STANFORD

*Lecture 14*

# MAC GDFEs and Design Measures

*May 21, 2024*

JOHN M. CIOFFI

Hitachi Professor Emeritus of Engineering

Instructor EE379B – Spring 2024

# Announcements & Agenda

- Announcements
  - Section 5.4
  - PS7 – last homework, 2 weeks, double weight.

- Problem Set 7 = PS7 (due June 5 or 7)
  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

- Agenda
  - ZF/MMSE Convergence Conditions (hold over from L13)
  - MAC and GDFE Comparison (Sec 5.4.1)
  - Tonal MAC with DMT (Section 5.4.2)
    - Tonal GDFE
    - SWF
  - Designs with weighted sums (Section 5.4.3)

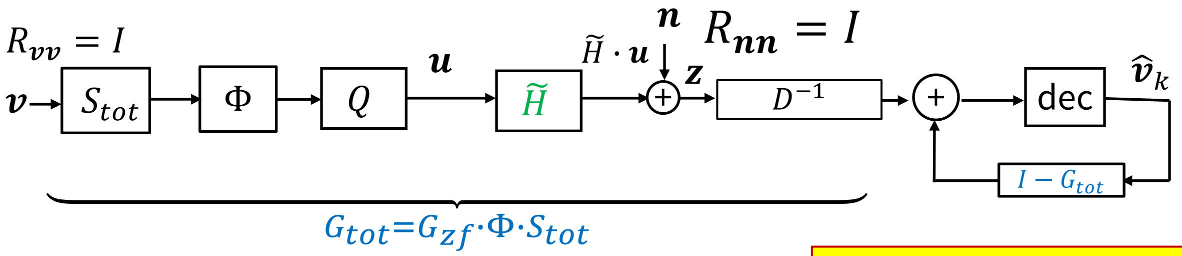
# ZF/MMSE convergence conditions

Section 5.3.5

# Use Zero Forcing GDFE with Water-fill?

- **To Show:** MMSE GDFE's triangular matrix $G$ equals the triangular factor of $\tilde{H} = D \cdot G_{zf} \cdot Q^*$?
  - **IF the input is water-fill and nonsingular (so resampled):**
- $D^{-1} \cdot \tilde{H} = F \cdot \Lambda \cdot M^*$ with energies $diag\{\mathcal{E}\} = K - \Lambda^{-2}$ → $R_{\boldsymbol{uu}} = M \cdot (K - \Lambda^{-2}) \cdot M^* = Q \cdot \Phi \cdot \Phi^* \cdot Q^*$
  - To find $\Phi$, Cholesky–Factor $Q^* \cdot R_{\boldsymbol{uu}} \cdot Q = \Phi \cdot \Phi^*$ .
  - Define monic triangular: $G_{tot} = G_{zf} \cdot \Phi \cdot S_{tot}$ and note the MSWF and ZF-GDFE with receiver diagonal $D^{-1}$ is:

Find this




Zero-forcing with WF

| | |
|---|---|
| always | $\tilde{H} = F \cdot \Lambda \cdot M^* = D \cdot G_{zf} \cdot Q^*$ |
| nonsingular | Cholesky: $Q^* \cdot R_{\boldsymbol{uu}} \cdot Q = \Phi \cdot \Phi^*$ |

**This zero-forcing actually has $G_{tot}$ , not $G_{zf}$, as feedback; however, designers often use flat input $S_{tot} = \bar{\mathcal{E}}_{\boldsymbol{x}} \cdot I$, where $K - \Lambda^{-2} \approx \bar{\mathcal{E}}_{\boldsymbol{x}} \cdot I$ which sets $G_{tot} = G_{zf}$ $(M = Q$ , so $\Phi = I)$.**

