STANFORD

Lecture 9

Broadcast Channels

May 1, 2023

JOHN M. CIOFFI

Hitachi Professor Emeritus of Engineering

Instructor EE392AA – Spring 2023

Announcements & Agenda

- Announcements
  - Problem Set #4 due tomorrow at 17:00
  - PS#3 Solutions were posted Saturday
  - Midterm is Wednesday May 3, in class (open book, laptop, calc)
  - Problem Set 5 goes out Monday May 8
  - Finish 2.7 and 2.8,
  - Background D.3.6 for Cholesky, B.2.1 for Lattices.

- Agenda (L9)
  - Simultaneous Water-Filling for MAC max rate sum
  - **MAC:** Capacity region for frequency-indexed MACs
  - **BC:** Precoder Basics for the Matrix AWGN
  - Scalar Gaussian BC
  - MMSE BC Design – all users' $R_{\boldsymbol{xx}}(u)$ are known

- L10 Broadcast continued – worst-case noise and rate sums

Simultaneous Water-Filling for MAC max rate sum

Sections 2.7.3-4

Revisit the rate-sum mutual information

$$b = \sum_{u=1}^{U} \tilde{b}_u \leq \mathcal{I}\,(\boldsymbol{x};\boldsymbol{y}\,) = \log_2 \frac{|H \cdot R_{\boldsymbol{xx}} \cdot H^* + R_{\boldsymbol{nn}}|}{|R_{\boldsymbol{nn}}|}$$

- Maximum rate-sum focuses on the numerator
  - when optimizing over $R_{\boldsymbol{xx}}$

$$\max_{\{R_{\boldsymbol{xx}}(u)\}} \left| H_u \cdot R_{\boldsymbol{xx}}(u) \cdot H_u^* + \underbrace{\sum_{i \neq u} H_i \cdot R_{\boldsymbol{xx}}(i) \cdot H_i^* + R_{\boldsymbol{nn}}}_{R_{noise}(u)} \right|$$

- Have we seen this problem before?
  - Yes, it is Vector Coding / Waterfilling , except with $\tilde{H}_u \to R_{noise}^{-1/2} \cdot H_u$ for each u

- But now it repeats U times, identical form for each user
  - The solution is each user **simultaneously water-fills** treating all other users (water-fill) spectra as noise

Simultaneous Waterfilling

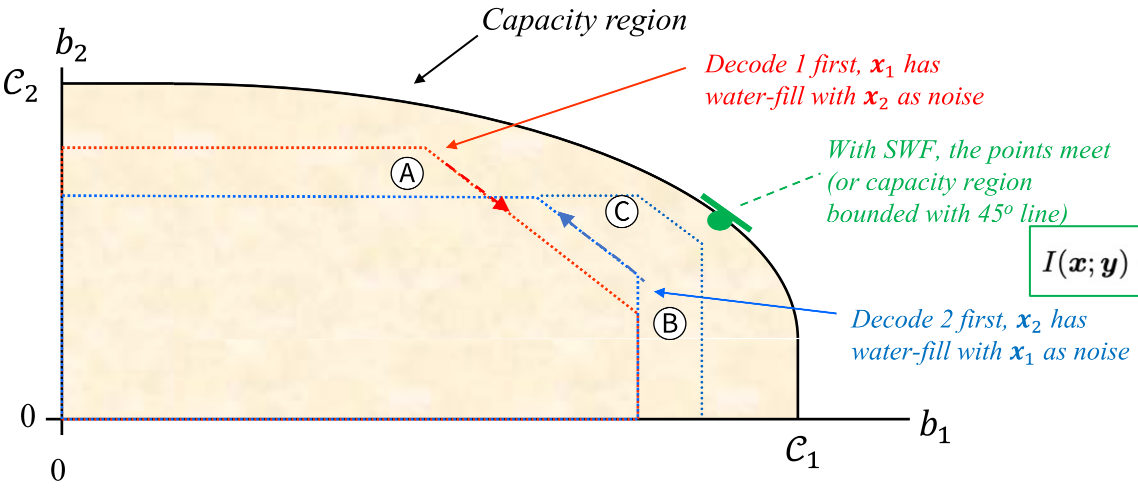
$$\mathcal{E}_{u,l} + \frac{1}{g_{u,l}} = K_u \;\forall\; u = 1, \ldots, U'$$

$$\sum_{l=1}^{L_x} \mathcal{E}_{u,l} = \mathcal{E}_u$$

$$\mathcal{E}_{u,l} \geq 0$$

$$\boldsymbol{x}_u = M_u \cdot \boldsymbol{v}_u$$

Compute Using Iterative Water-filling



$$I(\boldsymbol{x};\boldsymbol{y}) = I(\boldsymbol{x}_1;\boldsymbol{y}) + I(\boldsymbol{x}_2;\boldsymbol{y}/\boldsymbol{x}_1) = I(\boldsymbol{x}_2;\boldsymbol{y}) + I(\boldsymbol{x}_1;\boldsymbol{y}/\boldsymbol{x}_2)$$

- SWC problem is convex, and each single-water-fill step is like gradient in improving direction, swf.m
- E-Sum SWC is saddle point with enlarged region

SWF.m Program

```
function [Rxx, bsum , bsum_lin] = SWF(Eu, H, Lxu, Rnn, cb)

 Simultaneous water-filling MAC max rate sum (linear and nonlinear GDFE)
 The input is space-time domain h, and the user can specify a temporal
 block symbol size N (essentially an FFT size).

 Inputs:
 Eu  U x 1 energy/SAMPLE vector. Single scalar equal energy all users
    any (N/N+nu) scaling should occur BEFORE input to this program.
 H   The FREQUENCY-DOMAIN Ly x sum(Lx(u)) x N MIMO channel for all users.
    N is determined from size(H) where N = # used tones
 Lxu 1xU vector of each user's number of antennas
 Rnn The Ly x Ly x N noise-autocorrelation tensor (last index is per tone)
 cb  cb = 1 for complex, cb=2 for real baseband
      cb=2 corresponds to a frequency range at an sampling rate 1/T' of
      [0, 1/2T'] while with cb=1, it is [0, 1/T'].  The Rnn entered for
      these two situations may differ, depending on how H is computed.

 Outputs:
 Rxx A block-diagonal psd matrix with the input autocorrelation for each
    user on each tone. Rxx has size (sum(Lx(u)) x sum(Lx(u)) x N .
    sum trace(Rxx) over tones and spatial dimensions equal the Eu
 bsum the maximum rate sum.
 bsum  bsum_lin - the maximum sum rate with a linear receiver
    b is an internal convergence sum rate value, not output

 This program significantly modifies one originally supplied by student
 Chris Baca
```

- Eu is energy**/sample**
- For now, $N = 1$, so time/freq are same
  - H=h
- Lxu – number of antennas for each user
- Separate specification of Rnn removes need for noise whitening
- cb=1 for complex, 2 for real

Revisit Previous example (slides 26-29)

```
H =
   5   2   1
   3   1   1
>> [Rxx, bsum , bsum_lin] = SWF([1 1 1], H, [1 1 1], eye(2),  2)
Rxx =
   1   0   0
   0   1   0
   0   0   1
bsum =   2.7925
bsum_lin =   1.4349
```

- Same result as L8:29, so each user waterfills with all others as noise; this is trivial when each user has only 1 input dimension. (Why?)
- This is for input energy-vector constraint.
- Note linear solution (no feedback, so matrix MMSE-LE) loses roughly ½ the data rate
- SWF becomes more interesting when $N > 1$ tones or if $L_{x,u} > 1$ antennas

For $L_{x,u} = 2$; $u = 1,2$?

```
>> H2=[4 3 2 1
5 6 7 8];
>> [Rxx, bsum , bsum_lin] = SWF([0.5 0.5], H2, [2 2 ], eye(2),  2)
Rxx =
   0.7121   0.4528        0        0
   0.4528   0.2879        0        0
        0        0   0.2876   0.4527
        0        0   0.4527   0.7124
bsum =   5.3434
bsum_lin =   4.0920
>> trace(Rxx) =    2  (check)
```

Energy input is per sample!

