



STANFORD

Lecture 15

Duality and MAC-dual Basis

May 20, 2026

JOHN M. CIOFFI

Hitachi Professor Emeritus of Engineering

Instructor EE379B – Spring 2026

Announcements & Agenda

■ Announcements

- PS7 – last homework 6/4
- Section 5.5
- admMAC nominally works, but can run very long time
 - Use minPMAC if you experience this.

■ Agenda

- MAC/BC Duality Basics
 - Input deflection
 - Mappings
- Vector MAC/BC Duality
- MAC-dual Design

■ Problem Set 7 = PS7 (due 5/30. extend to 6/4)

1. 5.16 A tonal channel
2. 5.17 GDFE MAC Design
3. 5.18 Dual computations
4. 5.19 GDFE BC design via duality
5. 5.20 IC with/without GDFE

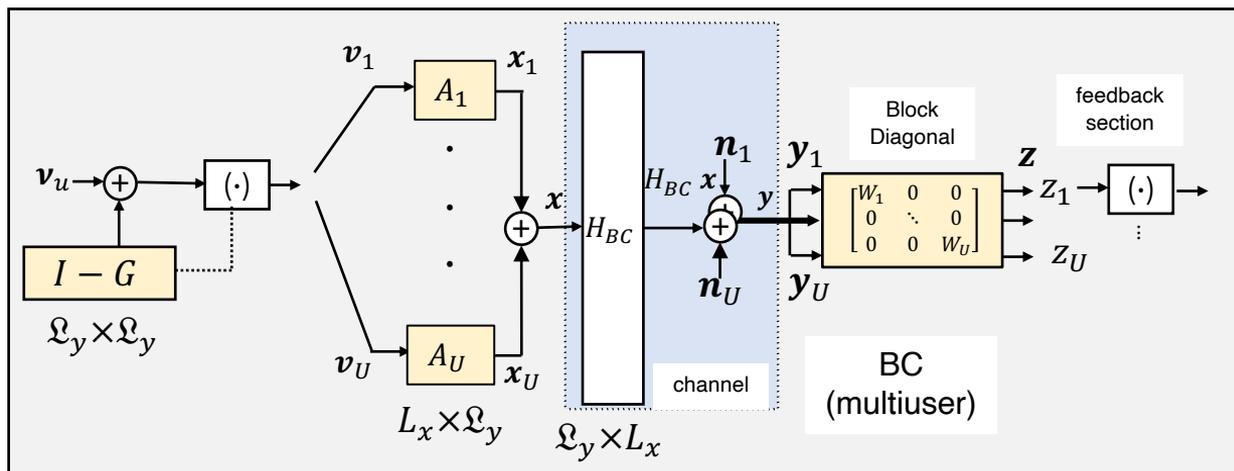


MAC / BC Duality Basics

Section 5.5

BC Input Addition & Design

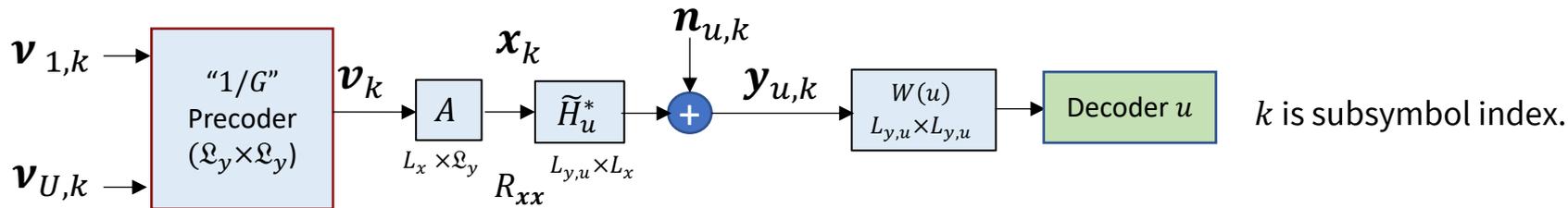
- The matrix-AWGN BC design adds user input symbols x_u before transmission.
 - $x = \sum_{u=1}^U x_u$ tends to “hide” independent input contributions.
 - Remember the “secondary-user component” precoder – “freeloading” compounds the hidden-subuser complexity – it is $\mathcal{L}_y \times \mathcal{L}_y$.
 - Primary users (or really any Q_H users) more productively separate.
 - $R_{xx} = \sum_{u=1}^U R_{xx}(u)$ - the contributions get “mixed” on the dimensions.
 - Key result: $R_b^{-1} = H_{BC}^* \cdot R_{nn}^{-1} \cdot H_{BC} + I = G \cdot S_0 \cdot G^*$.



- Design:
 - Generates a single precoder, with
 - no concern for primary / secondary,
 - which is MMSE based,
 - which finds A 's and G , (and W 's), &
 - derives from **dual** MAC's design.



GDFE per user (useful for BC), mu_bc.m



Up to L_y subuser components can affect decoder u (if $L_y > U^2$, design simplifies).

- Design achieves canonical performance for user u *for the given set of inputs and $\{\tilde{H}_u^*\}$ that is:*
 - *only a max rate sum if all- primary users with a corresponding special worst-case-noise-designed square root A .*
 - *But, this design is for specific $\{R_{xx}(u)\}$.*

- The receiver $W(u)$ is indeed MMSE for these inputs and \tilde{H}_u .

- Max rate is $\mathcal{I}_{BC}(u) = \log_2 \left(\frac{|I + \sum_{i=1}^u H_u^* \cdot R_{xx}(i) \cdot H_u|}{|I + \sum_{i=1}^{u-1} H_u^* \cdot R_{xx}(i) \cdot H_u|} \right)$ and corresponds to the individual- \tilde{H}_u MMSE GDFE.

- This is NOT a chain rule form because the H subscript is u , not i (but is a mini chain rule for rcvr u).
 - Thus, $\mathcal{I}(\mathbf{x}, \mathbf{y}) \neq \sum_{u=1}^U \mathcal{I}_{BC}(u)$; indeed $\mathcal{I}(\mathbf{x}, \mathbf{y}) \geq \sum_{u=1}^U \mathcal{I}_{BC}(u)$.



Order Reversal & The Dual

- Reverse MAC order so that BC has user 1 at top (still best position).
- \mathcal{J}_x applies to \mathbf{x} so that $\mathcal{J}_x \cdot \mathbf{x}$ reverses the input vector \mathbf{x} 's user order.

$$\mathcal{J}_x \triangleq \begin{bmatrix} 0 & 0 & I_{L_{x,1}} \\ 0 & \ddots & 0 \\ I_{L_{x,U}} & 0 & 0 \end{bmatrix}$$

- Thus, reversing a MAC's input order corresponds to $\tilde{H} \cdot \mathcal{J}_x$.
 - Can also have a \mathcal{J}_y on BC input side (but since BC has one joint input, less important).

$$H_{dual} = \mathcal{J}_x \cdot \tilde{H}^* \cdot \mathcal{J}_y$$

- The channel \tilde{H} 's dual is its order-reversed transpose.