**Equivalently, the input $v$ as $N \rightarrow \infty$ has $S_{tot} \rightarrow$ constant, so then exactly true.**

**So maybe just use ZF and do only 1 rq factorization (not 2 Cholesky's).**

# The two water-fill receivers

## Rearrange Water-fill

$R_{\boldsymbol{uu}} = M \cdot [K \cdot I - \Lambda^{-2}] \cdot M^* = Q \cdot \Phi \cdot \Phi^* \cdot Q^*$

Noting

$M \cdot \Lambda^{-2} \cdot M^* = \tilde{H}^{-1} \cdot D^2 \cdot \tilde{H}^{-*} = R_f^{-1}$

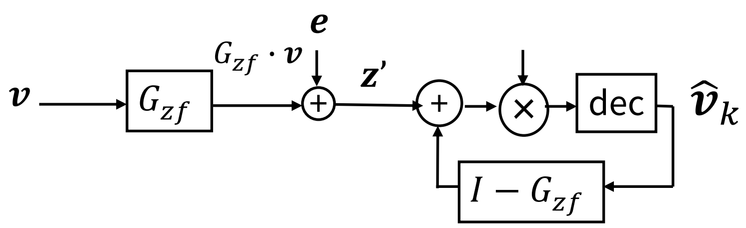
$$K \cdot I = Q \cdot \Phi \cdot \Phi^* \cdot Q^* + Q \cdot G_{zf}^{-1} \cdot G_{zf}^{-*} \cdot Q^*$$

$$K \cdot G_{zf} \cdot G_{zf}^* = \underbrace{G_{tot} \cdot S_{tot}^{-2} \cdot G_{tot}^*}_{R_f} + I$$

**This is $R_b^{-1}$ !**

Equivalent MMSE with WF (same $G = G'_{zf}$):

Performance still not same, MMSE slightly better




**$G = G_{zf}$ so water-filling leads to MMSE having same feedback as the water-fill zero-forcing, at least the flat-energy approximation to wf**

**However, $G_{tot}$ , is really the ZF cascade when non flat or for finite symbol length.**

- The $K$ calculation needs to be positive (or increase $K$ so slightly positive and then scale down the resulting energies); the water-fill input was not full rank so there is a loss; can be small in wireless.
- However, MMSE still has (slightly) higher SNR so use of $G = G_{zf}$ with waterfill as feedback, not $G_{tot}$, is highest SNR.
  - Note carefully on previous slide that the ZF-GDFE feedforward filter is not the same as MMSE-GDFE, even if feedback is same.

# Worst-Case Noise equates ZF and MMSE

- Easy proof: WCN diagonalizes the primary-user receivers from BC in Chapter 2.
  - Step 1: Separates noise-whitened-noise-matched channel's triangular part
    - $\widetilde{H} \triangleq R_{wcn}^{+} \cdot H = R_{zf} \cdot Q_{zf}^{*} = \begin{bmatrix} 0 & R_1 \\ 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} q \\ Q_1^{*} \end{bmatrix}$

$R_1$ is triangular part
$Q_1^{*}$ is corresponding column set

  - Step 2: cascade that triangular part with Cholesky of rotated input:
    - $R_{\tilde{x}\tilde{x}} = Q_1^{*} \cdot R_{xx} \cdot Q_1 = \Phi \cdot \Phi^{*}$ where $\Phi$ is triangular Cholesky factor.
    - $A = Q_1 \cdot \Phi$.
  - Step3: find the channel gains/SNR and feedback section:
    - The cascade of receiver triangular inverse is $D_A \cdot G_{zf} = R_1 \cdot \Phi$.