- Note block-diagonal Rxx
- Linear-only is about 25% less data rate

Or can use Macmax.m

```
function [Rxx, bsum , bsum_lin] = macmax(Eu, h, Lxu, N , cb)

 Simultaneous water-filling Esum MAC max rate sum (linear & nonlinear GDFE)
 The input is space-time domain h, and the user can specify a temporal
 block symbol size N (essentially an FFT size).

 This program uses the CVX package

 the inputs are:
 Eu  The sum-user energy/SAMPLE scalar.
    This will be increased by the number of tones N by this program.
    Each user energy should be scaled by N/(N+nu)if there is cyclic prefix
    This energy is the trace of the corresponding user Rxx (u)
    The sum energy is compouted as the sum of the Eu components
    internally.
 h   The TIME-DOMAIN Ly x sum(Lx(u)) x N channel for all users
 Lxu  The number of antennas for each user 1 x U
 N   The number of used tones (equally spaced over (0,1/T) at N/T.
 cb  cb = 1 for complex, cb=2 for real baseband

 the outputs are:
 Rxx A block-diagonal psd matrix with the input autocorrelation for each
    user on each tone. Rxx has size (sum(Lx(u)) x sum(Lx(u)) x N .
    sum trace(Rxx) over tones and spatial dimensions equal the Eu
 bsum the maximum rate sum.
 bsum  bsum_lin - the maximum sum rate with a linear receiver

    b is an internal convergence (vector, rms) value, but not sum rate
```

- ENERGY-SUM input (per sample)
  - Time-domain noise whitened
  - Lxu = numbers of xmit antennas/user
- This is actually a double loop with
  - Water-filling for each user for some current set of per-user energies
  - Adjustment of energies so they sum to total but increase the rate sum
- It corresponds to a saddle point
  - Not convex
- Will be easier understood later as a dual of a broadcast problem as to why this is true.

Back to Example

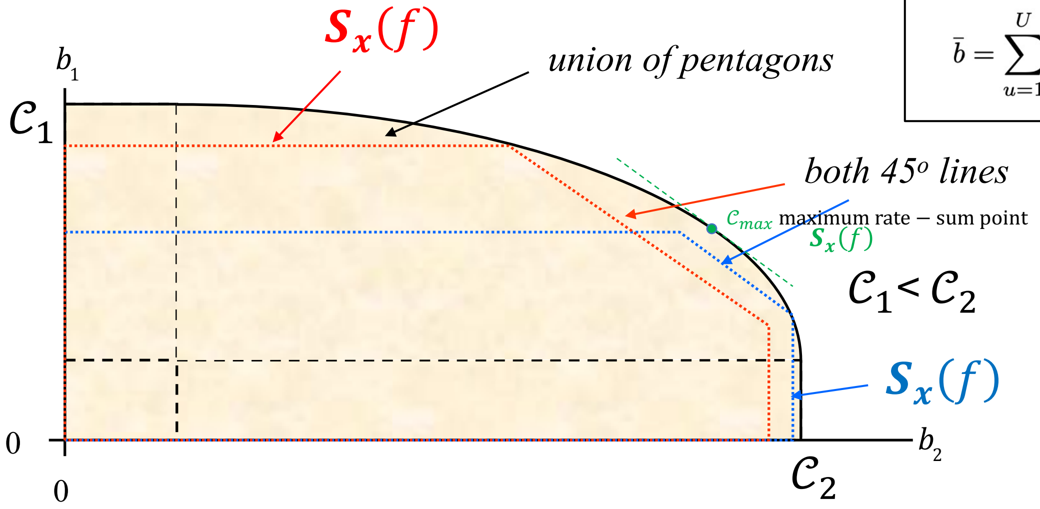
```
>> H3(:,:,1)=H

H =
    5    2    1
    3    1    1
;
>> [Rxx, bmacmax, bmaclin]=macmax(3, H, [1 1 1], 1, 2)
Rxx =
   3.0000        0        0
        0   0.0000        0
        0        0   0.0000
bmacmax =    3.3432
bmaclin =    3.3432
```

- Even larger data rate
- Rxx energizes just user 3! (it's all primary user component and users 2 and 1 are secondary)
- Linear is the same. Why?

Capacity region for frequency-indexed MACs

Sections 2.7.4.1-2

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



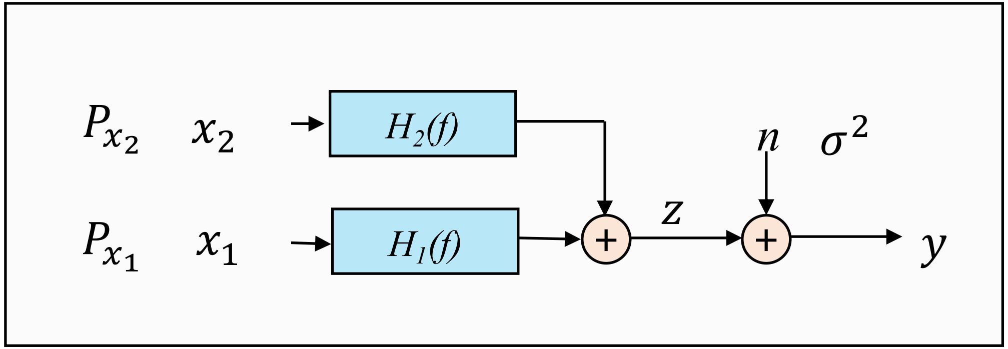
$$\bar{b} = \sum_{u=1}^{U} \bar{b}_u \leq \bar{\mathcal{I}}(\boldsymbol{x};\boldsymbol{y}) = \int_{-\infty}^{\infty} \frac{1}{2} \cdot \log_2 \left[1 + \frac{\sum_{u=1}^{U} S_{x,u}(f) \cdot |H_u(f)|^2}{S_n(f)} \right] df$$

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

Simultaneous water-filling
→ Maximum rate sum

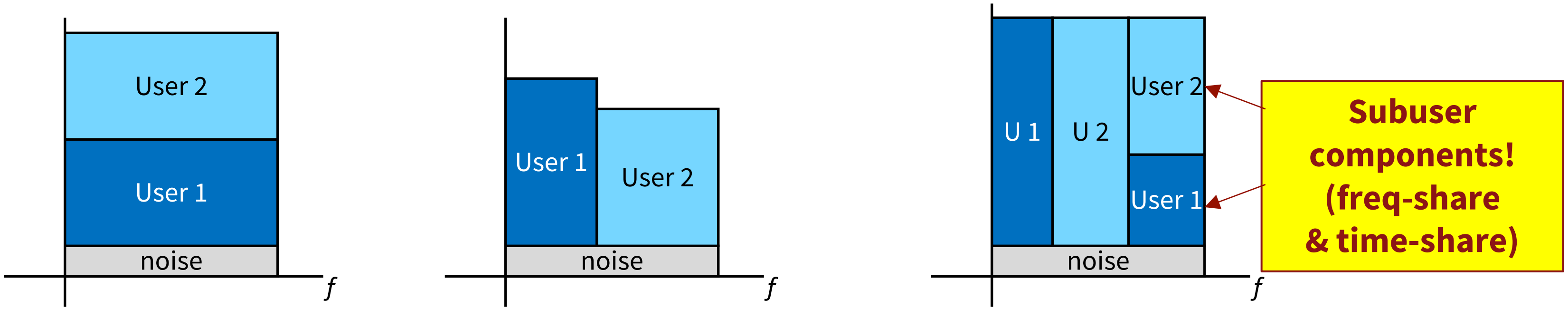
- Each pentagon corresponds to an $\boldsymbol{S_x}(f)$ choice.
  - The pentagons become triangles for the sum-energy MAC.
- The union (convex hull is union when inputs are Gaussian) can dimension-share in frequency as $N \to \infty$.

