STANFORD

*Lecture 16*

# Duality and MAC-dual Basis

*May 31, 2023*

**JOHN M. CIOFFI**

Hitachi Professor Emeritus of Engineering

Instructor EE392AA – Spring 2023

# Announcements & Agenda

- Announcements
  - PS7 – last normal homework due formally 6/2, but take to 6/7
  - Projects – by now, should know what topic it is
  - Section 5.5
  - admMAC nominally works, but can run very long time
    - Use minPMAC with the adjusting w yourself (as per L15 examples) for now

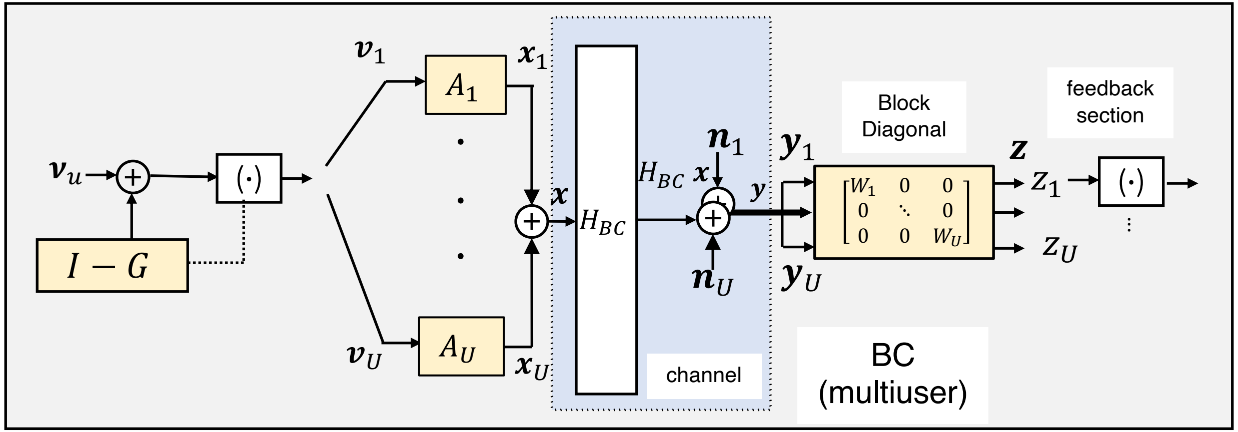
- Agenda
  - MAC/BC Duality Basics
    - Input deflection
    - Mappings
  - Vector MAC/BC Duality
  - MAC-dual Design

# MAC / BC Duality Basics

Section 5.5

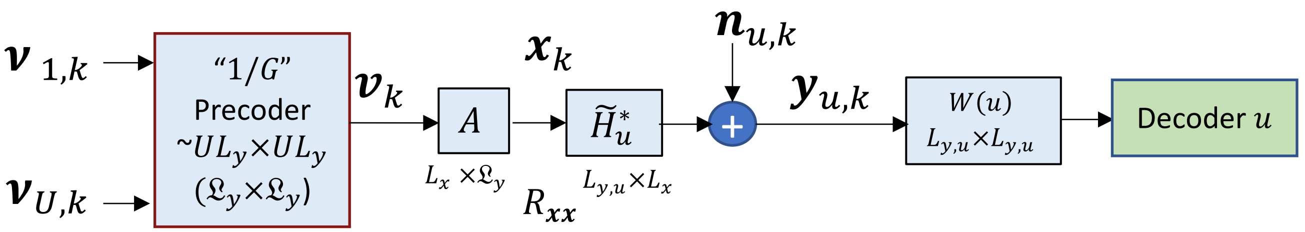
# BC Input Addition & Design

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




- Design
  - Would prefer single precoder
    - Not primary / secondary
  - MMSE based
  - How to find $A$'s and $G$? (and $W$'s)
  - Will come from **dual** MAC's GDFE(s)

# GDFE per user (useful for BC)




$k$ is symbol index

***Up to $\mathfrak{L}_y$ subuser components can affect decoder $u$ (if $\mathfrak{L}_y > U^2$, then still simplifies design)***

- Canonical performance for user $u$ *for the given set of inputs and $\tilde{H}_u^*$*
  - *Only a max rate sum if all- primary users with a corresponding special worst-case-noise-designed square root $A$.*
  - *But we now design for specific **b** anyway*
- The receiver $W(u)$ is indeed MMSE for these inputs and $\tilde{H}_u$
- The data rate is $\mathcal{I}_{BC}(u) = \log_2\left(\frac{\left|I+\sum_{i=1}^{u} H_u^* \cdot R_{\overline{xx}}(i) \cdot H_u\right|}{\left|I+\sum_{i=1}^{u-1} H_u^* \cdot R_{\overline{xx}}(i) \cdot H_u\right|}\right)$ and corresponds to the individual- $\tilde{H}_u$ MMSE GDFE
- This is NOT overall chain rule form because the $H$ subscript is $u$, not $i$ (mini chain rule for rcvr $u$)
  - Thus, $\mathcal{I} \neq \sum_{u=1}^{U} \mathcal{I}_{BC}(u)$ ; indeed $\mathcal{I} \geq \sum_{u=1}^{U} \mathcal{I}_{BC}(u)$ .