# Example of WCN's RCVR Diagonalization

```
>> H =
   0.9000   1.0000        0        0
        0   0.9000   1.0000        0
        0        0   0.9000   1.0000
H=(1/sqrt(.181))*H;
>> Rxx=eye(4);
>>  [Rwcn,b]=wcnoise(Rxx,H,1);
>> b =   4.8024
>> Htilde=inv(Rwcn)*H;
>> [R,Q]=rq(Htilde);
>> R =
        0  -2.7323  -1.4039   0.0071
        0        0  -2.7077  -1.2346
        0        0        0  -3.0719
Q1=Q(:,2:4);
Rxxrot=Q1'*Rxx*Q1;
Rup=R(:,2:4) ;
Phibar=lohc(Rxxrot);
DA=diag(diag(Rup*Phibar));
>> G=inv(DA)*Rup*Phibar
   1.0000   0.5138  -0.0026
        0   1.0000   0.4559
        0        0   1.0000
>> A=Q1*Phibar;
>> sri=inv(sqrtm(Rwcn));
```

**Rwcn is nonsingular here**

**Finding ZF modulator G**

```
>> [snrGDFEu, GU, WU, S0, MSWMFU,b,bbar] = computeGDFE(sri*H, A, 2, 4)
snrGDFEu =   1.1340 dB

GU =
   1.0000   0.5826  -0.0030
        0   1.0000   0.5160
        0        0   1.0000
WU =
   0.1339        0        0
  -0.0676   0.1317        0
   0.0244  -0.0470   0.1031
S0 =
   8.4657        0        0
        0   8.5956        0
        0        0  10.7006
MSWMFU =
  -0.3657  -0.0128   0.0054
  -0.0125  -0.3560  -0.0125
   0.0047  -0.0111  -0.3164
>> MSWMFU*sri' =
  -0.3660  -0.0000   0.0000
  -0.0000  -0.3565   0.0000
   0.0000  -0.0000  -0.3167
>> b' =   1.5408   1.5518   1.7098
>> bbar =   1.2006
>> sum(b) =   4.8024 (checks)
```

**Diagonal !**
**(so ZF = MMSE GDFE)**

**Only need**
**RQ fact and lohc**

- See Example 5.3.6 with non-white Rxx

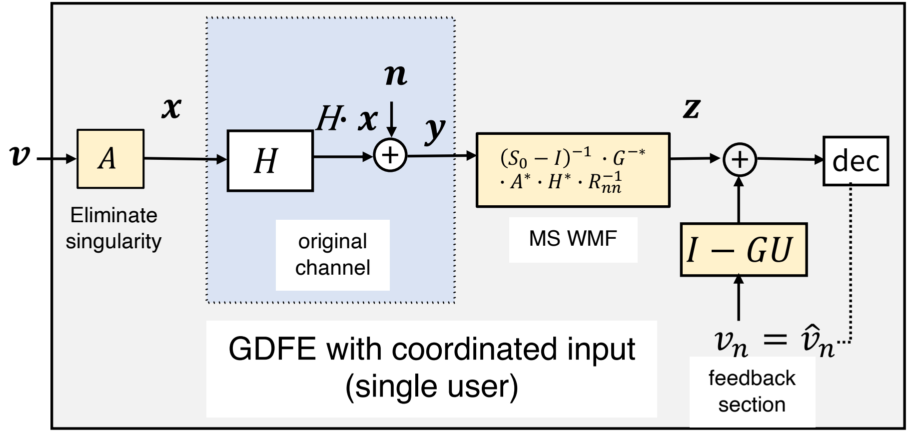
# Some Final Comments

- The GDFE is canonical – capacity rate is reliably achievable with $\Gamma = 0$ (or capacity less shaping loss).
- GDFE can have error propagation (limited to $\bar{N}$) if $\Gamma > 0$ dB.
  - Unless it is VC (~DMT) , which is ML decoder uniquely amoung all GDFEs.
  - Other GDFE's becoming increasingly less favorable performance relative to VC/DMT as gap grows.
- The DMT form benefits from FFT algorithms so also more cost effective than the others.
- By Separation Theorem, Coded-OFDM can capture the DMT benefits also without error propagation.
  - But will lose more rapidly lose performance relatively if input is not water filling.
- The MMSE-DFE is limiting (stationary) case of the CDFE and can be canonical.
  - Set of MMSE-DFE's for each of which PWC holds, which
  - has unlimited error propagation (use precoder) and also degrades more rapidly for nonzero-gap codes