MT MAC

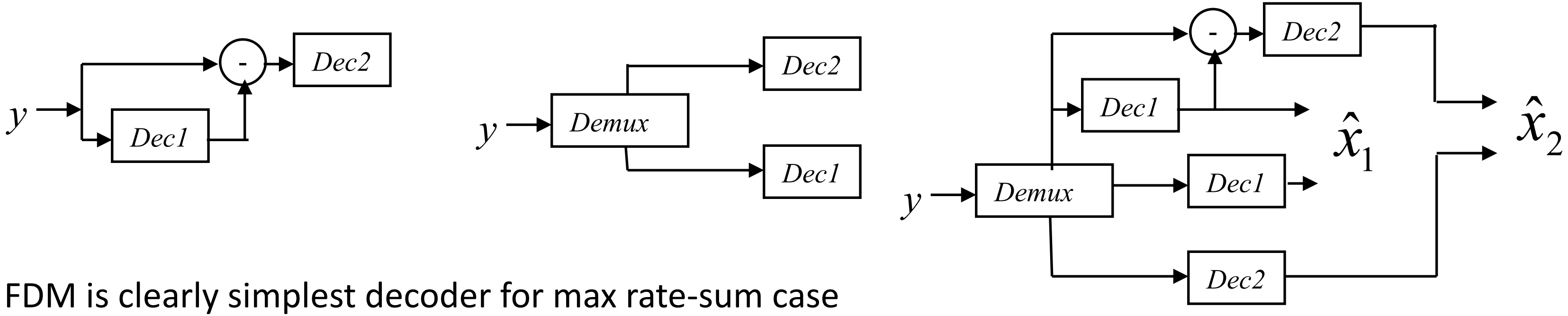


- The users have continuous-time/frequency channels → use MT on each, theoretically
- This really means dimensionality is infinite (or very large) so "dimension-sharing" may be inherent
- SWF applies, but with some interpretation (like power instead of energy, etc and power per dimension instead of power-spectral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

Decoders and SWF



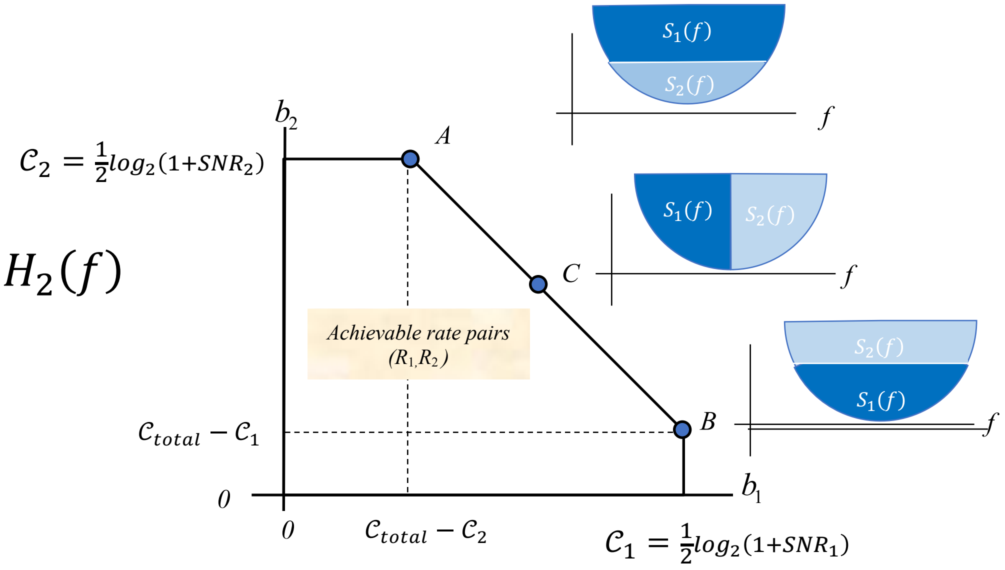
a). both flat *b). FDM* *c). Mixed – sub users*



- FDM is clearly simplest decoder for max rate-sum case
- Both users (and all components in case c) are primary

Symmetric 2-user channel and SWF

$$H_1(f) = H_2(f)$$

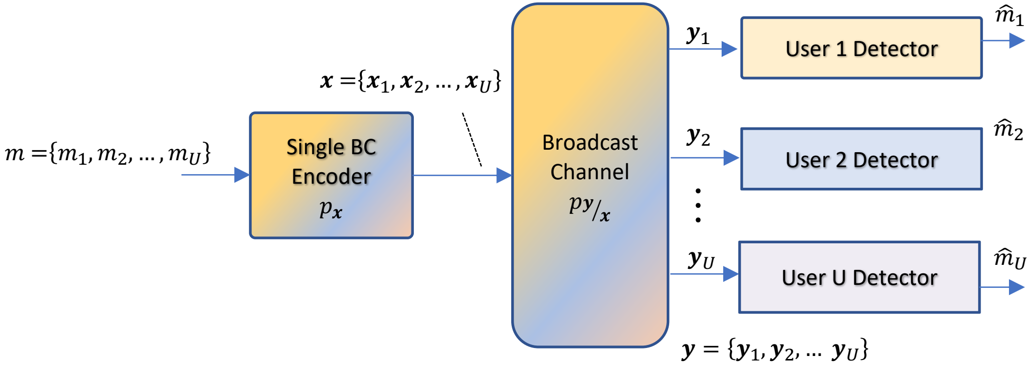


- Symmetric means $H_1(f) = H_2(f)$ (noise is one-dimensional and added to sum)
- Each of points A, B, and C have different SWF spectra – all have same (max) rate sum

Basic Precoders and the Matrix AWGN

PS5.1 - 2.28 modulo precoding function

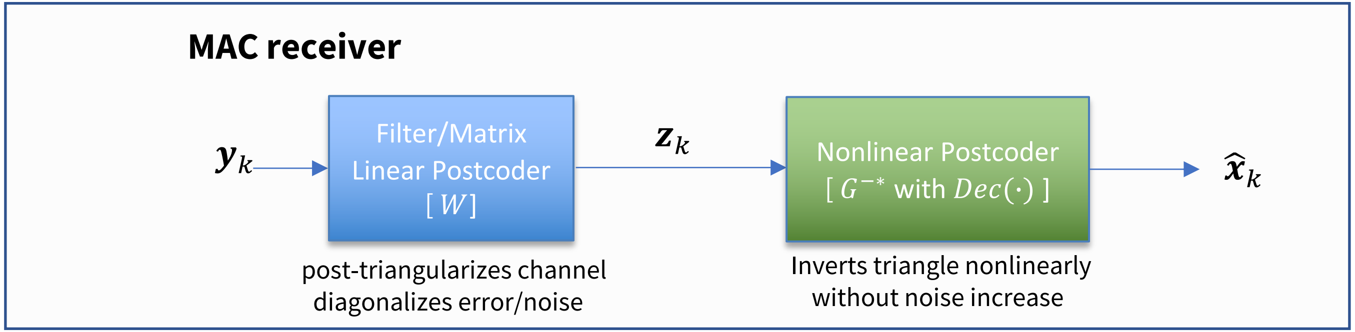
Broadcast Channel (BC)