# Order Reversal & The Dual

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

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

- Thus, reversing a MAC's input order is $\tilde{H} \cdot \mathcal{J}_x$
- The channel $\tilde{H}$'s dual is its transpose with order reversed

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

- SVD? $\tilde{H} = F \cdot \Lambda \cdot M^*$

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

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

# Specific to MAC and BC

- MAC Channel – normal notation

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

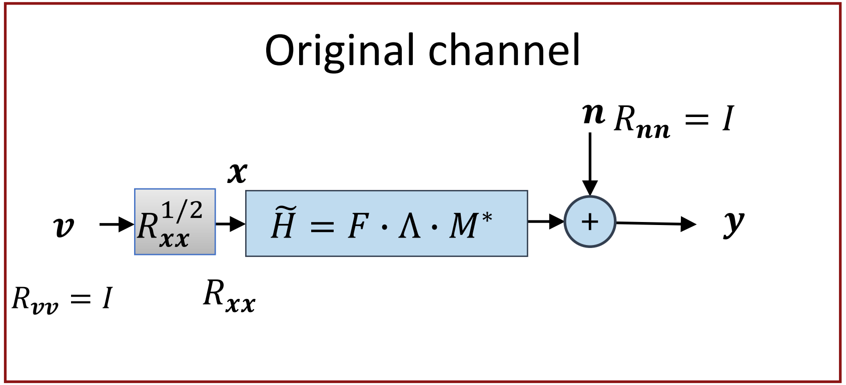
- Dual BC Channel – transpose of each user channel and reorders outputs and inputs so 1 is at top/left

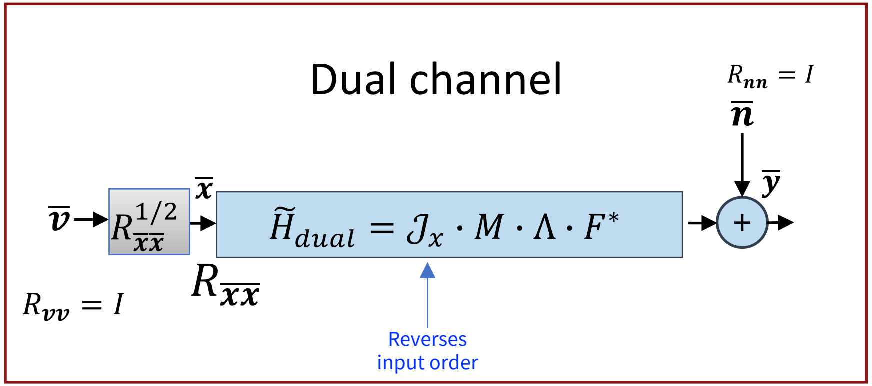
$$\tilde{H}_{BC} = \mathcal{J}_x \cdot \tilde{H}_{MAC}^* = \mathcal{J}_x \cdot \begin{bmatrix} \tilde{H}_U^* \\ \vdots \\ \tilde{H}_1^* \end{bmatrix} = \begin{bmatrix} \tilde{H}_1^* \\ \vdots \\ \tilde{H}_U^* \end{bmatrix}$$

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

# The (single-user or overall-channel) Duals

- Set mutual information equal? Then




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

- A solution $\quad \bar{\boldsymbol{x}} = F \cdot M^* \cdot \mathcal{J}_x \cdot \boldsymbol{x}$

$$R_{\bar{\boldsymbol{x}}\bar{\boldsymbol{x}}} = F \cdot M^* \cdot \mathcal{J}_x \cdot R_{\boldsymbol{xx}} \cdot \mathcal{J}_x \cdot M \cdot F^*$$

The $\mathcal{J}_x$'s don't change rate sums/determinants

- SNR's also equal because inputs are white

**Theorem 5.5.1** **[Dual Single-User Channels]** *The dual-squared channel $\tilde{H}$ based on an initial non-square channel $\tilde{H}_{init}$ as in Definition 5.5.1's (5.330), and its dual $H_{dual} = \mathcal{J}_y \cdot \tilde{H}^* \mathcal{J}_x$ in (5.335) with inputs related by (5.339) have the same:*

