achieves $\mathcal{I}(\boldsymbol{v}_u; \boldsymbol{y} / [\boldsymbol{v}_{u-1} \;\; \cdots \;\; \boldsymbol{v}_1])$

Matrix AWGN MAC Example 1

$$\tilde{H} = \begin{bmatrix} 5 & 2 \\ 3 & 1 \end{bmatrix}$$

```
>> Rf=H'*H =
   34    13
   13     5
>> Rbinv=Rf+eye(2) =
   35    13
   13     6

>> Gbar=chol(Rbinv)   =
   5.9161    2.1974
        0    1.0823

>> S0=diag(diag(Gbar))*diag(diag(Gbar)) =
  35.0000         0
        0    1.1714
>> G = inv(diag(diag(Gbar)))*Gbar =
   1.0000    0.3714
        0    1.0000
>> >> b=0.5*log2(diag(S0))  =
   2.5646
   0.1141
>> sum(b) =    2.6788
```

REVERSE ORDER – same commands –other vertex

```
>> H=[   2    5
   1    3] ;
Rbinv =
    6    13
   13    35
Gbar =
 2.4495   5.3072
      0   2.6141
S0 =
   6.0000       0
        0   6.8333
G =
 1.0000   2.1667
      0   1.0000
b =
 1.2925
 1.3863

sum(b) =   2.6788
```

- These are the two vertices for dimension-share (pentagon outer face).

Example 1 continued

- Receiver filters and bias are

Vertex 1

```
>> W=inv(S0)*inv(G') =
    0.0286         0
   -0.3171    0.8537
>> Wunb=S0*inv(S0-eye(2))*W =
    0.0294         0
   -2.1667    5.8333

>> MSWMFu=Wunb*H' =
    0.1471    0.0882
    0.8333   -0.6667
>> Gunb=eye(2)+S0*inv(S0-eye(2))*(G-eye(2)) =
    1.0000    0.3824
         0    1.0000

>> MSWMFu*H =
    1.0000    0.3824
    2.1667    1.0000
```

Vertex 2

```
>> W=inv(S0)*inv(G') =
    0.1667         0
   -0.3171    0.1463
>> Wunb=S0*inv(S0-eye(2))*W =
    0.2000         0
   -0.3714    0.1714
>> MSWMFu=Wunb*H'  =
    0.4000    0.2000
    0.1143    0.1429
>> Gunb=eye(2)+S0*inv(S0-eye(2))*(G-eye(2)) =
    1.0000    2.6000
         0    1.0000
>> MSWMFu*H=
    1.0000    2.6000
    0.3714    1.0000
```

- Not really triangular, why?

Easier with mu_mac.m

```
function [b, GU, WU, S0, MSWMFU] = mu_mac(H, A, Lxu, cb)

%
[~ , U] = size(Lxu);
b=zeros(1,U);

% Computing Ht: Ht = H*A
Ht = H*A;
%-----------------------------------------------------------------------
---



% Computing Rf, Rbinv, Gbar
Rf = Ht' * Ht;
Rbinv = Rf + eye(size(Rf));
Gbar = chol(Rbinv);

% Computing the matrices of interest
G = inv(diag(diag(Gbar)))*Gbar;
S0 = diag(diag(Gbar))*diag(diag(Gbar));
W = inv(S0)*inv(G');

GU = eye(size(G)) + S0*pinv(S0-eye(size(G)))*(G-eye(size(G)));
WU = pinv(S0-eye(size(G)))*inv(G');
MSWMFU = WU*Ht';
index=0;
for u=1:U
  for l=1:Lxu(u)
  b(u) = b(u)+ (1/cb)*log2(S0(index+l,index+l));
  end
  index=index+Lxu(u);
end
```

channel — H; Rxx1/2 — A; #/user xmit antennas — Lxu; 1 cplx, 2 real — cb

```
H=[5 2 ; 3 1];
[b, GU, WU, S0, MSWMFU] = mu_mac(H, eye(2), [1 1] , 2);

b =   2.5646   0.1141

GU =
   1.0000   0.3824
        0   1.0000

WU =
   0.0294        0
  -2.1667   5.8333
S0 =
  35.0000        0
        0   1.1714

MSWMFU =
   0.1471   0.0882
   0.8333  -0.6667
>> MSWMFU*H =
   1.0000   0.3824
   2.1667   1.0000

>> SNR = 10*log10(diag(S0)) =
  15.4407
   0.6872

>> sum(b) =  2.6788
```