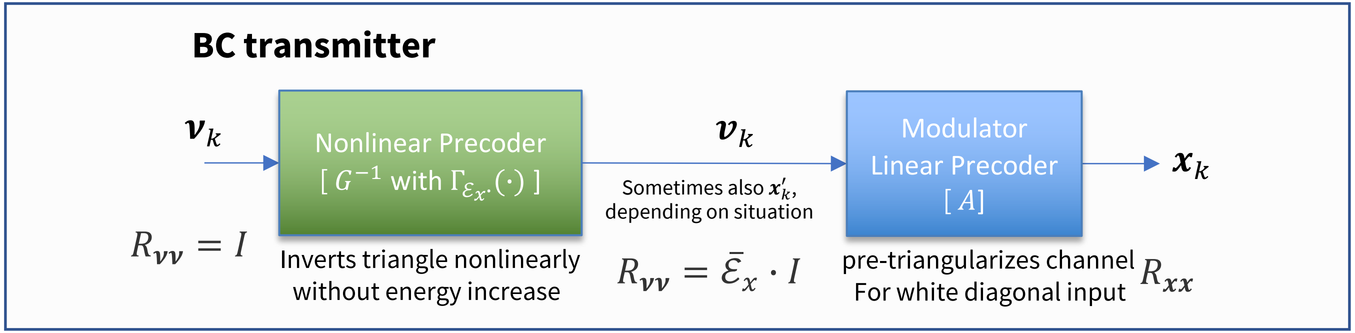


- "Dual" of MAC

- Receivers in different places – cannot "co-process" $\{\boldsymbol{y}_u\}$
- Transmitter can co-encode/generate $\boldsymbol{x}$, although input messages remain independent
  - Who encodes first? (may be at disadvantage)
  - Who encodes last? (knowing other users' signals is an advantage)
  - What then is the **order**?

BC is "reversed" MAC





- The MAC's uncoordinated user input is a kind of "worst case" transmitter, reducing data rate
  - With only an energy-sum constraint, these worst-case inputs' users best pass as primary user components; secondary components "freeload" on the primary's passage
- The BC similarly will effectively correspond to a worst-case noise for which receiver coordination is useless, reducing data rate
  - With worst-case noise, the channel best passes the primary components' ; secondary components freeload on the primary's passage

Triangular Matrices - Innovations and Prediction

- Prediction for some **user order** leads to a a way to have independent users' messages combine

$$\boxed{\boldsymbol{\nu}_u = \boldsymbol{x}_u - \hat{\boldsymbol{x}}_{u/\{x_{u+1}\ldots x_U\}}}$$

Innovations or predictions, but for BC they could be the independent-users' subsymbols, with normalization $R_{\boldsymbol{\nu}\boldsymbol{\nu}}(u) = I$

- This is a triangular relationship (inverse of upper triangular is upper triangular)

$$\boldsymbol{\nu} = \begin{bmatrix} \boldsymbol{\nu}_1 \\ \boldsymbol{\nu}_2 \\ \vdots \\ \boldsymbol{\nu}_U \end{bmatrix} = \begin{bmatrix} 1 & g_{1,2} & \ldots & g_{1,U} \\ 0 & 1 & \ldots & g_{2,U} \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \ldots & 1 \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_1 \\ \boldsymbol{x}_2 \\ \vdots \\ \boldsymbol{x}_U \end{bmatrix} = G^{-1} \cdot \boldsymbol{x}$$

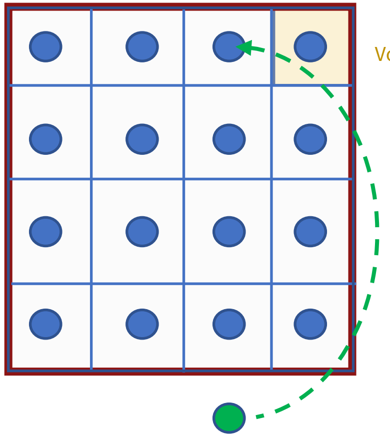
- OR, $\boldsymbol{x} = G \cdot \boldsymbol{\nu}$ **(which is also upper triangular relationship)**
- *Generating $\boldsymbol{x}$ from $\boldsymbol{\nu}$ can increase energy (or enhance noise in MAC) if implemented directly (linearly)*

(order reversal is intentional)

Voronoi Regions and Modulo Addition (Sec 2.1)

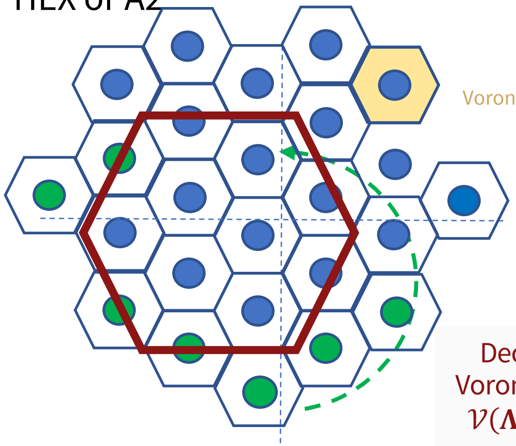
- A lattice is a (countable) group of vectors $\boldsymbol{\Lambda} = \{\boldsymbol{x}\}$ that is closed under an operation addition, so that
  - If $\boldsymbol{x}_1 \in \boldsymbol{\Lambda}$ and $\boldsymbol{x}_2 \in \boldsymbol{\Lambda}$, then $\boldsymbol{x}_1 + \boldsymbol{x}_2 \in \boldsymbol{\Lambda}$. (Section 2.2.1.1 and Appendix B.2)
  - A constellation is a finite subset of a lattice, plus a constant (coset) $C \subset \boldsymbol{\Lambda} + \boldsymbol{\lambda_0}$. ($\boldsymbol{\lambda_0}$ ensures average value is zero.)

SQ rectangular or $\mathbb{Z}^2$



HEX or A2

$\boldsymbol{\lambda_0}$ is the "coset leader"



Green indicates Modulo operation (split ties equally)

- Voronoi Region of a lattice, $\mathcal{V}(\Lambda_c)$ is the decision region around any point with volume $V(\Lambda_c)$.
  - Λ_c is the "coding" lattice; codes try to pack more points into limited space (volume/area). – HEX is better than SQ.
- A constellation C typically selects points in one (coding-gain) lattice, Λ_c , within the $\mathcal{V}(\Lambda_s)$ of another (shaping-gain) lattice $\boldsymbol{\Lambda}'$ that is larger (can be scaled versions of one another or possibly different). (Subtract any nonzero vector mean to save energy.)
  - All points in Λ_c outside of $\mathcal{V}(\Lambda_s)$ map into a point inside $\mathcal{V}(\Lambda_s)$ - disguised detector problem.