**mutual information**

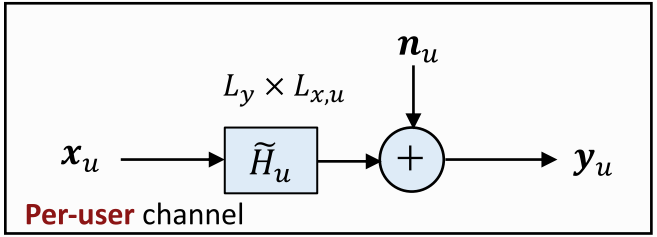
$$\mathcal{I}(\boldsymbol{x};\boldsymbol{y}) = \mathcal{I}(\bar{\boldsymbol{x}};\bar{\boldsymbol{y}}) \tag{5.341}$$

**input energy** , *and*

$$trace\{R_{\boldsymbol{xx}}\} = trace\{R_{\bar{\boldsymbol{x}}\bar{\boldsymbol{x}}}\} \quad . \tag{5.342}$$

Proof in Sec 5.5

# Rearrange the noise and crosstalk to create:

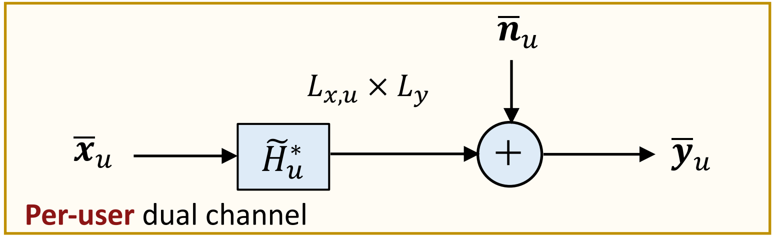



$$\mathcal{R}_{\boldsymbol{nn}}(u) = I + \sum_{i=u+1}^{U} \widetilde{H}_i \cdot R_{\boldsymbol{xx}}(i) \cdot \widetilde{H}_i^*$$

$L_y \times L_y$

**Noise + Xtalk not white**

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




$$\mathcal{R}_{\bar{\boldsymbol{n}}\bar{\boldsymbol{n}}}(u) = I + \sum_{i=1}^{u-1} \widetilde{H}_u^* \cdot R_{\bar{\boldsymbol{x}}\bar{\boldsymbol{x}}}(i) \cdot \widetilde{H}_u$$

$L_x \times L_x$

- For equal individual-user mutual information

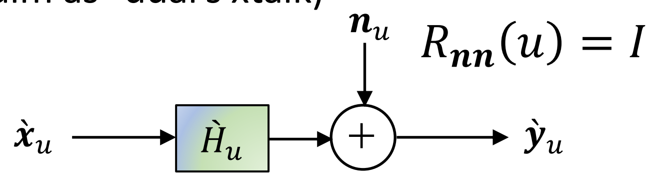
$$\mathcal{I}_u = \frac{\left|\widetilde{H}_u \cdot R_{\boldsymbol{xx}}(u) \cdot \widetilde{H}_u^* + \mathcal{R}_{\boldsymbol{nn}}(u)\right|}{|\mathcal{R}_{\boldsymbol{nn}}(u)|} = \frac{\left|\widetilde{H}_u^* \cdot R_{\bar{\boldsymbol{x}}\bar{\boldsymbol{x}}}(u) \cdot \widetilde{H}_u + \mathcal{R}_{\bar{\boldsymbol{n}}\bar{\boldsymbol{n}}}(u)\right|}{|\mathcal{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

**Suggests an input adjustment (dimensionaly consistent)**

- Duality design: follows MAC's independent ($\boldsymbol{x}_u$ and $\boldsymbol{x}_{i\neq u}$) to cause ($\bar{\boldsymbol{x}}_u$ and $\bar{\boldsymbol{x}}_{i\neq u}$) 's independence also

# Input Deflection

- Deflect input so that it offsets the channel shaping (same dim as "dual's xtalk)



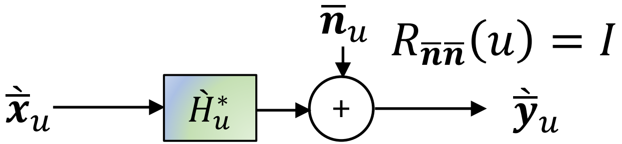

$$\grave{\boldsymbol{x}}_u = \mathcal{R}_{\boldsymbol{nn}}^{1/2}(u) \cdot \boldsymbol{x}_u$$

$$\grave{\bar{\boldsymbol{x}}}_u = \mathcal{R}_{\boldsymbol{nn}}^{*/2}(u) \cdot \bar{\boldsymbol{x}}_u$$