- SVDs:

$$\tilde{H} = F \cdot \Lambda \cdot M^* \quad H_{dual} = \mathcal{J}_x \cdot M \cdot \Lambda \cdot F^* \cdot \mathcal{J}_y$$

Still an SVD (singular values the same); L_y **inputs**, L_x **outputs**



Specific to MAC and BC

- MAC Channel has normal notation:

$$\tilde{H}_{MAC} = [\tilde{H}_U \quad \cdots \quad \tilde{H}_1]$$

- Dual BC Channel transposes each user channel and reorders outputs and inputs so 1 is at top/left:

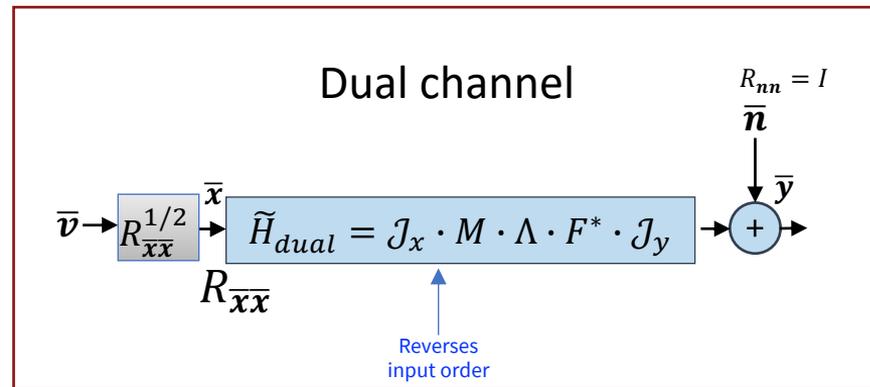
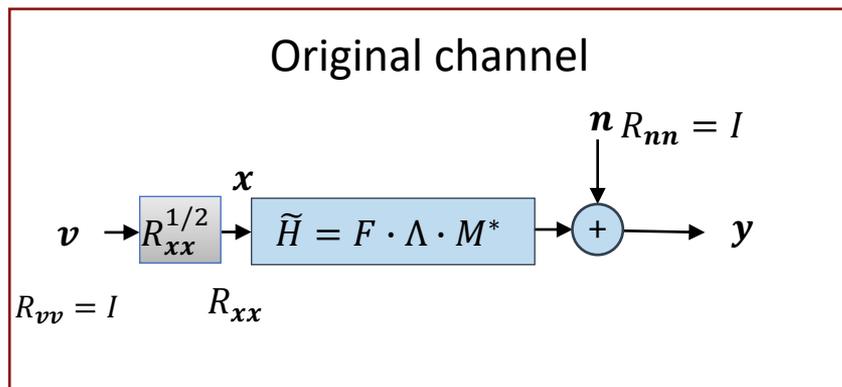
$$\tilde{H}_{BC} = \mathcal{J}_x \cdot \tilde{H}_{MAC}^* \cdot \mathcal{J}_y = \mathcal{J}_x \cdot \begin{bmatrix} \tilde{H}_U^* \\ \vdots \\ \tilde{H}_1^* \end{bmatrix} \cdot \mathcal{J}_y = \begin{bmatrix} \tilde{H}_1^* \\ \vdots \\ \tilde{H}_U^* \end{bmatrix} \cdot \mathcal{J}_y$$

- The reversal allows some simplification of notation (its worse without the reversal).



The (single-user & overall-channel) Dual

- Original and dual have the same mutual information equal:

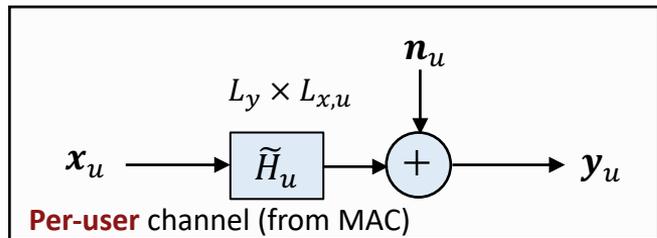


$$\mathcal{I}(\mathbf{x}, \mathbf{y}) = \log_2 |\tilde{\mathbf{H}} \cdot R_{xx} \cdot \tilde{\mathbf{H}}^* + I| = \log_2 |\mathbf{J}_x \cdot \tilde{\mathbf{H}}^* \cdot R_{\bar{x}\bar{x}} \cdot \tilde{\mathbf{H}} \cdot \mathbf{J}_x + I|$$

- A solution is $\bar{\mathbf{x}} = \mathbf{J}_y \cdot \mathbf{F} \cdot \mathbf{M}^* \cdot \mathbf{J}_x \cdot \mathbf{x}$, which yields
- $R_{\bar{x}\bar{x}} = \mathbf{J}_y \cdot \mathbf{F} \cdot \mathbf{M}^* \cdot \mathbf{J}_x \cdot R_{xx} \cdot \mathbf{J}_x \cdot \mathbf{M} \cdot \mathbf{F}^* \cdot \mathbf{J}_y$
The $\mathbf{J}_{x,y}$'s don't change rate sums/determinants
- SNR's also equal because $R_{vv} = I = R_{\bar{v}\bar{v}}$



Each user's dual with crosstalk as noise:

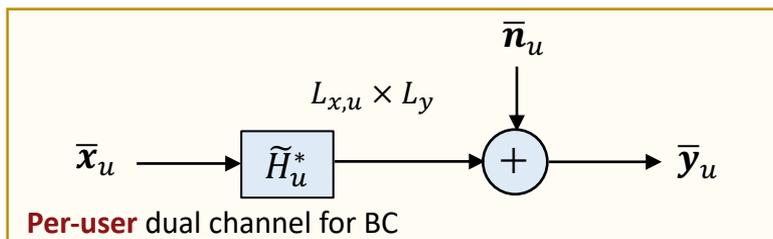


$$\tilde{H} = R_{nn}^{-1/2} \cdot H = [\tilde{H}_U \quad \dots \quad \tilde{H}_1]$$

$$\mathcal{R}_{nn}(u) = I + \sum_{i=u+1}^U \tilde{H}_i \cdot R_{xx}(i) \cdot \tilde{H}_i^* \quad \text{implies a user order.}$$

$L_y \times L_y$

Noise + Xtalk not white



$$\mathcal{R}_{\bar{n}\bar{n}}(u) = I + \sum_{i=1}^{u-1} \tilde{H}_u^* \cdot R_{\bar{x}\bar{x}}(i) \cdot \tilde{H}_u \quad \text{reverses implied order.}$$

$L_x \times L_x$

- For equal individual-user mutual information:

$$\mathcal{I}_u = \frac{|\tilde{H}_u \cdot R_{xx}(u) \cdot \tilde{H}_u^* + \mathcal{R}_{nn}(u)|}{|\mathcal{R}_{nn}(u)|} = \frac{|\tilde{H}_u^* \cdot R_{\bar{x}\bar{x}}(u) \cdot \tilde{H}_u + \mathcal{R}_{\bar{n}\bar{n}}(u)|}{|\mathcal{R}_{\bar{n}\bar{n}}(u)|}$$

**Suggests an input adjustment
(so dimensionality is consistent).**

- Duality design follows MAC's independent (x_u and $x_{i \neq u}$) & causes BC's (\bar{x}_u and $\bar{x}_{i \neq u}$) 's to be independent.