Example 2: 2 x 3 MAC (secondary users)

```
 H=[5 2 1
3 1 1];   basically added a 3rd user
[b, GU, WU, S0, MSWMFU] = mu_mac(H, eye(3), [1 1 1] , 2)

b =   2.5646      0.1141   0.1137
GU =    1.0000   0.3824    0.2353
                     0   1.0000    0.1667
                     0          0    1.0000
WU =
   0.0294       0       0
  -2.1667   5.8333       0
  -1.2857  -0.1429   5.8571
S0 =
  35.0000       0       0
       0   1.1714       0
       0       0   1.1707
MSWMFU =
   0.1471   0.0882
   0.8333  -0.6667
  -0.8571   1.8571
>> sum(b) =   2.7925
>> MSWMFU*H=
   1.0000   0.3824   0.2353
   2.1667   1.0000   0.1667
   1.2857   0.1429   1.0000
>>  SNR10*log10(diag(S0))=
  15.4407
   0.6872
   0.6846
```

- The channel rank is 2 so at least 1 secondary comp = 3-2.
- But secondary applies to energy-sum MAC (which this is not, yet).
- Original 2 units of energy is spread over 3 users?

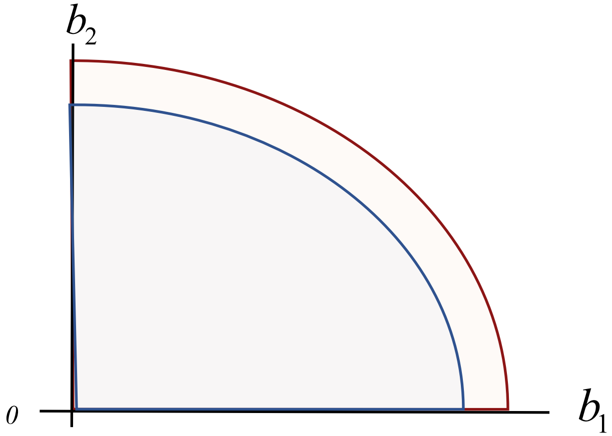
```
>> [b, GU, WU, S0, MSWMFU] = mu_mac(H, (2/3)*eye(3), [1 1 1] , 2)

b =   2.0050   0.1009   0.0696
GU =
   1.0000   0.3824   0.2353
        0      1.0000   0.3878
        0           0      1.0000
WU =
   0.0662       0           0
  -2.3878   6.6582       0
  -2.0000  -0.5000   9.8750
S0 =
  16.1111       0           0
       0     1.1502          0
       0            0     1.1013
MSWMFU =
   0.2206   0.1324
   0.9184  -0.3367
  -0.7500   2.2500
>> sum(b) =  2.1755  (lower than 2x2 value of 2.6788)
```

- Relatively more energy on secondary-user comp(s), bsum ↓.

Non-Zero Gap Achievable Region

- Construct $\mathcal{C}(\boldsymbol{b})$ with $\Gamma = 0$ dB.
- Reduce all rates by γ_b relative to boundary points.
- Inscribe smaller region $\mathcal{C}(\boldsymbol{b})$- $(\gamma_b \odot \mathbf{1})$.
- Square constellations instead of spheres (AWGN) loss 1.53 dB in gap above (0.25 bit/dimension).