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

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

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




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

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

$$\triangleq \left|\grave{H}_u \cdot R_{\grave{\boldsymbol{x}}\grave{\boldsymbol{x}}}(u) \cdot \grave{H}_u^* + I\right| \triangleq \left|\grave{H}_u^* \cdot R_{\grave{\bar{\boldsymbol{x}}}\grave{\bar{\boldsymbol{x}}}}(u) \cdot \grave{H}_u + I\right|$$

# Scalar duality revist and example

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

$$
\begin{aligned}
x_1^{BC} \cdot \sqrt{1+\mathcal{E}_2^{MAC} \cdot g_2 + \ldots + \mathcal{E}_U^{MAC} \cdot g_U} &= x_1^{MAC} \\
x_2^{BC} \cdot \sqrt{1+\mathcal{E}_3^{MAC} \cdot g_3 + \ldots + \mathcal{E}_U^{MAC} \cdot g_U} &= x_2^{MAC} \cdot \sqrt{1+\mathcal{E}_1^{BC} \cdot g_2} \\
\vdots &= \vdots \\
x_U^{BC} &= x_U^{MAC} \cdot \sqrt{\left(1+\left[\mathcal{E}_1^{BC} + \ldots + \mathcal{E}_{U-1}^{BC}\right] \cdot g_U\right)}
\end{aligned}
$$

- Simple form of input deflection as written
- $g_u = \frac{|h_u|^2}{\sigma_u^2}$ , this is independent of $g_1$ (BC, which corresponds to $g_U$ on MAC with order reversed , so independent of 80 below)
- L10's duality example was for $H_{MAC} = \begin{bmatrix} 50 \\ 80 \end{bmatrix}$ user 2 / user 1

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

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

# Simple example – treat BC as reversed MAC

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

- Recall (sec 2.8) 6.322 is max sum rate (for this BC)

- Direct Design with MAC

```
>> Hmac=[50 80];
>> Rxx=diag([1/7504 7503/7504]);
>> A=sqrtm(Rxx);
>> [Bu, GU, WU, S0, MSWMFU] = mu_mac(Hmac, A, [1 1] , 2)
Bu =
   0.2074    6.1146
GU =
   1.0000  138.5918
        0    1.0000
WU =
   3.0016        0
  -0.0072   0.0002
S0 =   1.0e+03 *
   0.0013        0
        0   4.8010
MSWMFU =
   1.7325
   0.0125
>> sum(Bu) =     6.3220
>> MSWMFU*Hmac*A  =
   1.0000  138.5918
   0.0072    1.0000
```

- Direct Design with BC

```
>> Hbc=[80 ; 50];
>> AU=[sqrt(.75) sqrt(.25)];
>> Lyu=[1 1];
>> cb=2;
>> [Bu, GU, S0, MSWMFunb , B] = mu_bc(Hbc, AU, Lyu , cb)
Bu =
    6.1146    0.2074
GU =  2x1 cell array
S0 =  2x1 cell array
    {[4.8010e+03]}
    {[    1.3332]}
MSWMFunb = 2x1 cell array
    {[0.0144]}
    {[0.0400]}
B =  2x1 cell array
    {[6.1146]}
    {[0.2074]}
>> sum(Bu) =     6.3220
>> MSWMFunb{1}*Hbc(1)*AU(1) =     1
>> MSWMFunb{2}*Hbc(2)*AU(2) =     1.0000
>> GU{:,:} =
    1.0000    0.5774
     0           1
```

$[\sqrt{user\ 1} \quad \cdots \quad \sqrt{user\ 2}]$

Crosstalk in BC scales @ xmit versus in MAC @ rcvr, but same cancellation

MAX single user is $b = 6.56 > 6.322$

# BC Loss

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

$$\gamma_{BC} \triangleq \frac{2^{2\cdot\bar{\mathcal{C}}} - 1}{2^{2\cdot\overline{\mathcal{C}_{BC-max-sum}}} - 1} = \gamma_{E-sum,dual-MAC} \geq \gamma_{single-user}$$