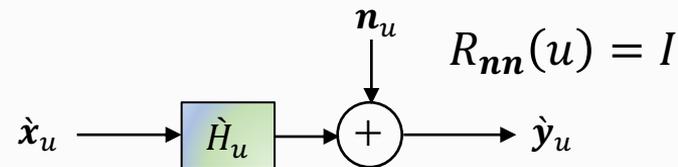


Input Deflection

- Duality **deflects input** so that it offsets the channel shaping (same dim as “dual’s xtalk):

$$\dot{\mathbf{x}}_u = \mathcal{R}_{\bar{\mathbf{n}}\bar{\mathbf{n}}}^{1/2}(u) \cdot \mathbf{x}_u$$

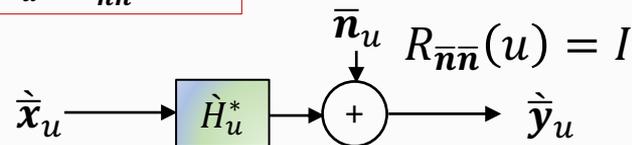
$$\dot{\bar{\mathbf{x}}}_u = \mathcal{R}_{\mathbf{n}\mathbf{n}}^{*/2}(u) \cdot \bar{\mathbf{x}}_u$$



$$R_{\mathbf{x}\mathbf{x}}(u) = \mathcal{R}_{\bar{\mathbf{n}}\bar{\mathbf{n}}}^{-1/2}(u) \cdot R_{\dot{\mathbf{x}}\dot{\mathbf{x}}}(u) \cdot \mathcal{R}_{\bar{\mathbf{n}}\bar{\mathbf{n}}}^{-*/2}(u)$$

$$\dot{H}_u = \mathcal{R}_{\bar{\mathbf{n}}\bar{\mathbf{n}}}^{-1/2}(u) \cdot \tilde{H}_u \cdot \mathcal{R}_{\bar{\mathbf{n}}\bar{\mathbf{n}}}^{-*/2}(u)$$

$$R_{\bar{\mathbf{x}}\bar{\mathbf{x}}}(u) = \mathcal{R}_{\mathbf{n}\mathbf{n}}^{-*/2}(u) \cdot R_{\dot{\bar{\mathbf{x}}}\dot{\bar{\mathbf{x}}}}(u) \cdot \mathcal{R}_{\mathbf{n}\mathbf{n}}^{-1/2}(u)$$



- This then **whitens both channels' noises** and is 1-to-1 on input deflection, so $\mathcal{I}(\dot{\mathbf{x}}_u; \dot{\mathbf{y}}_u) = \mathcal{I}(\dot{\bar{\mathbf{x}}}_u; \dot{\bar{\mathbf{y}}}_u) = \mathcal{I}(\mathbf{x}_u; \mathbf{y}_u)$

$$\begin{aligned} 2^{\mathcal{I}_u} &= \frac{|\tilde{H}_u \cdot R_{\mathbf{x}\mathbf{x}}(u) \cdot \tilde{H}_u^* + \mathcal{R}_{\mathbf{n}\mathbf{n}}(u)|}{|\mathcal{R}_{\mathbf{n}\mathbf{n}}(u)|} = \frac{|\tilde{H}_u^* \cdot R_{\bar{\mathbf{x}}\bar{\mathbf{x}}}(u) \cdot \tilde{H}_u + \mathcal{R}_{\bar{\mathbf{n}}\bar{\mathbf{n}}}(u)|}{|\mathcal{R}_{\bar{\mathbf{n}}\bar{\mathbf{n}}}(u)|} \\ &= \frac{|\dot{H}_u \cdot R_{\dot{\mathbf{x}}\dot{\mathbf{x}}}(u) \cdot \dot{H}_u^* + I|}{|I|} = \frac{|\dot{H}_u^* \cdot R_{\dot{\bar{\mathbf{x}}}\dot{\bar{\mathbf{x}}}}(u) \cdot \dot{H}_u + I|}{|I|} \end{aligned}$$



Scalar duality revisit and example

- Rewrite the scalar-duality input-deflection equations (follow from L10 scalar-energy duality):

$$\begin{aligned}
 x_1^{BC} \cdot \sqrt{1 + \varepsilon_2^{MAC} \cdot g_2 + \dots + \varepsilon_U^{MAC} \cdot g_U} &= x_1^{MAC} \\
 x_2^{BC} \cdot \sqrt{1 + \varepsilon_3^{MAC} \cdot g_3 + \dots + \varepsilon_U^{MAC} \cdot g_U} &= x_2^{MAC} \cdot \sqrt{1 + \varepsilon_1^{BC} \cdot g_2} \\
 &\vdots = \vdots \\
 x_U^{BC} &= x_U^{MAC} \cdot \sqrt{(1 + [\varepsilon_1^{BC} + \dots + \varepsilon_{U-1}^{BC}] \cdot g_U)}
 \end{aligned}$$

$\mathcal{R}_{nn}^{-1/2}(u)$

- Recognize these as a simple form of input deflection.
- $g_u = \frac{|h_u|^2}{\sigma_u^2}$, above equations do not depend on (MAC's) g_1 , which is BC's g_2 .

- L9's duality example was for $H_{MAC} = \begin{bmatrix} 50 & 80 \\ \text{user 2} & \text{user 1} \end{bmatrix}$

$$H_{BC} = \begin{bmatrix} 80 \\ 50 \end{bmatrix} \begin{matrix} \text{user 1} \\ \text{user 2} \end{matrix}$$

$$\varepsilon^{BC} = \begin{bmatrix} 3/4 \\ 1/4 \end{bmatrix} \rightarrow \begin{bmatrix} 1/7504 \\ 7503/7504 \end{bmatrix} = \varepsilon^{MAC}$$



BC Loss

- BC Loss - ratio of single-user capacity SNR to BC maximum-rate-sum SNR (for $[H \ R_{nn}]$):

$$\gamma_{BC} \triangleq \frac{2^{2 \cdot \bar{c}} - 1}{2^{2 \cdot \bar{c}_{E\text{-sum,dual-MAC}}} - 1} = \gamma_{E\text{-sum,dual-MAC}} \geq 1$$

- This equality is assured by duality.
- For slide L15:12's example:

$$\gamma_{BC} = \frac{2^{2 \cdot 6.556} - 1}{2^{2 \cdot 6.322} - 1} = 1.5 \text{ dB.}$$



Try a different input on each dual

$$\boldsymbol{\varepsilon}^{MAC} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \rightarrow \begin{bmatrix} 1/2502 \\ 2501/2502 \end{bmatrix} = \boldsymbol{\varepsilon}^{BC}$$

special case with scalar and $U = 2$

- Direct Design with BC

```
>> Hmac=[50 80];  
>> Rxx=diag([1/2 1/2]);  
>> A=sqrtm(Rxx);  
>> [Bu, GU, WU, S0, MSWMFU] = mu_mac(Hmac, A, [1 1], 2)
```

```
Bu =  
 5.1444  0.9155
```

```
GU =  
 1.0000  1.6000  
 0  1.0000
```

```
S0 = 1.0e+03 *  
 1.2510  0  
 0  0.0036
```

```
MSWMFU =  
 0.0283  
 0.0177
```

```
>> MSWMFU*Hmac*A =  
 1.0000  1.6000  
 0.6250  1.0000
```

```
>> sum(Bu) = 6.0600
```

```
>> Hbc=[80 ; 50];  
>> AU=[sqrt(1/2502) sqrt(2501/2502)];  
>> Lyu=[1 1]; cb=2;  
>> [Bu, GU, S0, MSWMFunb , B] = mu_bc(Hbc, AU, Lyu , cb)
```

```
Bu =  
 0.9155  5.1444
```

```
>> GU(:, :) =  
 1.0000  50.0100  
 0  1
```

```
>> diag(cell2mat(S0)) = 1.0e+03 *  
 0.0036  0  
 0  1.2510
```

```
>> MSWMFunb(:, :) =  
 0.6252  
 0.0200
```

```
>> diag(cell2mat(MSWMFunb))*Hbc*AU =  
 1.0000  50.0100  
 0.0200  1.0000
```

```
>> sum(Bu) = 6.0600
```

Data rates & SNRs
reverse in order.