Formal modulo & simple precoder

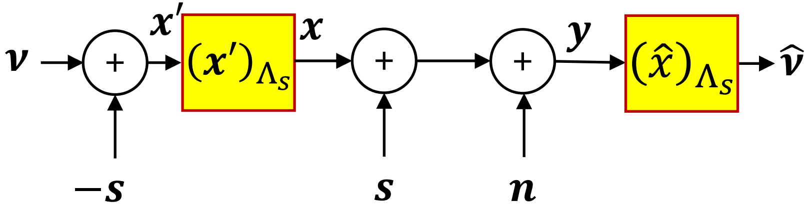
- Definition of **Modulo Operation** $\boxed{(\boldsymbol{\nu})_{\Lambda_s} = \boldsymbol{e} \ni \min_{\boldsymbol{\lambda}\in\Lambda_s} \|\boldsymbol{e}\|^2 \ \text{ where } \ \boldsymbol{e} = \boldsymbol{\nu} - \boldsymbol{\lambda}}$

- $\boldsymbol{e}$ does not necessarily need to be a point in Λ_c ; instead, it is a point in $\mathcal{V}(\Lambda_s)$

- useful **Lemma 2.8.1 (distribution of modulo addition)** *Modulo addition distributes as*

$$(\boldsymbol{\mu}+\boldsymbol{\nu})_\Lambda = (\boldsymbol{\mu})_\Lambda \oplus_\Lambda (\boldsymbol{\mu})_\Lambda \quad . \tag{2.371}$$

- Side info $\boldsymbol{s}$ (simple precoder)
  - Added anywhere but $\boldsymbol{s}$ *is known*
  - Pre-subtract (precode) and use modulo to set $\mathcal{E}_x$ level (no increase for - $\boldsymbol{s}$)
  - $\boldsymbol{x}$ will effectively have continuous uniform distributions over Λ_s



Dirty paper

Nonlinear

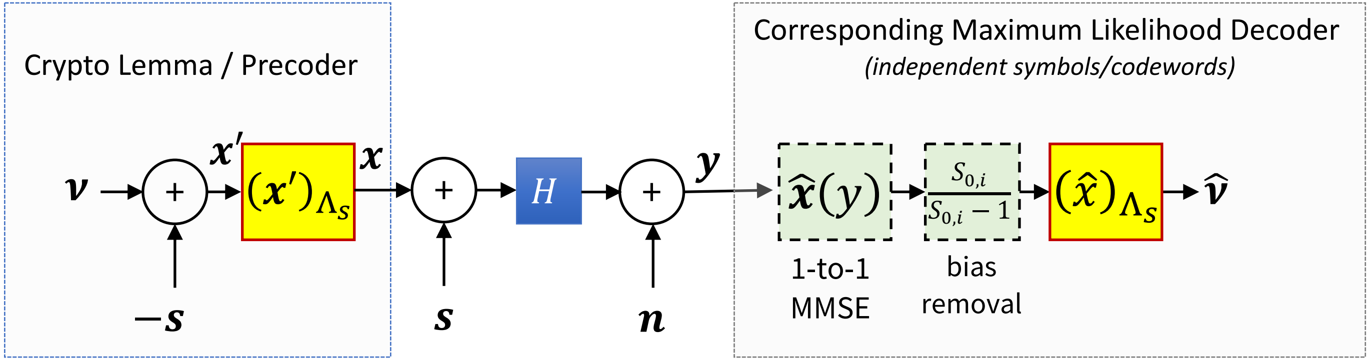
precoder

…

For BC: $\boldsymbol{s}$ will be the earlier users (but their xtalk cancels) ~ **order**

With nontrivial channel, need MMSE version

Forney's Crypto Lemma – 2003 (Section 2.8.1.2)



No xmit energy increase
Simplifies ML detection

- The MMSE part can be important in non-trivial cases (often missed in most precoder texts)
  - It's undoing the channel crosstalk and/or ISI in MMSE sense
- When $\boldsymbol{s}$ is uniform over $\mathcal{V}(\Lambda_s)$, then so is $\boldsymbol{x}$, **AND** $\boldsymbol{x}$ is independent of both $\boldsymbol{s}$ and $\boldsymbol{\nu}$ (like encryption), $\boldsymbol{s}$ is the "key"
  - Or "writing on dirty paper" ($\boldsymbol{s}$ is the dirt, $\boldsymbol{\nu}$ is the writing, and the second modulo cleans it)
- Sometimes the channel adds $\boldsymbol{s}$, sometimes the transmitter adds $\boldsymbol{s}$ (xmit case, $\boldsymbol{s}$ shares dimensions and energy with $\boldsymbol{x}$)

Non-Causal ?

- Subtly, the lattice Λ_s has a dimensionality N over which $\boldsymbol{s}$ and $\boldsymbol{x}$ are uniform distributed.

- Wise dimension use with fixed energy $\mathcal{E}_x$ suggests Λ_s has a hyper-spherical boundary, as $N \to \infty$.

- Asymtotically, the modulo has infinite number of dimensions, so requires infinite delay for $\boldsymbol{s}$ to be fully known in the formation of $\boldsymbol{x}$**;** whence "non-causal."
  - Approximated with finite delay in practice, $\boldsymbol{s}$ becomes another user's encoded signal known first (~ non-causal) → **order .**

$$\boxed{(\cdot)_{\Lambda_s}} \Longrightarrow \boxed{(\cdot)_{\mathcal{E}_x}}$$

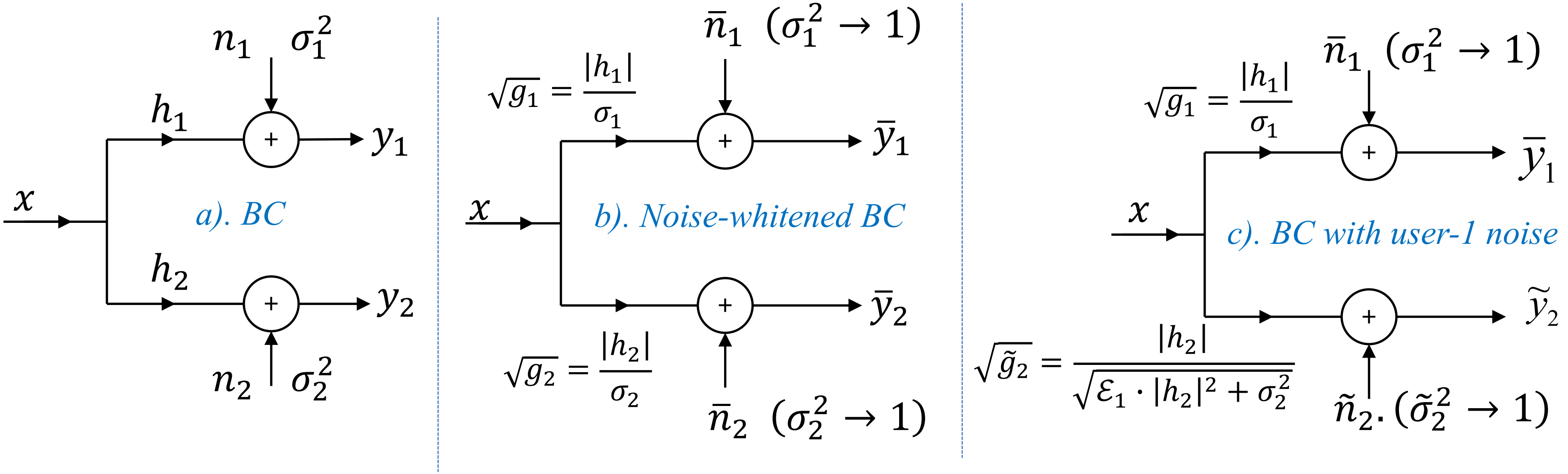
- Mod holds energy at $\mathcal{E}_x$ (Gaussian in any finite number of dimensions, uniform in infinite dimensional hypersphere)

- If Λ_s is hypercube, Forney's crypto still holds but with SNR loss of (up to) 1.53 dB (the maximum shaping gain).
  - So reuse code with $\Gamma \to 0$ dB, with QAM constellations and the (up to) 1.53 dB loss remains (greatly simplifies precoder implementation)

Scalar Gaussian BC

PS 5.2 - 2.29   scalar BC region

3 scalar-BC "scalings"