- Assured by Duality

- For slide 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

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

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 reversed in order

Filters not same

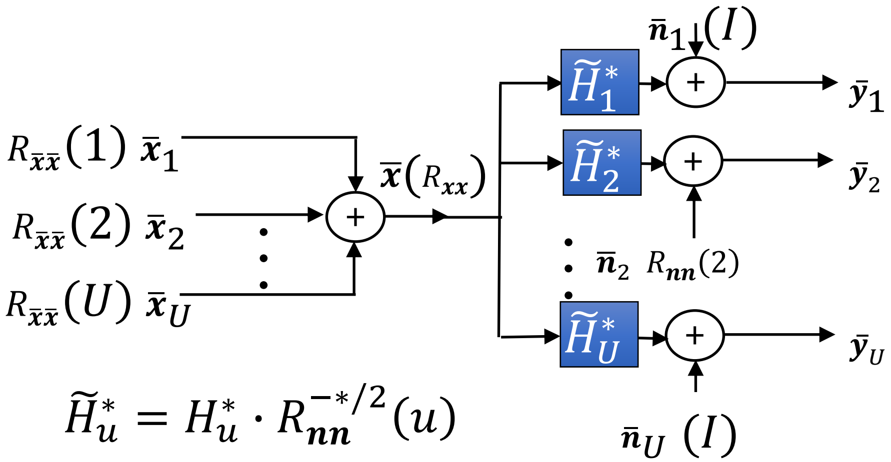
Crosstalk cancels

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

# Vector MAC / BC Duality

Section 5.5.2

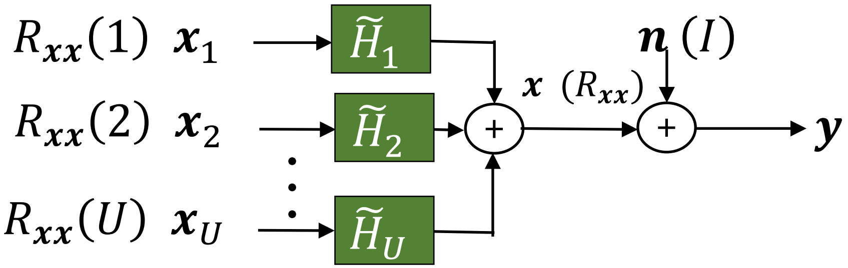
# Vector MAC/BC Duals




$$\tilde{H}_u^* = H_u^* \cdot R_{nn}^{-*/2}(u)$$

**Broadcast – input is $\bar{x}$**

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




**Multiple Access input is $x$**

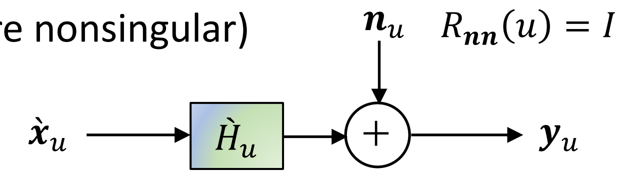
Order reversed

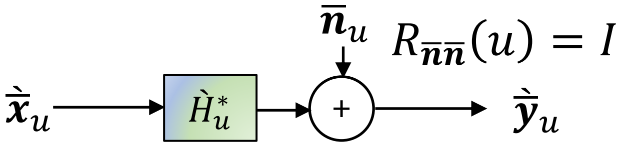
$$\mathcal{R}_{nn}(u-1) = \sum_{i=u}^{U} \tilde{H}_i \cdot R_{xx}(i) \cdot \tilde{H}_i^* + I$$

- Esum-MAC $\mathcal{E}_x = \sum_{u=1}^{U} \mathcal{E}_u = \sum_{u=1}^{U} trace\{R_{xx}(u)\} = \sum_{u=1}^{U} trace\{R_{\bar{x}\bar{x}}(u)\}$

# Reminder: Input Deflection – 2 new channels

- Deflect input to offset the channel shaping (deflectors are nonsingular)
  - $\grave{\boldsymbol{x}}_u = \mathcal{R}_{\boldsymbol{nn}}^{1/2}(u) \cdot \boldsymbol{x}_u$
  - $\bar{\grave{\boldsymbol{x}}}_u = \mathcal{R}_{\boldsymbol{nn}}^{*/2}(u) \cdot \bar{\boldsymbol{x}}_u$
- Also , re-whiten the noise
  - $\grave{H}_u = \mathcal{R}_{\boldsymbol{nn}}^{-1/2}(u) \cdot \widetilde{H}_u \cdot \mathcal{R}_{\bar{\boldsymbol{n}}\bar{\boldsymbol{n}}}^{-1/2}(u)$
- Relate autocorrelation matrices
  - $R_{\boldsymbol{xx}}(u) = \mathcal{R}_{\bar{\boldsymbol{n}}\bar{\boldsymbol{n}}}^{-1/2}(u) \cdot R_{\grave{\boldsymbol{x}}\grave{\boldsymbol{x}}}(u) \cdot \mathcal{R}_{\bar{\boldsymbol{n}}\bar{\boldsymbol{n}}}^{-*/2}(u)$
  - $R_{\bar{\boldsymbol{x}}\bar{\boldsymbol{x}}}(u) = \mathcal{R}_{\boldsymbol{nn}}^{-*/2}(u) \cdot R_{\bar{\grave{\boldsymbol{x}}}\bar{\grave{\boldsymbol{x}}}}(u) \cdot \mathcal{R}_{\boldsymbol{nn}}^{-1/2}(u)$