Filters are not same.

Crosstalk cancels.

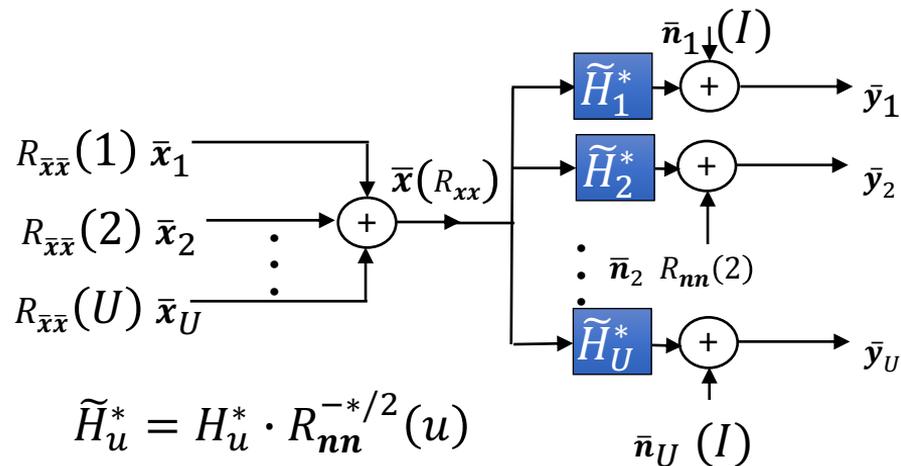
6.06 < 6.322 = previous rate sum because different input energy of [0.5 0.5] in this sum.



Vector MAC / BC Duality

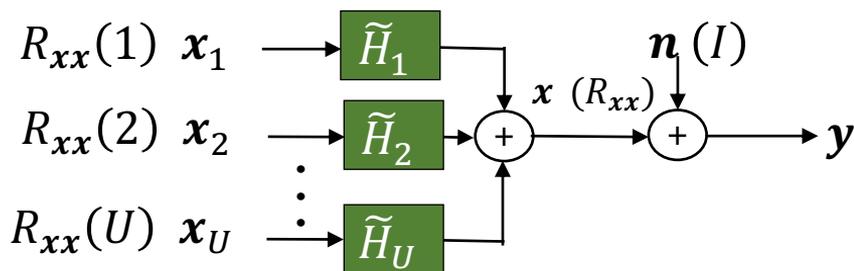
Section 5.5.2

Vector MAC/BC Duals



Broadcast - input is \bar{x}

$$R_{nn}(u+1) = \sum_{i=1}^u \tilde{H}_i^* \cdot R_{xx}(i) \cdot \tilde{H}_i + I$$



Multiple Access input is x

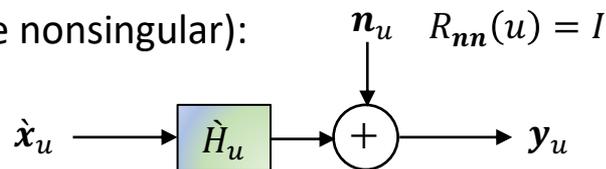
Order reversed

$$R_{nn}(u-1) = \sum_{i=u}^U \tilde{H}_i \cdot R_{xx}(i) \cdot \tilde{H}_i^* + I$$

- $\text{Esum-MAC } \varepsilon_x = \sum_{u=1}^U \varepsilon_u = \sum_{u=1}^U \text{trace}\{R_{xx}(u)\} = \sum_{u=1}^U \text{trace}\{R_{\bar{x}\bar{x}}(u)\}$

Summary: Input Deflection – 2 new channels

- Deflect input to offset the channel shaping (deflectors are nonsingular):



- $\dot{\mathbf{x}}_u = \mathcal{R}_{\mathbf{nn}}^{1/2}(u) \cdot \mathbf{x}_u,$

- $\dot{\bar{\mathbf{x}}}_u = \mathcal{R}_{\mathbf{nn}}^{*/2}(u) \cdot \bar{\mathbf{x}}_u.$

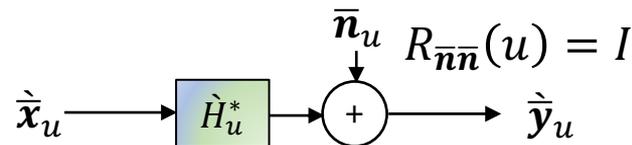
- Also, re-whiten the noise:

- $\dot{H}_u = \mathcal{R}_{\mathbf{nn}}^{-1/2}(u) \cdot \tilde{H}_u \cdot \mathcal{R}_{\mathbf{nn}}^{-1/2}(u).$

- Relate autocorrelation matrices

- $R_{\mathbf{xx}}(u) = \mathcal{R}_{\mathbf{nn}}^{-1/2}(u) \cdot R_{\dot{\mathbf{x}}\dot{\mathbf{x}}}(u) \cdot \mathcal{R}_{\mathbf{nn}}^{-*/2}(u),$

- $R_{\bar{\mathbf{x}}\bar{\mathbf{x}}}(u) = \mathcal{R}_{\mathbf{nn}}^{-*/2}(u) \cdot R_{\dot{\bar{\mathbf{x}}}\dot{\bar{\mathbf{x}}}}(u) \cdot \mathcal{R}_{\mathbf{nn}}^{-1/2}(u).$



- Whitens both channels' distortion and is 1-to-1 on input deflection, so $\mathcal{I}(\dot{\mathbf{x}}_u; \mathbf{v}_u) = \mathcal{I}(\dot{\bar{\mathbf{x}}}_u; \dot{\bar{\mathbf{y}}}_u) = \mathcal{I}(\mathbf{x}_u; \mathbf{y}_u).$

$$2^{\mathcal{I}_u} = \frac{|\tilde{H}_u \cdot R_{\mathbf{xx}}(u) \cdot \tilde{H}_u^* + \mathcal{R}_{\mathbf{nn}}(u)|}{|\mathcal{R}_{\mathbf{nn}}(u)|} = \frac{|\tilde{H}_u^* \cdot R_{\bar{\mathbf{x}}\bar{\mathbf{x}}}(u) \cdot \tilde{H}_u + \mathcal{R}_{\bar{\mathbf{n}}\bar{\mathbf{n}}}(u)|}{|\mathcal{R}_{\bar{\mathbf{n}}\bar{\mathbf{n}}}(u)|}$$

$$= \left| \dot{H}_u \cdot R_{\dot{\mathbf{x}}\dot{\mathbf{x}}}(u) \cdot \dot{H}_u^* + I \right| = \left| \dot{H}_u^* \cdot R_{\dot{\bar{\mathbf{x}}}\dot{\bar{\mathbf{x}}}}(u) \cdot \dot{H}_u + I \right|$$