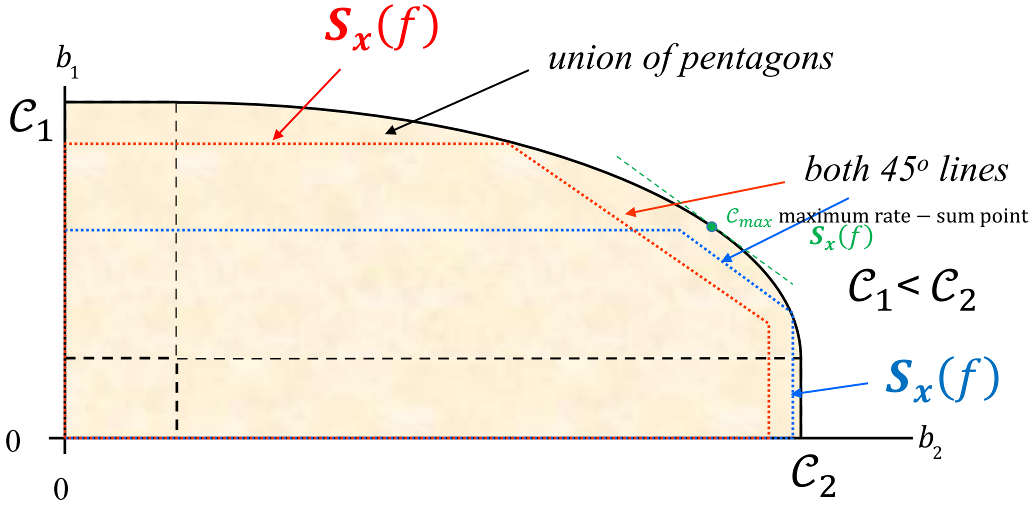


End Lecture 8
(back-up material FYI)

Capacity Region for continuous-frequency-indexed channels

Sections 2.7.4.1-2

$\mathcal{C}(b)$ is Union of $S_x(f)$-indexed Pentagons



$$\bar{b} = \sum_{u=1}^{U} \bar{b}_u \le \bar{\mathcal{I}}(\boldsymbol{x};\boldsymbol{y}) = \int_{-\infty}^{\infty} \frac{1}{2} \cdot \log_2 \left[1 + \frac{\sum_{u=1}^{U} S_{x,u}(f) \cdot |H_u(f)|^2}{S_n(f)} \right] df$$

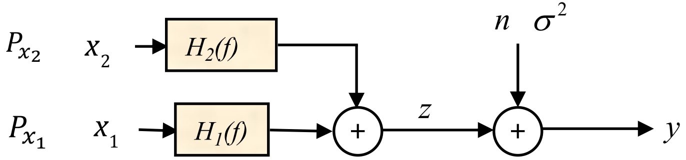
Calculus of variations again, decomposes into U water-fills, each with other users as noise – more details in L9.

$$S_{x,u}(f) + \frac{\sigma^2 + \sum_{i \neq u} S_{x,i}(f) + |H_i(f)|^2}{|H_u(f)|^2} = K_u$$

Simultaneous water-filling
→ Maximum rate sum

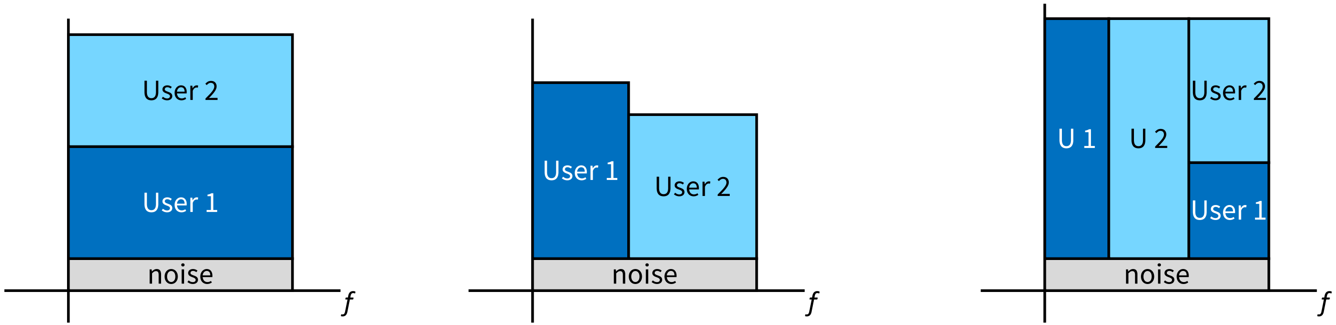
- Each pentagon corresponds to an $S_x(f)$ choice.
  - The pentagons become triangles for the sum-energy MAC.
- The union (convex hull is union when inputs are Gaussian) can dimension-share in frequency as $N \to \infty$.