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

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

$$\triangleq \left|\grave{H}_u \cdot R_{\grave{\boldsymbol{x}}\grave{\boldsymbol{x}}}(u) \cdot \grave{H}_u^* + I\right| \triangleq \left|\grave{H}_u^* \cdot R_{\bar{\grave{\boldsymbol{x}}}\bar{\grave{\boldsymbol{x}}}}(u) \cdot \grave{H}_u + I\right|$$

# Finish Vector Duality

- Write deflected in terms of actual autocorrelation matrices

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

- SVD: $F_u \cdot \Lambda_u \cdot M_u^* = svd\left(\grave{H}_u\right)$
  - use "economy mode" 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 $F_u \cdot \Lambda_u \cdot M_u^* \cdot \mathcal{R}_{\bar{\boldsymbol{n}}\bar{\boldsymbol{n}}}^{*/2} \cdot R_{\boldsymbol{xx}}(u) \cdot \mathcal{R}_{\bar{\boldsymbol{n}}\bar{\boldsymbol{n}}}^{1/2} \cdot M_u \cdot \Lambda_u \cdot F_u^* + I$
- Pre/post multiply by $F$'s causes no change (nor by $M$'s in 2nd determinant)
- Same if $R_{\grave{\boldsymbol{x}}\grave{\boldsymbol{x}}}(u) = M_u \cdot F_u^* \cdot R_{\grave{\bar{\boldsymbol{x}}}\grave{\bar{\boldsymbol{x}}}}(u) \cdot F_u \cdot M_u^*$
- MAC2BC: $R_{\overline{\boldsymbol{xx}}}(u) = \mathcal{R}_{\boldsymbol{nn}}^{-1/2}(u) \cdot F_u M_u^* \cdot \mathcal{R}_{\overline{\boldsymbol{nn}}}^{-*/2}(u) \cdot R_{\boldsymbol{xx}}(u) \cdot \mathcal{R}_{\overline{\boldsymbol{nn}}}^{-1/2}(u) \cdot M_u F_u^* \cdot \mathcal{R}_{\boldsymbol{nn}}^{-*/2}(u)$
- BC2MAC: $R_{\boldsymbol{xx}}(u) = \mathcal{R}_{\overline{\boldsymbol{nn}}}^{-*/2}(u) \cdot M_u \cdot F_u^* \cdot \mathcal{R}_{\boldsymbol{nn}}^{-1/2}(u) \cdot R_{\overline{\boldsymbol{xx}}}(u) \cdot \mathcal{R}_{\boldsymbol{nn}}^{-*/2}(u) \cdot F_u \cdot M_u^* \cdot \mathcal{R}_{\overline{\boldsymbol{nn}}}^{-1/2}(u)$

# MAC2BC Full Algorithm

$R_{xx}(u)$ from minPMAC

**Given:**

$$R_{xx}(u) \text{ for } u = 1, \ldots, U \;;\; R_{\overline{xx}}(u) = R_{\overline{xx}} = 0$$
$$\mathcal{R}_{nn}(U) = \mathcal{R}_{\overline{nn}}(1) = I$$
$$\text{BC: } \tilde{H}_u^*, R_{nn}(u)$$
$$\text{MAC: } \tilde{H}_u = H_u \cdot R_{nn}^{-1/2}(u)$$

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

$$\text{For } u = 1, \ldots, U$$
$$\grave{H}_u = \mathcal{R}_{nn}^{-1/2}(u) \cdot \tilde{H}_u \cdot \mathcal{R}_{\overline{nn}}^{-1/2}(u)$$
$$F_u \cdot \Lambda_u \cdot M_u^* = svd\left(\grave{H}_u\right)$$
$$R_{\overline{xx}}(u) = \mathcal{R}_{nn}^{-1/2}(u) \cdot F_u \cdot M_u^* \cdot \mathcal{R}_{\overline{nn}}^{*/2}(u) \cdot R_{xx}(u) \cdot \mathcal{R}_{\overline{nn}}^{1/2}(u) \cdot M_u \cdot F_u^* \cdot \mathcal{R}_{nn}^{-*/2}(u)$$
$$R_{\overline{xx}} = R_{\overline{xx}} + R_{\overline{xx}}(u)$$
$$\mathcal{R}_{\overline{nn}}(u+1) = \mathcal{R}_{\overline{nn}}(u) + \tilde{H}_u^* \cdot R_{\overline{xx}} \cdot \tilde{H}_u \;;\; \text{skip } u = U$$

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

# BC2MAC Full Algorithm

Given:

$$R_{\overline{xx}}(u) \text{ for } u = 1, \ldots, U \;;\; \mathcal{R}_{nn}(u) = \mathcal{R}_{\overline{nn}}(u) = 0$$
$$\mathcal{R}_{nn}(U) = \mathcal{R}_{\overline{nn}}(1) = I \;;\; R_{\overline{xx}} = 0$$
$$\text{BC: } \tilde{H}_u^*, R_{nn}(u)$$
$$\text{MAC: } \tilde{H}_u = H_u \cdot R_{nn}^{-1/2}(u)$$