Finish Vector Duality

- Write equality for deflected with actual autocorrelation matrices explicitly:

$$\left| \dot{H}_u \cdot \mathcal{R}_{\bar{n}\bar{n}}^{*/2}(u) \cdot R_{xx}(u) \cdot \mathcal{R}_{\bar{n}\bar{n}}^{1/2}(u) \cdot \dot{H}_u^* + I \right| = \left| \dot{H}_u^* \cdot \mathcal{R}_{\bar{n}\bar{n}}^{*/2}(u) \cdot R_{\bar{x}\bar{x}}(u) \cdot \mathcal{R}_{\bar{n}\bar{n}}^{1/2}(u) \cdot \dot{H}_u + I \right|$$

- SVD: $F_u \cdot \Lambda_u \cdot M_u^* = \text{svd}(\dot{H}_u)$
 - use “economy mode SVD” so that F_u and M_u may be non-square and the multiplication $F_u \cdot M_u^*$ is dimensionally ok.

- Inside determinant above is $F_u \cdot \Lambda_u \cdot M_u^* \cdot \mathcal{R}_{\bar{n}\bar{n}}^{*/2} \cdot R_{xx}(u) \cdot \mathcal{R}_{\bar{n}\bar{n}}^{1/2} \cdot M_u \cdot \Lambda_u \cdot F_u^* + I$.

- Pre/post multiply by F 's causes no change (nor by M 's in 2nd determinant).

- Data Rate Equality occurs if $R_{\hat{x}\hat{x}}(u) = M_u \cdot F_u^* \cdot R_{\bar{x}\bar{x}}(u) \cdot F_u \cdot M_u^*$

- MAC2BC:** $R_{\bar{x}\bar{x}}(u) = \mathcal{R}_{\bar{n}\bar{n}}^{-1/2}(u) \cdot F_u \cdot M_u^* \cdot \mathcal{R}_{\bar{n}\bar{n}}^{-*/2}(u) \cdot R_{xx}(u) \cdot \mathcal{R}_{\bar{n}\bar{n}}^{-1/2}(u) \cdot M_u \cdot F_u^* \cdot \mathcal{R}_{\bar{n}\bar{n}}^{-*/2}(u)$

- BC2MAC:** $R_{xx}(u) = \mathcal{R}_{\bar{n}\bar{n}}^{-*/2}(u) \cdot M_u \cdot F_u^* \cdot \mathcal{R}_{\bar{n}\bar{n}}^{-1/2}(u) \cdot R_{\bar{x}\bar{x}}(u) \cdot \mathcal{R}_{\bar{n}\bar{n}}^{-*/2}(u) \cdot F_u \cdot M_u^* \cdot \mathcal{R}_{\bar{n}\bar{n}}^{-1/2}(u)$.



MAC2BC Full Algorithm

$R_{xx}(u)$ from minPMAC

Given:

$$R_{xx}(u) \text{ for } u = 1, \dots, U ; R_{\bar{x}\bar{x}}(u) = R_{\bar{x}\bar{x}} = 0$$

$$\mathcal{R}_{nn}(U) = \mathcal{R}_{\bar{n}\bar{n}}(1) = I$$

$$\text{BC: } \tilde{H}_u^*, R_{nn}(u)$$

$$\text{MAC: } \tilde{H}_u = R_{nn}^{-1/2}(u) \cdot H_u$$

$$\text{For } u = U, \dots, 2 ; \mathcal{R}_{nn}(u-1) = \mathcal{R}_{nn}(u) + \tilde{H}_u \cdot R_{xx}(u) \cdot \tilde{H}_u^*$$

For $u = 1, \dots, U$

$$\dot{H}_u = \mathcal{R}_{nn}^{-1/2}(u) \cdot \tilde{H}_u \cdot \mathcal{R}_{\bar{n}\bar{n}}^{-1/2}(u)$$

$$F_u \cdot \Lambda_u \cdot M_u^* = \text{svd}(\dot{H}_u)$$

$$R_{\bar{x}\bar{x}}(u) = \mathcal{R}_{nn}^{-1/2}(u) \cdot F_u \cdot M_u^* \cdot \mathcal{R}_{\bar{n}\bar{n}}^{*/2}(u) \cdot R_{xx}(u) \cdot \mathcal{R}_{\bar{n}\bar{n}}^{1/2}(u) \cdot M_u \cdot F_u^* \cdot \mathcal{R}_{nn}^{-*/2}(u)$$

$$R_{\bar{x}\bar{x}} = R_{\bar{x}\bar{x}} + R_{\bar{x}\bar{x}}(u)$$

$$\mathcal{R}_{\bar{n}\bar{n}}(u+1) = \mathcal{R}_{\bar{n}\bar{n}}(u) + \tilde{H}_u^* \cdot R_{\bar{x}\bar{x}} \cdot \tilde{H}_u ; \text{ skip } u = U$$

- So, find the $R_{xx}(u)$ for the dual-BC's original MAC that has necessary data rate/energy.



BC2MAC Full Algorithm

Given:

$$R_{\bar{x}\bar{x}}(u) \text{ for } u = 1, \dots, U ; \mathcal{R}_{nn}(u) = \mathcal{R}_{\bar{n}\bar{n}}(u) = 0$$

$$\mathcal{R}_{nn}(U) = \mathcal{R}_{\bar{n}\bar{n}}(1) = I ; R_{\bar{x}\bar{x}} = 0$$

$$\text{BC: } \tilde{H}_u^*, R_{nn}(u)$$

$$\text{MAC: } \tilde{H}_u = R_{nn}^{-1/2}(u) \cdot H_u$$

For $u = 1, \dots, U - 1$;

$$R_{\bar{x}\bar{x}} = R_{\bar{x}\bar{x}} + R_{\bar{x}\bar{x}}(u)$$

$$\mathcal{R}_{\bar{n}\bar{n}}(u + 1) = \mathcal{R}_{\bar{n}\bar{n}}(u) + \tilde{H}_u^* \cdot R_{\bar{x}\bar{x}} \cdot \tilde{H}_u$$

For $u = U, \dots, 1$

$$\hat{H}_u = \mathcal{R}_{nn}^{-1/2}(u) \cdot \tilde{H}_u \cdot \mathcal{R}_{\bar{n}\bar{n}}^{-1/2}(u)$$

$$F_u \cdot \Lambda_u \cdot M_u^* = \text{svd}(\hat{H}_u)$$

$$R_{xx}(u) = \mathcal{R}_{\bar{n}\bar{n}}^{-*/2}(u) \cdot M_u F_u^* \cdot \mathcal{R}_{nn}^{1/2}(u) \cdot R_{\bar{x}\bar{x}}(u) \cdot \mathcal{R}_{nn}^{*/2}(u) \cdot F_u M_u^* \cdot \mathcal{R}_{\bar{n}\bar{n}}^{-1/2}(u)$$

$$\mathcal{R}_{nn}(u - 1) = \mathcal{R}_{nn}(u) + \tilde{H}_u \cdot R_{xx}(u) \cdot \tilde{H}_u^* ; \text{ skip } u = 1$$

- Reverse is less interesting, but provided for completeness.