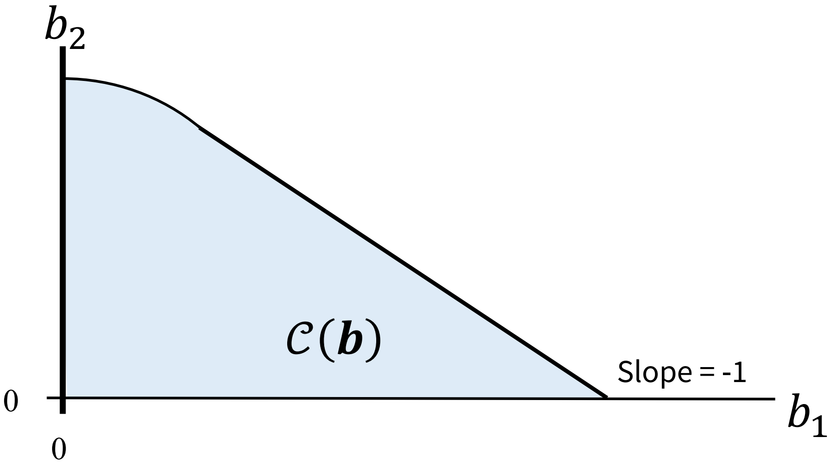


- They're all equivalent, but 3 different scalings
- Best order? $g_1 > g_2$ both users data rates are higher if 2 decoded first with 1 as noise
- Inductively, $g_1 > \cdots > g_U$ is the single best order (no search needed!)

Rate region

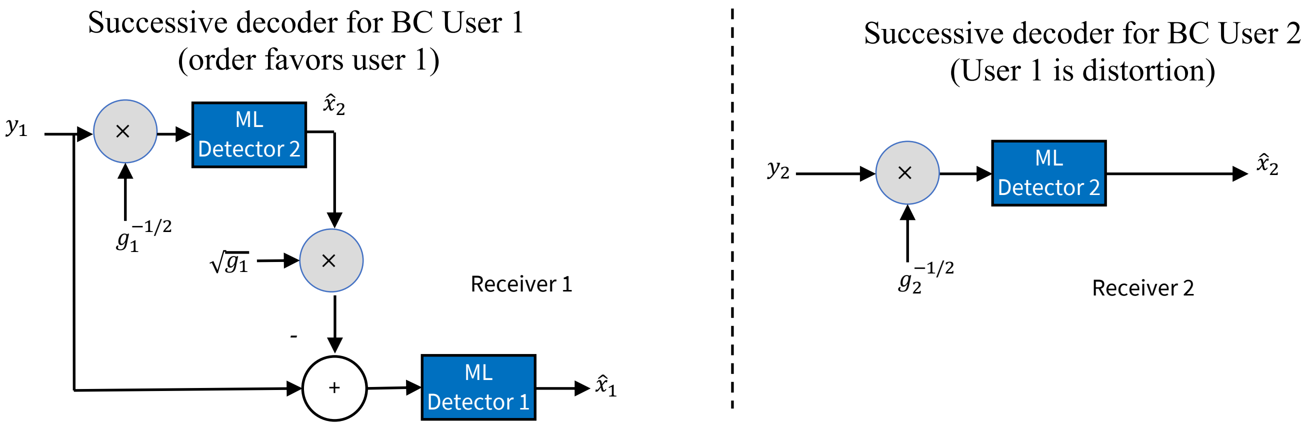
$$\bar{b}_1 \leq \mathcal{I}(x_1 : y_1/x_2) = \frac{1}{2}\cdot\log_2(1+\bar{\mathcal{E}}_x\cdot g_1)$$

$$\bar{b}_2 \leq \mathcal{I}(x_2 : y_2) = \frac{1}{2}\cdot\log_2\left(1+\frac{(1-\alpha)\cdot\bar{\mathcal{E}}_x\cdot g_2}{1+\alpha\cdot\bar{\mathcal{E}}_x\cdot g_2}\right)$$



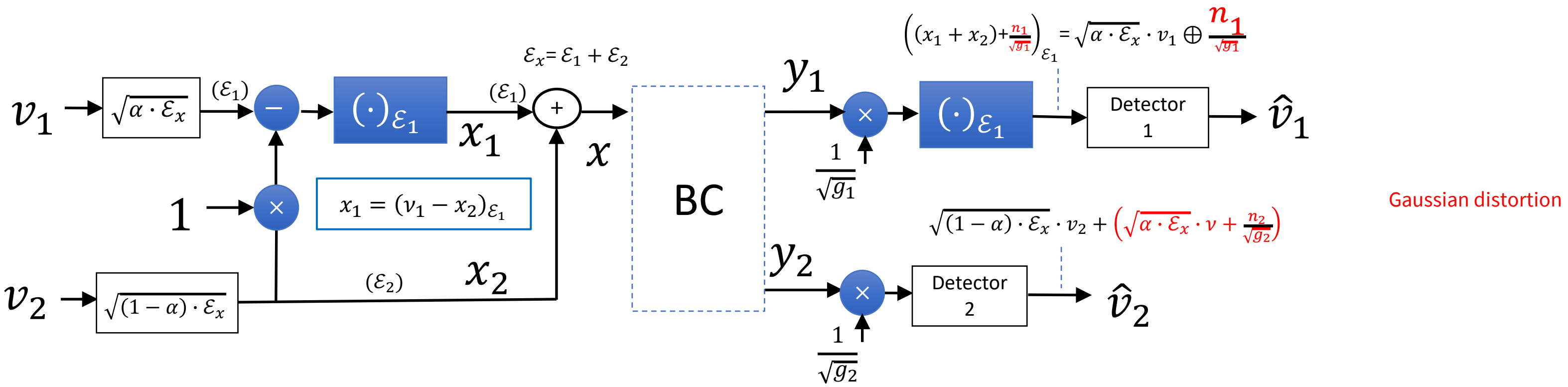
- Run through all energy splits (this is single parameter α in 2-user BC)

BC Successive Decoders

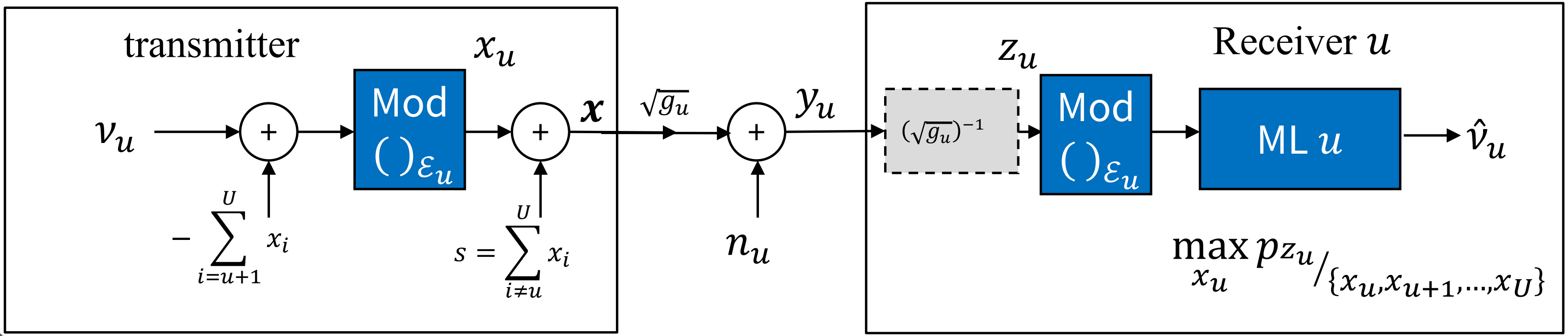


- U ML-U detectors; or really $\frac{U}{2} \cdot (U+1) = \sum_{u=1}^{U} u$ total detectors
- A precoder simplifies to U uses of the same modulo at transmitter (+ 1 modulo at each receiver)