↓

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

↓

$$\text{For } u = U, \ldots, 1$$
$$\grave{H}_u = \mathcal{R}_{nn}^{-1/2}(u) \cdot \tilde{H}_u \cdot \mathcal{R}_{\overline{nn}}^{-1/2}(u)$$
$$F_u \cdot \Lambda_u \cdot M_u^* = svd\left(\grave{H}_u\right)$$
$$R_{xx}(u) = \mathcal{R}_{\overline{nn}}^{-*/2}(u) \cdot M_u F_u^* \cdot \mathcal{R}_{nn}^{1/2}(u) \cdot R_{\overline{xx}}(u) \cdot \mathcal{R}_{nn}^{*/2}(u) \cdot F_u M_u^* \cdot \mathcal{R}_{\overline{nn}}^{-1/2}(u)$$
$$\mathcal{R}_{nn}(u-1) = \mathcal{R}_{nn}(u) + \tilde{H}_u \cdot R_{xx}(u) \cdot \tilde{H}_u^* \;;\; \text{skip } u = 1$$

- Reverse is less interesting

# Duality conversion program (Lx constant)

- So duality and the Rxxb = mac2bcMimo(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 $n = 1, \ldots, \bar{N}$.

```
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

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

Matlab order
Is reverse of MAC
(same as BC)

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

- 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 $R_{\boldsymbol{xx}}(u)$
- There will be corresponding zeroed columns/rows on $R_{\overline{\boldsymbol{xx}}}(u)$ outputs
- Duality algorithm's inverted/square-root matrices remain nonsingular

**Possible project**
Variable-Lxu mac2bc / bc2mac

# Reversal bc2mac program (Lx constant)

- Rxxm = bc2mac(Rxxb , Hmac) program
  - The input Rxxb is $L_x \times L_x \times U$ where $L_x$ is for the dual MAC.
  - The input Hmac is $L_y \times L_x \times U$ is for the dual MAC (not the BC).
    - To reverse from mac2bc, input the mac2bc output (Rxxb) with Hbc= **conj**(permute(Hvec(:,:,end:-1:1), [order' 3])
  - The output Rxxm 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 $n = 1, \ldots, \bar{N}$.

```
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 Lx , and N=1, then the designer can find the Hbc by
Hbc=conj( **permute(** Hmac(:,:,end:-1:1) **, [2 1  3] )** )
Or reverse from BC to MAC with
Hmac=conj( **permute(** Hbc(:,:,end:-1:1) **, [2 1  3] )** )

- This reverse program is usually not necessary because the designer optimizes for the dual MAC
- This Rxxm then leads through mac2bc to Rxxb to complete the 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 close
  - Note order reversal – user 1 is in best position on BC, but worst position in MAC – a feature of duality

# MAC-dual Design

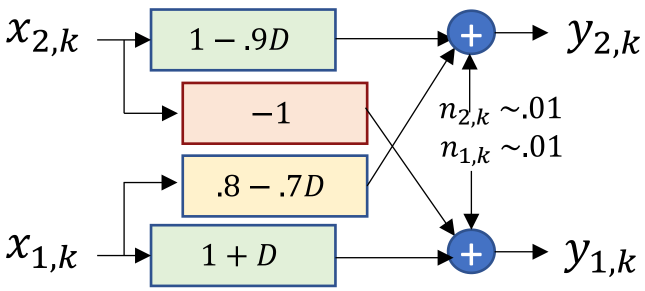
Section 5.5.4

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

$\mathcal{E}_2 = 1$

Per symbol

$\mathcal{E}_1 = 1$




- Table

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

```
>> H
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
```

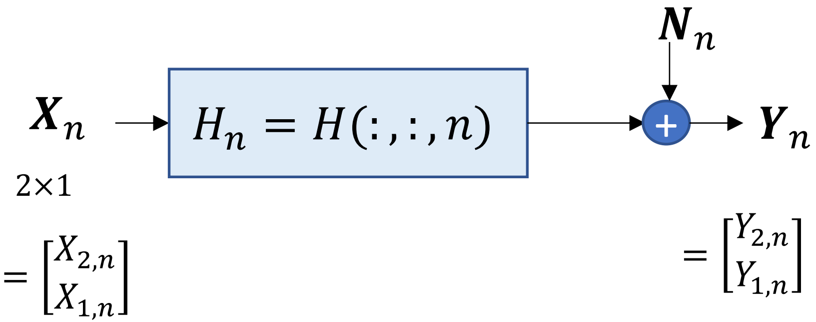
Use 2×2 Vector DMT

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

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

**So far, just H →**

**8 tonal 2×2 channels**



# 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

>> 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 Au**
**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?**