Duality conversion program (Lx constant)

- Duality design uses the `Rxxb = mac2bc(Rxxm , Hmac)` program:
 - The input `Rxxm` is $L_x \times L_x \times U$ where L_x is for the MAC.
 - The input `Hmac` is $L_y \times L_x \times U$ where L_y AND `Hmac` are for the MAC
 - `Hmac(:,1)` is on the left and so equivalent to \tilde{H}_U - so then `Hmac(:,U)` is on the right
 - The output `Rxxb` is $L_y \times L_y \times U$ for the BC.
- With appropriate tensors, this I/O set can be repeated for each tone $n = 1, \dots, \bar{N}$.

```
Rxxm=zeros(1,1,2);
Rxxm(1,1,1)=1/7504;
Rxxm(1,1,2)=7503/7504;
Hmac=zeros(1,1,2);
Hmac(1,1,1)=50;
Hmac(1,1,2)=80;
0.5*log2(det([50 80]*diag([1/7504 7503/7504])*[50 80]'+1)) = 6.3220
```

Matlab order
is reverse of MAC
(same as BC).

```
Rxxb=mac2bc(Rxxm,Hmac)
Rxxb(:,1) = 0.7500
Rxxb(:,2) = 0.2500
```

```
Rxxm(1,1,1)=1/2;
Rxxm(1,1,2)=1/2;
Rxxb=mac2bc(Rxxm,Hmac)
Rxxb(:,1) = .9998
Rxxb(:,2) = 1.5620e-04
```

```
>> [Rwcn , bsum]=wcnnoise(1, [80 ; 50], 1)
Rwcn =
    1.0000    0.6250
    0.6250    1.0000
bsum = 6.3220
```

Note Hmac order corresponds to examples on slide 12
 $b_2 = .2074 \quad b_1 = 6.116$

- What if variable $L_{x,u}$, and so then variable $L_{y,u}$, on the dual?
- Set $L_x = \max_u L_{x,u}$ and append $L_x - L_{x,u}$ zero columns to each H_u and columns/rows of each $R_{xx}(u)$.
- There will be corresponding zeroed columns/rows on $R_{xx}(u)$ outputs.
- Duality algorithm's inverted/square-root matrices remain nonsingular.

Possible extra credit project
Variable- L_{xu} `mac2bc` / `bc2mac`



Reversal bc2mac program (Lx constant)

- $R_{xxm} = \text{bc2mac}(R_{xxb}, H_{mac})$ program has:
 - The input R_{xxb} is $L_x \times L_x \times U$ where L_x is for the dual MAC.
 - The input H_{mac} is $L_y \times L_x \times U$ is for the dual MAC (not the BC).
 - To reverse from mac2bc, input the mac2bc output (R_{xxb}) with $H_{bc} = \text{conj}(\text{permute}(H_{vec}(:,:,end:-1:1), [\text{order}' 3]))$
 - The output R_{xxm} is $L_x \times L_x \times U$ for the dual MAC.
- With appropriate tensors, this I/O set can be repeated for each tone $n = 1, \dots, \bar{N}$.

```
Rxxm=zeros(1,1,2);
Rxxm(1,1,1)=1/7504; (1.3326e-04) user 2
Rxxm(1,1,2)=7503/7504; (0.9999) user 1
Hmac=zeros(1,1,2);
Hmac(1,1,1)=50;
Hmac(1,1,2)=80;
Rxxb=mac2bc(Rxxm,Hmac)
Rxxb(:,:,1) = 0.7500
Rxxb(:,:,2) = 0.2500
```

```
bc2mac(Rxxb, Hmac)
ans(:,:,1) = 1.3326e-04
ans(:,:,2) = 0.9999
Checks reverses to original.

bsum = 6.322 on this channel.
```

```
Rxxm(1,1,1)=1/2;
Rxxm(1,1,2)=1/2;
Rxxb=mac2bc(Rxxm,Hmac)
Rxxb(:,:,1) = 3.9968e-04 (1/2502_
Rxxb(:,:,2) = 0.9996 (2501/2502)
>> bc2mac(Rxxb,Hmac)
ans(:,:,1) = 0.5000
ans(:,:,2) = 0.5000
```

```
conj( permute( Hmac(:,:,end:-1:1) , [2 1 3] ) ) =
80
50

conj( permute( Hbc(:,:,end:-1:1) , [2 1 3] ) ) =
50
80
```

With constant L_x , and $N=1$, then the designer can find the H_{bc} by $H_{bc} = \text{conj}(\text{permute}(H_{mac}(:,:,end:-1:1), [2 1 3]))$
 Or reverse from BC to MAC with $H_{mac} = \text{conj}(\text{permute}(H_{bc}(:,:,end:-1:1), [2 1 3]))$

- This last reverse step is usually not necessary because the designer optimizes for the dual MAC.
- This R_{xxm} then leads through mac2bc to R_{xxb} to complete the BC's needed input for design.



Example with 2 dimensions/user

```
Rxxm=zeros(2,2,2);  
Rxxm(:,:,1)=eye(2);  
Rxxm(:,:,2)=[2 1  
1 2]  
Hmac=zeros(2,2,2);  
Hmac(:,:,1)=[80 70  
50 60];  
Hmac(:,:,2)=[80 -50  
40 -25]  
Rxxb=mac2bc(Rxxm,Hmac)
```

```
Rxxb(:,:,1) =
```

```
0.0036 -0.0050  
-0.0050 0.0074
```

```
Rxxb(:,:,2) =
```

```
2.7951 0.9843  
0.9843 3.1939
```

```
>> bc2mac(Rxxb, Hmac)  
ans(:,:,1) =  
1.0000 0.0000  
0.0000 1.0000  
  
ans(:,:,2) =  
2.0000 1.0000  
1.0000 2.0000 (checks)
```



64-tone 3-user channel dual

```
N=64;
nu=3;
h=cat(3,[1 0 .8 ; 0 1 1],[.9 -.3 0 ; .5 -1 -1],[0 .2 0 ; .4 -.63 0],[0 0 0 ; 0 .648 0])*10;
H = fft(h, N, 3);
Hbc=zeros(3,2,N);

Rxxm=zeros(1,1,3);
Rxxm(1,1,:)= N/(N+nu)*[1 1 1];
Rxxb=zeros(2,2,3,N);
bbc=zeros(3,N);
Hbc=zeros(3,2,N);
for n=1:N
Rxxb(:,:,n)=mac2bc(Rxxm, reshape(H(:,:,n),2,1,3)); % input needs to be Ly x Lxu x U
Hbc(:,:,n)=H(:,end:-1:1,n)';
bbc(1,n)=real(log2(1+Hbc(1,:,n)*Rxxb(:,:,1,n)*Hbc(1,:,n)'));
bbc(2,n)=real(log2((1+Hbc(2,:,n)*(Rxxb(:,:,2,n)+Rxxb(:,:,1,n))*Hbc(2,:,n))/(1+Hbc(2,:,n)*Rxxb(:,:,1,n)*Hbc(2,:,n))));
bbc(3,n)=real(log2((1+Hbc(3,:,n)*(Rxxb(:,:,3,n)+Rxxb(:,:,2,n)+Rxxb(:,:,1,n))*Hbc(3,:,n))/(1+Hbc(3,:,n)*(Rxxb(:,:,2,n)+Rxxb(:,:,1,n))*Hbc(3,:,n))));
end
bvec=sum(bbc') = 132.7477 412.8794 445.1264

>> bsum=sum(bvec) = 990.7535
```

- Equal energy used on MAC input here, so this example's dual and original MAC are not max rate sum.
 - But they are equal -- & close to max sum rate as well.
 - Note order reversal – user 1 is in best position on BC, but worst position in MAC – a feature of duality.