Scalar Precoder



- The side information becomes x_2 and $\mathcal{E}_x = \mathcal{E}_1 + \mathcal{E}_2$; the receiver modulo removes x_2
- Can be inductively (recursively) applied from U … 1

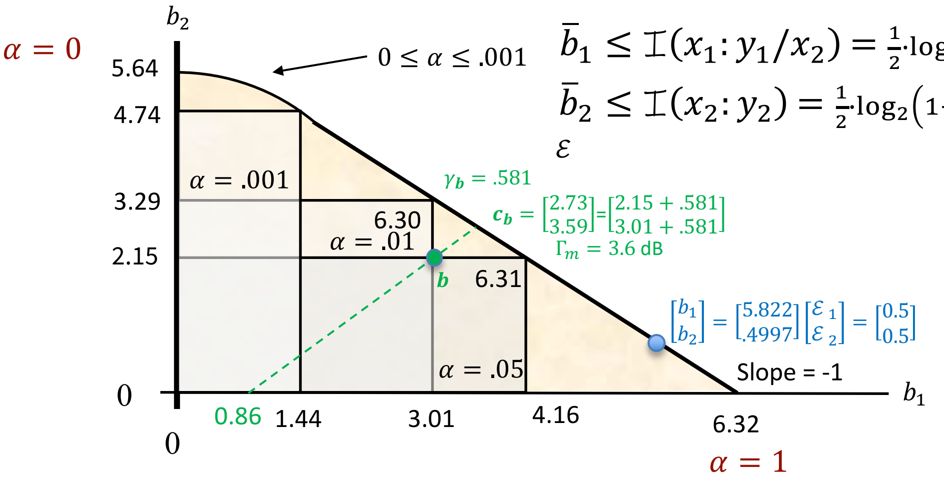


Example

- $h_1 = 0.8\,; h_2 = 0.5;\ \sigma_1^2 = \sigma_2^2 = .0001$

$$\mathcal{I}(x_1 : \boldsymbol{y}) = \frac{1}{2}\cdot\log_2\left(\frac{|R_{\boldsymbol{xx}}|}{|R_{\boldsymbol{nn}}|}\right) = \frac{1}{2}\cdot\log_2\left(\frac{(.6401)\cdot(.2501)-.4^2}{.0001^2}\right) = 6.56$$

α	$\bar{b}_1$	$\bar{b}_2$	$\bar{b} = \bar{b}_1 + \bar{b}_2$
1.0	6.32	0	6.32
.75	6.12	.20	6.32
.50	5.82	.50	6.32
.25	5.32	1.0	6.32
.10	4.66	1.66	6.32
.05	4.16	2.15	6.31
.01	3.01	3.29	6.30
.001	1,44	4.74	6.18
0	0	5.64	5.64



$$\bar{b}_1 \le \mathcal{I}(x_1 : y_1/x_2) = \frac{1}{2}\cdot\log_2(1+\bar{\mathcal{E}}_x\cdot g_1)$$

$$\bar{b}_2 \le \mathcal{I}(x_2 : y_2) = \frac{1}{2}\cdot\log_2\left(1+\frac{(1-\alpha)\cdot\mathcal{E}_x\cdot g_2}{1+\alpha\cdot\mathcal{E}_x\cdot g_2}\right)$$

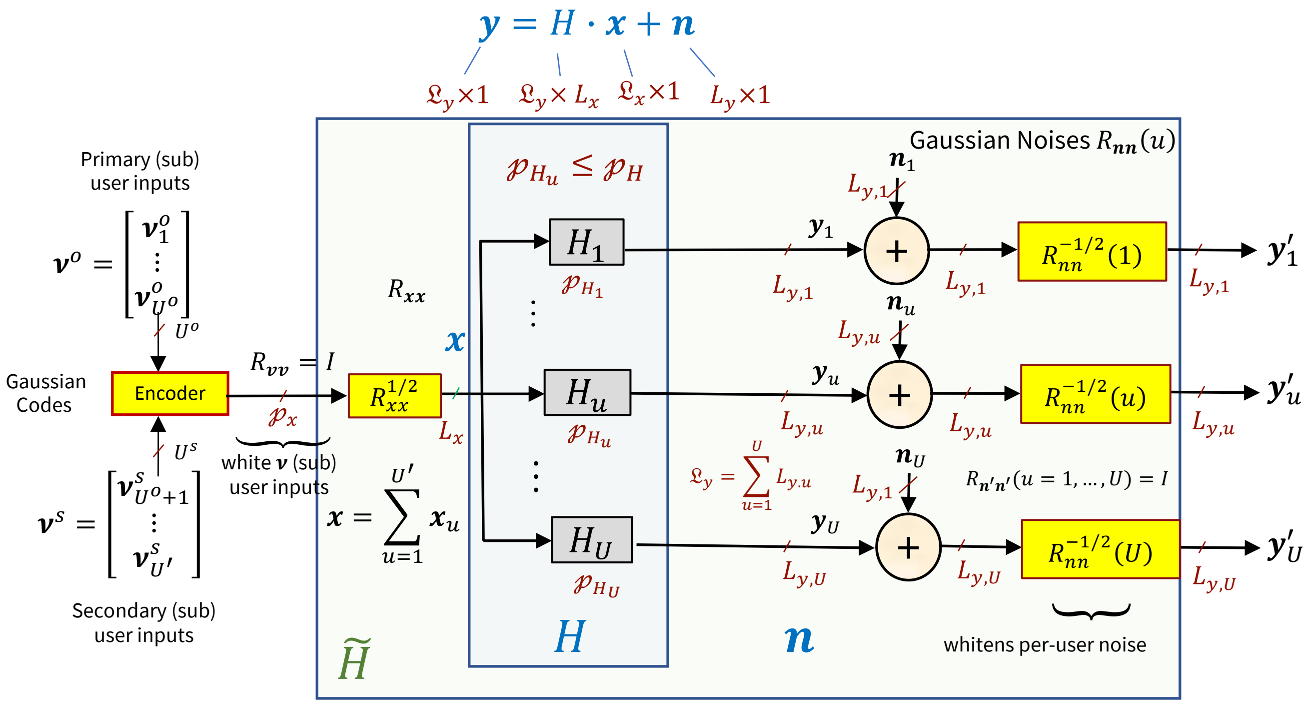
$\mathcal{E}$

- User 1 has highest sum rate when User 2 has zero energy
  - User 1 is a primary user/component
  - User 2 is a secondary user/component

Vector MMSE BC Design

Known $R_{xx}(u)$ Section 2.8.3.1

Vector Gaussian BC



- The users' independent message subsymbol vectors sum to a single BC input x

$$\# \text{ of subusers} = \boxed{U' \le \sum_{u=1}^{U} \min\left(\wp_x, \wp_{H_u}\right)}$$