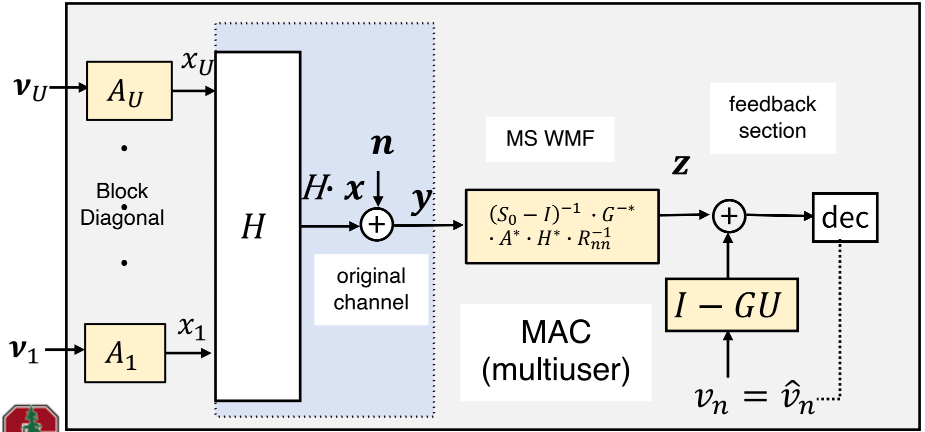
**Eventual Global Conclusion: Use DMT (wireline) or C-OFDM (wireless) on almost all difficult single-user transmission systems.**

# MAC and GDFE Comparison

# The MMSE MAC vs MMSE GDFE




- GDFE is:
  - Designed for "single-user" $H$ ; $A = R_{xx}^{1/2}$ ,
  - MMSE: $R_b^{-1} = R_{xx}^{-*/2} \cdot H^* \cdot R_{nn}^{-1} \cdot H \cdot R_{xx}^{-1/2} + I = G \cdot S_0 \cdot G^*$ ,
  - Canonical (decisions correct),
  - Input has only $trace\{R_{xx}\} \leq \mathcal{E}_x$ energy constraint, &
  - Rate-independent of dimensional order.




- MAC has:
  - **block-diag** $R_{xx}$, with $trace\{R_{xx}\} \leq \mathcal{E}_x$ ,
    - only in energy-sum case, and otherwise
  - input energies $trace\{R_{xx}(u)\} \leq \mathcal{E}_u$ ,
  - separated locations so $A_u = R_{xx}^{1/2}(u)$ ; $A = R_{xx}^{1/2}$,
  - MMSE: $R_b^{-1} = R_{xx}^{-*/2} \cdot H^* \cdot R_{nn}^{-1} \cdot H \cdot R_{xx}^{-1/2} + I = G \cdot S_0 \cdot G^*$
  - Canonical performance (decisions correct)
    - Rates per user – order shifts sum rate among users.

# A Scalar Example Revisited

- **MAC** 80/60 channel

```
>> H=[80 60];
>> Rxx=0.5*eye(2);   (equal energy both dim/users)
>> A=[sqrt(.5) 0' ; 0 sqrt(.5) ];
>> Lxu=[1 1];
>> cb=2;

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

b    =   5.8222   0.3218
GU =   1.0000   0.7500     MSWMFU =   0.0177
             0   1.0000                0.0236
S0 =  1.0e+03 *
        3.2010         0
             0    0.0016

>> sum(b) =   6.1440
>> 10*log10(2^(6.1440)-1) =   18.4334 dB
```

- **GDFE** – remove singularity

```
>> [F,L,M]=svd(H);
>> [F,L,M]=svd(H)
F = 1
L =  100     0
M =
    0.8000   -0.6000
    0.6000    0.8000
>> 0.5*log2(1+ 0.5*L(1)^2) =  6.1440

All energy on pass space
>> 0.5*log2(1+ 1*L(1)^2) =   6.64 > 6.144
```

- For 6.64, (single-user) input is:
  $$\boldsymbol{x} = \begin{bmatrix} .800 \\ .600 \end{bmatrix} \cdot v$$
- $v$ goes to both channel input dimensions (not MAC).
- All GDFE's with this input $R_{xx} = [1\ 0\ ;0\ 0]$ perform same
  - and trivially have $G = 1$.