MAC-dual Design

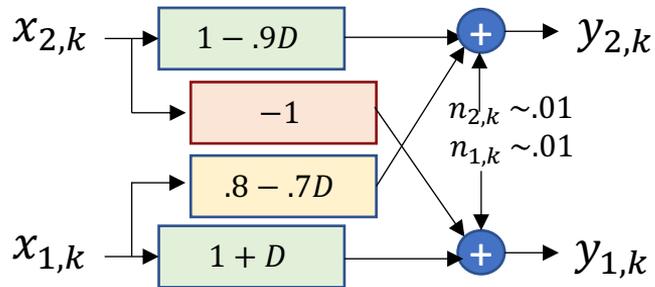
Section 5.5.4

Two-user channel with memory (Low/high pass)

$$\mathcal{E}_2 = 1$$

Per symbol

$$\mathcal{E}_1 = 1$$



$$H(D) = \begin{bmatrix} 1 - .9D & .8 - .7D \\ -1 & 1 + D \end{bmatrix} \text{ complex baseband}$$

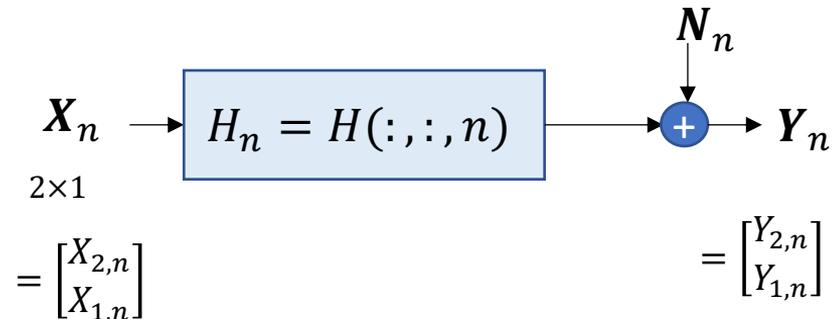
Use 2×2 Vector DMT

```
h = cat(3, [1 .8; -1 1], [-.9 -.7; 0 1])*10;
He = fft(h, 8, 3); % (the matlab FFT increases energy)
```

Cyclic prefix $\nu = 1$, energy loss 8/9

So far, just $H \rightarrow$

8 tonal 2×2 channels



>> H ■ Table

```
H(:, :, 1) =
 1.0000 + 0.0000i 1.0000 + 0.0000i
-10.0000 + 0.0000i 20.0000 + 0.0000i
H(:, :, 2) =
 3.6360 + 6.3640i 3.0503 + 4.9497i
-10.0000 + 0.0000i 17.0711 - 7.0711i
H(:, :, 3) =
 10.0000 + 9.0000i 8.0000 + 7.0000i
-10.0000 + 0.0000i 10.0000 -10.0000i
H(:, :, 4) =
 16.3640 + 6.3640i 12.9497 + 4.9497i
-10.0000 + 0.0000i 2.9289 - 7.0711i
```

```
H(:, :, 5) =
 19.0000 + 0.0000i 15.0000 + 0.0000i
-10.0000 + 0.0000i 0.0000 + 0.0000i
H(:, :, 6) =
 16.3640 - 6.3640i 12.9497 - 4.9497i
-10.0000 + 0.0000i 2.9289 + 7.0711i
H(:, :, 7) =
 10.0000 - 9.0000i 8.0000 - 7.0000i
-10.0000 + 0.0000i 10.0000 +10.0000i
H(:, :, 8) =
 3.6360 - 6.3640i 3.0503 - 4.9497i
-10.0000 + 0.0000i 17.0711 + 7.0711i
```



Single-User Upper Bound

- The highest data rate is the single-user capacity for the matrix AWGN – need SVD of big H.

```
Hblock=blkdiag(H(:,:,1),H(:,:,2),H(:,:,3),H(:,:,4),H(:,:,5),H(:,:,6),H(:,:,7),H(:,:,8));
g=svd(Hblock);
gains=(g.*g)' =
 651.4623 567.9619 567.9619 500.2007 447.4561 447.4561 405.2081 405.2081
188.7919 188.7919 91.0919 91.0919 81.4901 81.4901 34.5377 1.7993
```

- Water-fill on these singular-value-squared parallel channels ($\Gamma = 0$ dB):

```
>> En = waterfill(128/9, gains', 1);
>> En' =
0.9285 0.9283 0.9283 0.9280 0.9278 0.9278 0.9276 0.9276
0.9247 0.9247 0.9191 0.9191 0.9178 0.9178 0.9011 0.3743
Single-user waterfill is close to equal energy on all tones and spatial dimensions - very high SNR.

>> bvec=log2(ones(16,1)+En.*gains')
9.2429 9.0450 9.0450 8.8617 8.7010 8.7010 8.5579 8.5579
7.4560 7.4560 6.4046 6.4046 6.2439 6.2439 5.0055 0.7428

>> sumrate = sum(bvec) = 116.6695
>>sum(bvec)/9 = 12.9633
```



There are now 8 MMSE-MAC GDFEs, 1 for each tone

- Repetitive process for each n , initialize/size:

```
GU=zeros(2,2,8);  
WU=zeros(2,2,8);  
S0=zeros(2,2,8);  
Bu=zeros(2,8);  
MSWMFU=zeros(2,2,8);  
AU=zeros(2,2,8);  
for n=1:8 AU(:, :, n)=(sqrt(8)/3)*eye(2); end
```

Note the $\sqrt{8}/3$ on each of 8 dim's for each of 2 users.
So $1/3 = 1/\sqrt{9}$ for each A_u
a second factor of 8 matches unit-noise-whitening on 8 tones.

- Compute the MAC GDFE for each n ,

```
>> for n=1:8  
[Bu(:,n), GU(:, :, n), WU(:, :, n), S0(:, :, n), MSWMFU(:, :, n)] = mu_mac(H(:, :, n), AU(:, :, n), [1 1], 1);  
end  
bvec=sum(Bu') = 62.3515 54.1393  
Bsum =sum(bvec) = 116.4908
```

- Bits/user/tone

10.1778

```
Bu =  
6.5043 7.1048 7.9703 8.5075 8.6822 8.5075 7.9703 7.1048  
3.6736 7.7329 7.9256 6.8322 5.4843 6.8322 7.9256 7.7329  
sum(Bu) =  
10.1778 14.8377 15.8959 15.3396 14.1665 15.3396 15.8959 14.8377
```

bvec = 62.3515 54.1393

Bsum = 116.4908

so then $bsum/9 = 12.9434$ bits/tone or roughly 13 bits/Hz for both users

Why are single-user and MAC close?