MT MAC

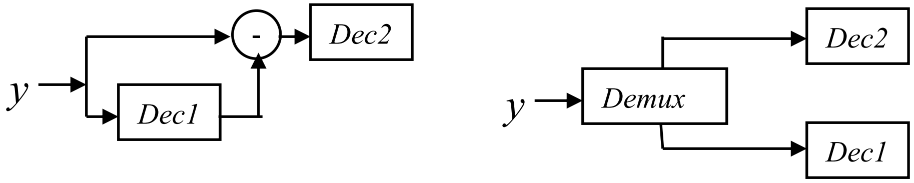


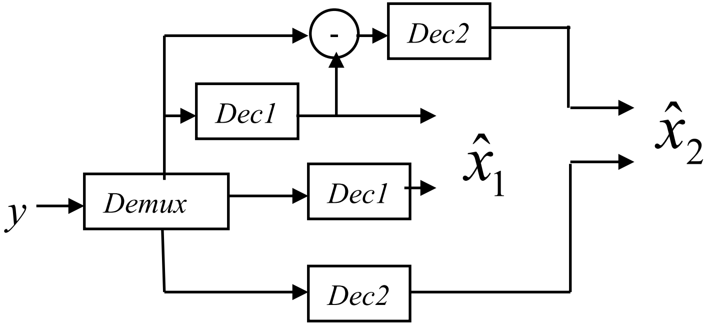
- The users have continuous-time/frequency channels → use MT on each, theoretically.
- This really means dimensionality is infinite (or very large) so "dimension-sharing" may be inherent.
- SWF applies, but with some interpretation (like power instead of energy, etc and power per dimension instead of power-spectral density, etc.)

Decoders and SWF



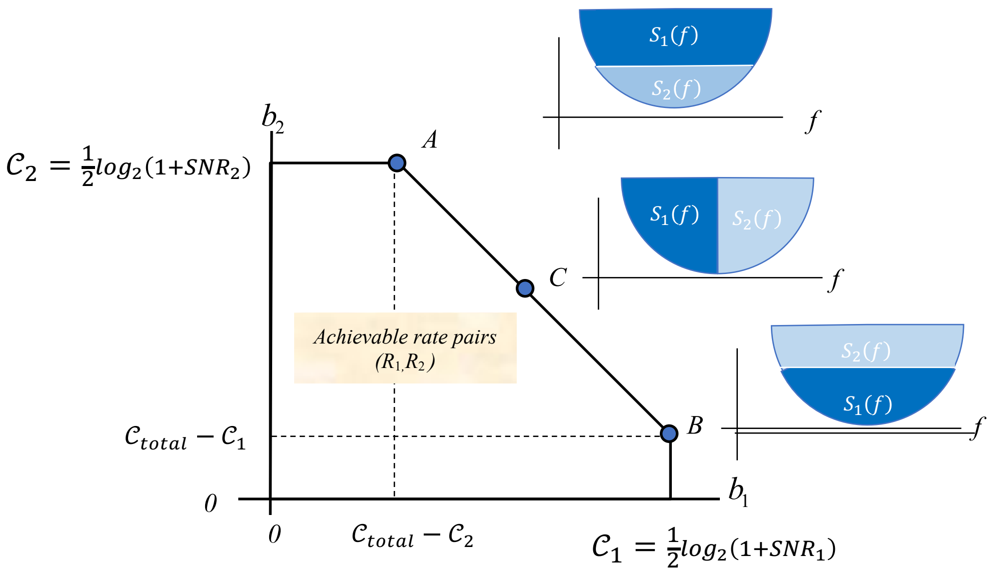
a). both flat *b). FDM* *c). mixed*





- FDM is clearly simplest decoder for max rate sum case.

Symmetric 2-user channel and SWF



- Symmetric means $H_1(f) = H_2(f)$ (noise is one-dimensional and added to sum)
- Each of points A, B, and C have different SWF spectra – all have same (max) rate sum