# Or use computeGDFE.m

```
>> [snrGDFEu, GU, WU, S0, MSWMFU, b, bbar,
snrGLEu] = computeGDFE(H, A, cb)

snrGDFEu =   18.4334 dB
GU =
    1.0000    0.7500
         0    1.0000
WU =
    0.0003         0
   -1.3333    1.7783
S0 =   1.0e+03 *
    3.2010         0
         0    0.0016
MSWMFU =
    0.0177
    0.0236
b =
    5.8222
    0.3218
bbar =    3.0720
snrGLEu =   16.8125 dB
>> sum(b) =    6.1440
```

**No Lx input**

```
>> [F,L,M]=svd(H);
>> Lx=2;
[snrGDFEu, GU, WU, S0, MSWMFU, b, bbar, snrGLEu] =
computeGDFE(H, M(:,1), cb, Lx)
snrGDFEu =   19.9566
GU =     1
WU =   1.0000e-04
S0 = 10001
MSWMFU =   0.0100
b =    6.6439
bbar =    3.3220
snrGLEu =   19.9566(dB) – linear is same for VC

(same energy, but more going to best mode)
```

**Not a MAC, A is not diag.**

**Better to use mu_mac with a MAC, than to play with cb & Lx on computeGDFE, which is really for single user GDFEs.**

**Similarly: use computeGDFE on single user. ComputeGDFE also provides linear output.**

Correct comparison with GDFE notes the A input has 2 real dimensions
VC resets the Lx to 2 as optional $4^{th}$ computeGDFE input.

# MAC Loss

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

$$\gamma_{MAC} \triangleq \frac{2^{2\cdot \bar{\mathcal{C}}} - 1}{2^{2\cdot \bar{\mathcal{C}}_{e-sum}} - 1}$$

```
>> [~, btsum , ~]=macmax(1,[80 60],[1 1],1,2)
btemp =    6.3220
```

- For the previous example: $\gamma_{MAC} = \frac{2^{6.64}-1}{2^{6.322}-1} = 1.5$ dB.

- Clearly $0 \leq \gamma_{MAC} \leq 1$.


See also *split-dimensionality* example in Section 5.4.1.

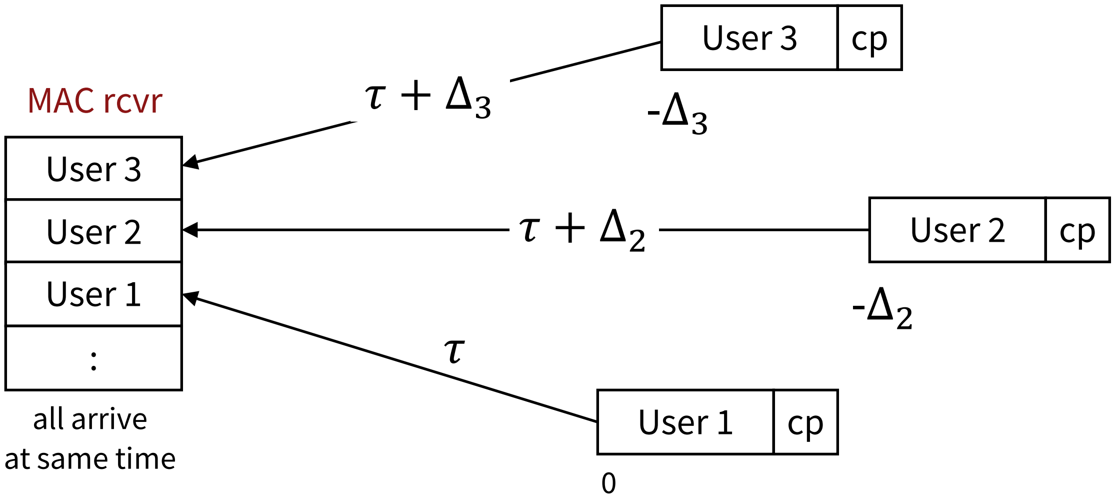
# Tonal MAC (with DMT)