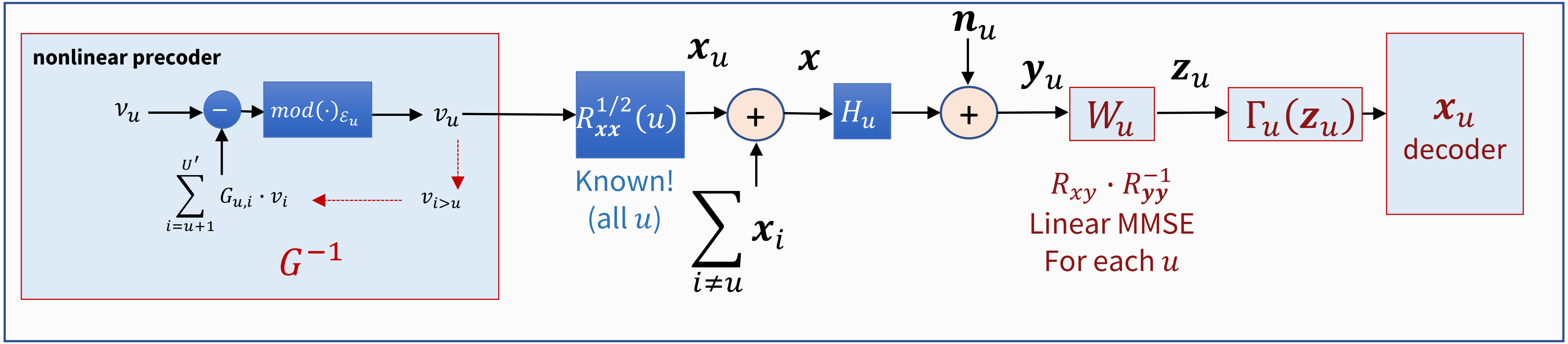
Modulator

$A = R_{xx}^{1/2}$

need not be square

MMSE – BC and Mutual Information – user u

- $\mathbb{I}(\boldsymbol{x}_u : \boldsymbol{y}_u / \boldsymbol{x}_{u+1,\ldots,U}) = \frac{1}{2} \cdot \log_2 \frac{|R_{\boldsymbol{xx}}(\boldsymbol{u})|}{|R_{\boldsymbol{ee}}(\boldsymbol{u})|}$ corresponds to a MMSE problem (like MAC, except $\boldsymbol{y}_u$).

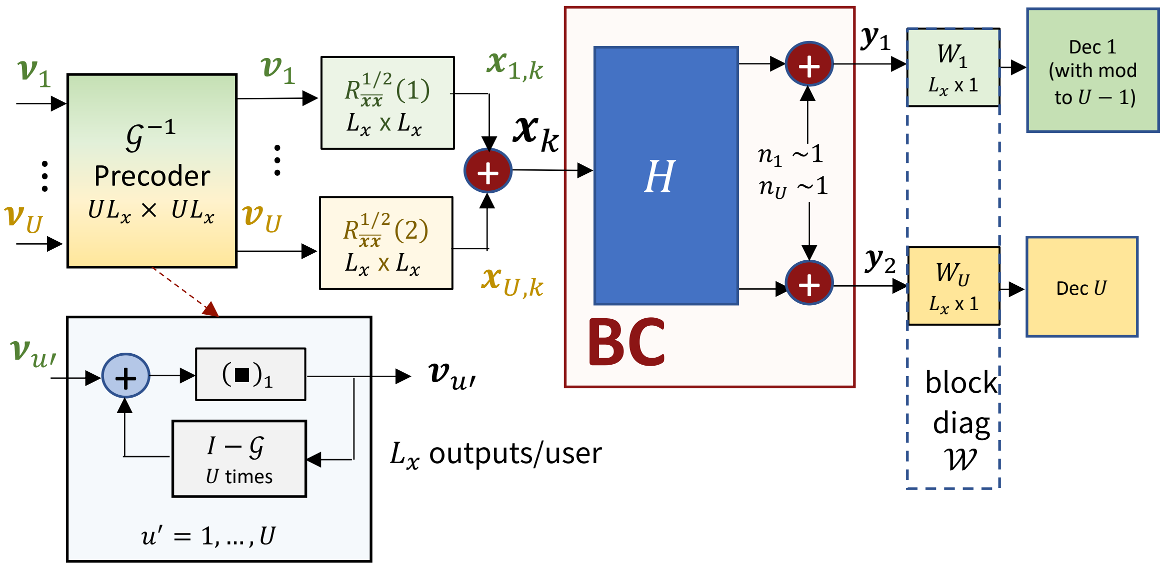


- There is successive-decoding ("GDFE") canonical performance (up to U' components).

- This structure reliably achieves highest rate for given input $R_{\boldsymbol{xx}}(u)$, and order $\boldsymbol{\pi}_u$.

- The catch? Designer must know $\{R_{\boldsymbol{xx}}(\boldsymbol{u})\}$ and order beforehand.

Structure for all user components $u \in U'$



- This structure needs a little more interpretation when channel rank < number of energized users.

The program mu_bc.m

```
function [Bu, GU, S0, MSWMFunb , B] = mu_bc(H, AU, Lyu , cb)
 --------------------------------------------------------------------------
  Inputs: Hu, AU , Usize, cb
  Outputs: Bu, Gunb, Wunb, S0, MSWMFunb

  H: noise-whitened BC matrix [H1 ; ... ; HU] (with actual noise, not wcn)
    sum-Ly x Lx x N
  AU: Block-row square-root discrete modulators, [A1 ... AU]
     Lx  x (U * Lx) x N
  Lyu: # of (output, Lyu) dimensions for each user U ... 1 in 1 x U row vector
  cb: = 1 if complex baseband or 2 if real baseband channel

  GU: unbiased precoder matrices: (Lx U) x (Lx U) x N
     For each of U users, this is Lx x Lx matrix on each tone
  S0: sub-channel dimensional channel SNRs: (Lx U) x (Lx U) x N
  MSWMFunb: users' unbiased diagonal mean-squared whitened matched matrices
    For each of U cells and Ntones, this is an Lx x Lyu  matrix
  Bu - users bits/symbol 1 x U
    the user should recompute SNR if there is a cyclic prefix
  B - the user bit distributions (U x N) in cell array
```

Set $N = 1$ (for now)

```
>> H =
   80
   50
>> Lyu=[1 1];
>> [Bu, Gunb, S0, MSWMFunb] = mu_bc(H, [1/sqrt(2) 1/sqrt(2)], Lyu, 2)
Bu =
   5.8222    0.4997
>> Gunb{:,:} =
   1.0000    1.0000
---
     0     1
S0 =
  2 x 1 cell array
   {[3.2010e+03]}     User 1 SNR
   {[    1.9992]}     User 2 SNR
MSWMFunb =
  2 x 1 cell array
   {[0.0177]}         multiplies y1
   {[0.0283]}         multiplies y2
>> sum(Bu) =    6.3219
>> [Bu, GU, S0, MSWMFunb , B] = mu_bc(H, [1 0], 1 , 2);
>> B =  1 × 1 cell array    {[6.3220]}
```

Equal energy on both users

Add user 2 to user 1 – we already knew this

Add nothing to user 2

Receivers are each simple scaling

- Same values as blue rate vector point on slide 16
- For this channel the rate sum is already close to maximum which occurs at $b = 6.3220$

More Examples

```
 H=[50 30
10 20];
>> A =
   0.5000        0   0.5000        0
        0   0.5000        0   0.5000
[Bu, Gunb, S0, MSWMFunb] = mu_bc(H, A, [1 1], 2);
Bu =
   4.8665    0.4971
>> Gunb{:,:} =
   1.0000    0.6000    1.0000    0.6000
        0    1.0000    1.6667    1.0000
  -----------------
        0         0    1.0000    2.0000
        0         0         0    1.0000
>> MSWMFunb{:,:} =
   0.0400
   0.0667
------
   0.2000
   0.1000
```

user 1's own xtalk and user 2 also

user 2's own xtalk

Each receiver estimates 2 input dimensions for its user, each a subuser.

```
>> >> S0{:,:} =
  626.0000         0
         0    1.3594
------------------
    1.1984         0
         0    1.6623

>> sum(Bu) =   5.3636
```

User 1's dimensional SNR's

User 2's dimensional SNR's

- mu_bc.m solves two MMSE problems here (for receiver 1 and receiver 2).
- It also aggregates them into right places in single matrix (cell array) of feedback/precoder, receiver filters.
- The receiver filters' rows apply to only their specific user/component (subuser) through MSWMFunb.

End Lecture 9