**(Lowpass & Highpass & all tones used)**

# 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)

- 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)
```

- Will be useful to know 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 maintained on each tone, but reversed decoding order**

# And the loss

- For this problem, the data rate is already close to single-user WF

- Answer for E=[1 1] using SWF:

```
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
```

- Maximum Esum MAC rate sum:

```
[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)

STANFORD

# End Lecture 16

# SVD "economy"

$$\tilde{H} = [F \quad d_F] \cdot \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} M^* \\ d_M^* \end{bmatrix}$$

- The rank is $\wp_{\tilde{H}}$ and so $\Lambda$ is a $\wp_{\tilde{H}} \times \wp_{\tilde{H}}$ diagonal matrix
- The last $(L_y - \wp_{\tilde{H}})$ columns of left-unitary matrix are "don't care" $d_F$ ; $F$ is $L_y \times \wp_{\tilde{H}}$ matrix
- The last $(L_x - \wp_{\tilde{H}})$ columns of right-unitary matrix are "don't care" $d_M$; $M$ is $L_x \times \wp_{\tilde{H}}$ matrix

$$\tilde{H} = F \cdot \Lambda \cdot M^*$$

- $F$ and $F^*$ are pseudoinverses; $F^* \cdot F = I_{\wp_{\tilde{H}}}$ and $F \cdot F^* = \begin{bmatrix} I_{\wp_{\tilde{H}}} & 0 \\ 0 & 0 \end{bmatrix}$
- Similarly, $M^* \cdot M = I_{\wp_{\tilde{H}}}$ and $M \cdot M^* = \begin{bmatrix} I_{\wp_{\tilde{H}}} & 0 \\ 0 & 0 \end{bmatrix}$
- $F \cdot M^*$ is an $L_y \times L_x$ matrix; similarly, $M \cdot F^*$ is a $L_x \times L_y$

# Vector Duality Dimensionality Observations

- If the MAC $H$ is $L_y \times \mathcal{L}_x$ where $\mathcal{L}_x = \sum_{u=1}^{U} L_{x,u}$ , the dual BC $H_{BC} = \mathcal{J}_x \cdot H^* \cdot \mathcal{J}_x$ has the following
  - $H_{BC}$ is $\mathcal{L}_x \times L_y$ and thus will have $L_{x,u}$ output dimensions at each receiver
  - $R_{\overline{xx}}(u)$ is $L_y \times L_y$ and thus forms through an $L_y \times L_y$ user-$u$-component modulator matrix $A_u$
  - There are thus $U \cdot L_y$ total "GDFE-Precoder" input dimensions for the BC that transform to $L_y$ through

$$\bar{x} = \sum_{u=1}^{U} A_u \cdot \bar{\boldsymbol{v}}_u$$

- Thus $G_{BC}$ is $U \cdot L_y \times U \cdot L_y$ , but $G_{MAC}$ is $\mathcal{L}_x \times \mathcal{L}_x$
- The BC inputs are summed, and so are the autocorrelation matrix contributions $R_{\overline{xx}}(u)$ that vector duality produces.
- The receivers through $\mathcal{W}_u$ are $L_y \times L_{x,u}$ matrix multiplies (followed by any modulos) and $L_y$-dimensional decoders

Example: $H$ is 2 x 3 with $L_y = 2$; $L_{x,u} = 1$; $U = 3$ , then $G_{BC}$ is $6 \times 6$ , but **dual's** $G_{MAC}$ is $3 \times 3$

$\mathcal{W}_{BC}$ is $6 \times 6$ , with 3 , $2 \times 2$ blocks $\mathcal{W}_{MAC}$ is $3 \times 2$

# Order Reversal & The Dual

- Reverse the order of MAC so that BC has user 1 at top
- $\mathcal{J}_x$ applies to $\boldsymbol{x}$ so that $\mathcal{J}_x \cdot \boldsymbol{x}$ reverses the input vector $\boldsymbol{x}$'s user order

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

- $\mathcal{J}_y$ applies to $\boldsymbol{y}$ so that $\mathcal{J}_y \cdot \boldsymbol{y}$ reverses the output vector $\boldsymbol{y}$'s user order
  - Don't really use this much in MAC/BC duality, our main purpose
  - The text hangs on to it a little longer, but not here in lecture

$$\mathcal{J}_y \triangleq \begin{bmatrix} 0 & 0 & I_{L_{y,U}} \\ 0 & \iddots & 0 \\ I_{L_{y,1}} & 0 & 0 \end{bmatrix}$$

- Thus, reversing a MAC's input order is $\tilde{H} \cdot \mathcal{J}_x$
- The channel $\tilde{H}$'s dual is its transpose with order reversed

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

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

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