(The flat high energy input on ALL dimensions means that SVD M might as well be I . Lots of square roots and one is I)



MAC Receiver Designs

- Unbiased total linear rcvr processing, each tone

```
MSWMFU
MSWMFU(:,:,1) =
    0.0105 + 0.0000i -0.1050 + 0.0000i
    0.2364 + 0.0000i  0.0412 + 0.0000i
MSWMFU(:,:,2) =
    0.0251 - 0.0439i -0.0690 - 0.0000i
    0.0397 - 0.0382i  0.0392 + 0.0116i
MSWMFU(:,:,3) =
    0.0377 - 0.0340i -0.0377 + 0.0000i
    0.0374 - 0.0084i  0.0449 + 0.0254i
MSWMFU(:,:,4) =
    0.0425 - 0.0165i -0.0260 + 0.0000i
    0.0456 + 0.0096i  0.0681 + 0.0449i
MSWMFU(:,:,5) =
    0.0437 + 0.0000i -0.0230 + 0.0000i
    0.0707 + 0.0000i  0.1329 + 0.0000i
MSWMFU(:,:,6) =
    0.0425 + 0.0165i -0.0260 + 0.0000i
    0.0456 - 0.0096i  0.0681 - 0.0449i
MSWMFU(:,:,7) =
    0.0377 + 0.0340i -0.0377 + 0.0000i
    0.0374 + 0.0084i  0.0449 - 0.0254i
MSWMFU(:,:,8) =
    0.0251 + 0.0439i -0.0690 + 0.0000i
    0.0397 + 0.0382i  0.0392 - 0.0116i
```

- Unbiased feedback sections for each tone

```
GU
GU(:,:,1) =
    1.0000 + 0.0000i -1.9703 + 0.0000i
    0.0000 + 0.0000i  1.0000 + 0.0000i
GU(:,:,2) =
    1.0000 + 0.0000i -0.8335 + 0.4508i
    0.0000 + 0.0000i  1.0000 + 0.0000i
GU(:,:,3) =
    1.0000 + 0.0000i  0.1530 + 0.3488i
    0.0000 + 0.0000i  1.0000 + 0.0000i
GU(:,:,4) =
    1.0000 + 0.0000i  0.5244 + 0.1697i
    0.0000 + 0.0000i  1.0000 + 0.0000i
GU(:,:,5) =
    1.0000 + 0.0000i  0.6182 + 0.0000i
    0.0000 + 0.0000i  1.0000 + 0.0000i
GU(:,:,6) =
    1.0000 + 0.0000i  0.5244 - 0.1697i
    0.0000 + 0.0000i  1.0000 + 0.0000i
GU(:,:,7) =
    1.0000 + 0.0000i  0.1530 - 0.3488i
    0.0000 + 0.0000i  1.0000 + 0.0000i
GU(:,:,8) =
    1.0000 + 0.0000i -0.8335 - 0.4508i
    0.0000 + 0.0000i  1.0000 + 0.0000i
```



What about the other vertex (puts user 1 at top)

- Design uses the same initialization because both users had same energy anyway

```
J=hankel([0 1]);
for n=1:8
Hflip(:,n)=J*H(:,n)*J;
end
>> for n=1:8
[Bu(:,n), GU(:,n), WU(:,n),S0(:,n), MSWMF(:,n)] = mu_mac(Hflip(:,n), AU(:,n), [1 1], 1);
end
Bu
sum(Bu)
bvec=sum(Bu')
bsum=sum(bvec)
```

- The user bit rates are useful in duality n ,

```
Bu =
 8.4816  8.3860  8.1253  7.8068  7.6511  7.8068  8.1253  8.3860
 1.6963  6.4517  7.7706  7.5328  6.5155  7.5328  7.7706  6.4517

sum(Bu) =
10.1778 14.8377 15.8959 15.3396 14.1665 15.3396 15.8959 14.8377

bvec = 64.7687 51.7220
bsum = 116.4908, so same (check)
```

10.1778

Rate sum is maintained on each tone, but decoding order is reversed



And the loss

- For this example, the data rate is already close to single-user WF.
- Answer for $E=[1 \ 1]$ using SWF:

- Maximum E_{sum} MAC rate sum:

```
Rnn=zeros(2,2,8);
for n=1:8
Rnn(:,:,n)=eye(2);
end
>> [Rxx, bsum , bsum_lin] = SWF((8/9)*[1 1], H, [1 1], Rnn, 1)
```

```
Rxx(:,:,1) =      Rxx(:,:,2) =      Rxx(:,:,3) =      Rxx(:,:,4) =
0.5500    0    0.9339    0    0.9401    0    0.9394    0
      0 0.8401    0 0.8991    0 0.8996    0 0.8954
Rxx(:,:,5) =      Rxx(:,:,6) =      Rxx(:,:,7) =      Rxx(:,:,8) =
0.9344    0    0.9394    0    0.9401    0    0.9339    0
      0 0.8829    0 0.8954    0 0.8996    0 0.8991
bsum = 116.5835
```

```
gamma_mac = 10*log10((2^(116.5835/9) -1)/(2^(116.4908/9)-1)) = 0.0310 dB
```

```
bsum_lin = 106.1991
```

```
Linear loss is
10*log10( (2^(116.4908/9)-1) / (2^(106.1991/9)-1) ) = 3.4430 dB
```

```
[Rxx, bsum, bsum_lin] = macmax(16/9, h, [1 1], 8, 1);
>> bsum = 116.5891 (so greater than Evec, but less than vector code 116.6695)

>> bsum_lin = 106.2249
```

- So, **116.5** is (also) best E_{sum} MAC rate sum (and also for Evec of $[1 \ 1]$ – pretty close already).



Order Reversal

- Semantics – alternate dual definition $\tilde{H}_{dual} = (\mathcal{J}_y \cdot \tilde{H} \cdot \mathcal{J}_x)^* = \mathcal{J}_x \cdot \tilde{H}^* \cdot \mathcal{J}_y$.

$$\mathcal{J}_y \triangleq \begin{bmatrix} 0 & 0 & I_{L_y, U} \\ 0 & \ddots & 0 \\ I_{L_y, 1} & 0 & 0 \end{bmatrix}$$

- All information, SVD, energy, etc are preserved as without \mathcal{J}_y :
 - Single output on MAC corresponds to single input on BC.
 - \mathcal{J}_y just re-indexes dimensions (not users) – but this is matlab’s usual indexing.
 - But the definition said reverse order (so can do it explicitly with \mathcal{J}_y).
- The mac2bc and bc2mac programs basically do this tacitly in finding inputs:
 - L16’s `Hbc=conj(permute(Hmac(:, :, end:-1:1) , [2 1 3]))` for $N = 1$ presumes the channel input has dimension 1 at top
 - So, this is essentially multiplying by \mathcal{J}_y **tacitly**.

- Example has:
$$H_{dual}(D) = \left(\mathcal{J}_y \cdot \underbrace{\begin{bmatrix} 1 - .9D & .8 - .7D \\ -1 & 1 + D \end{bmatrix}}_{H(D)} \cdot \mathcal{J}_x \right)^* = \begin{bmatrix} 1 + D^* & .8 - .7D^* \\ -1 & 1 - .9D^* \end{bmatrix}$$

$D^* \rightarrow e^{j\omega}$ on unit circle (FFT)

- Essentially dual here causes:
 - User direct (magnitudes, - phase) to be the same, but priority is reversed.
 - Crosstalk flow to flip from transfer of $u \rightarrow u'$ on original to $u \leftarrow u'$.





End Lecture 15