## Section 5.4.2

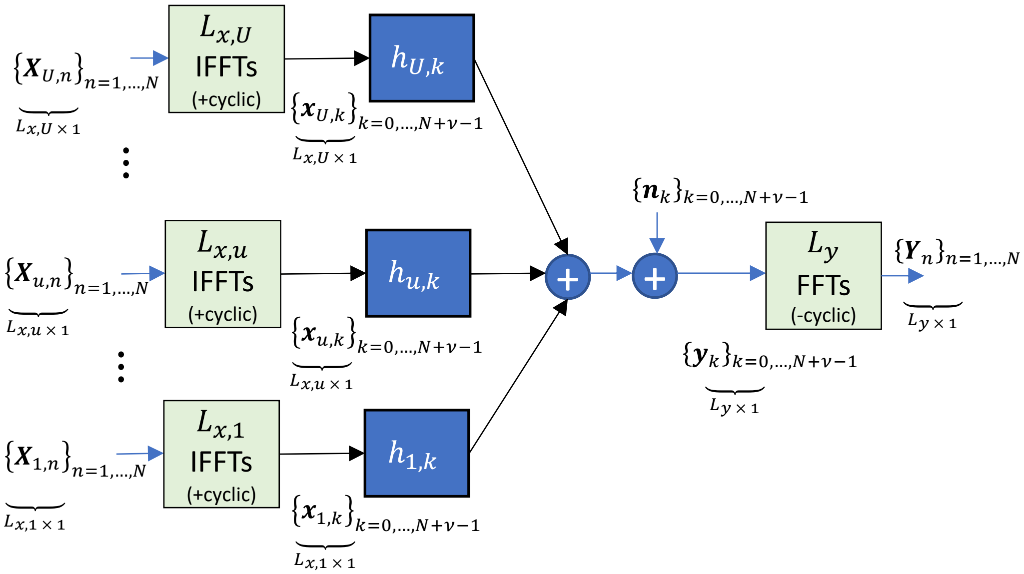
Note Section 5.4.1’s two-user ISI-GDFE is interesting, but largely becomes superfluous with the tonal vector-DMT system.

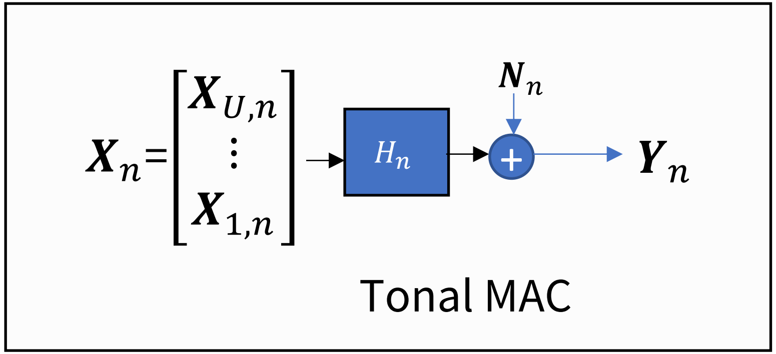
# Align Receiver DMT Symbols for MAC

- So far, examples have largely been space time (with a few antennas).
- In practice, there usually is also a temporal (time-freq) C-OFDM or DMT system **also** present.



# Vector DMT/OFDM with MAC




$$R_{\boldsymbol{XX}}(u,n) = \mathbb{E}\left[\boldsymbol{X}_{u,n} \cdot \boldsymbol{X}^*_{u,n}\right]$$

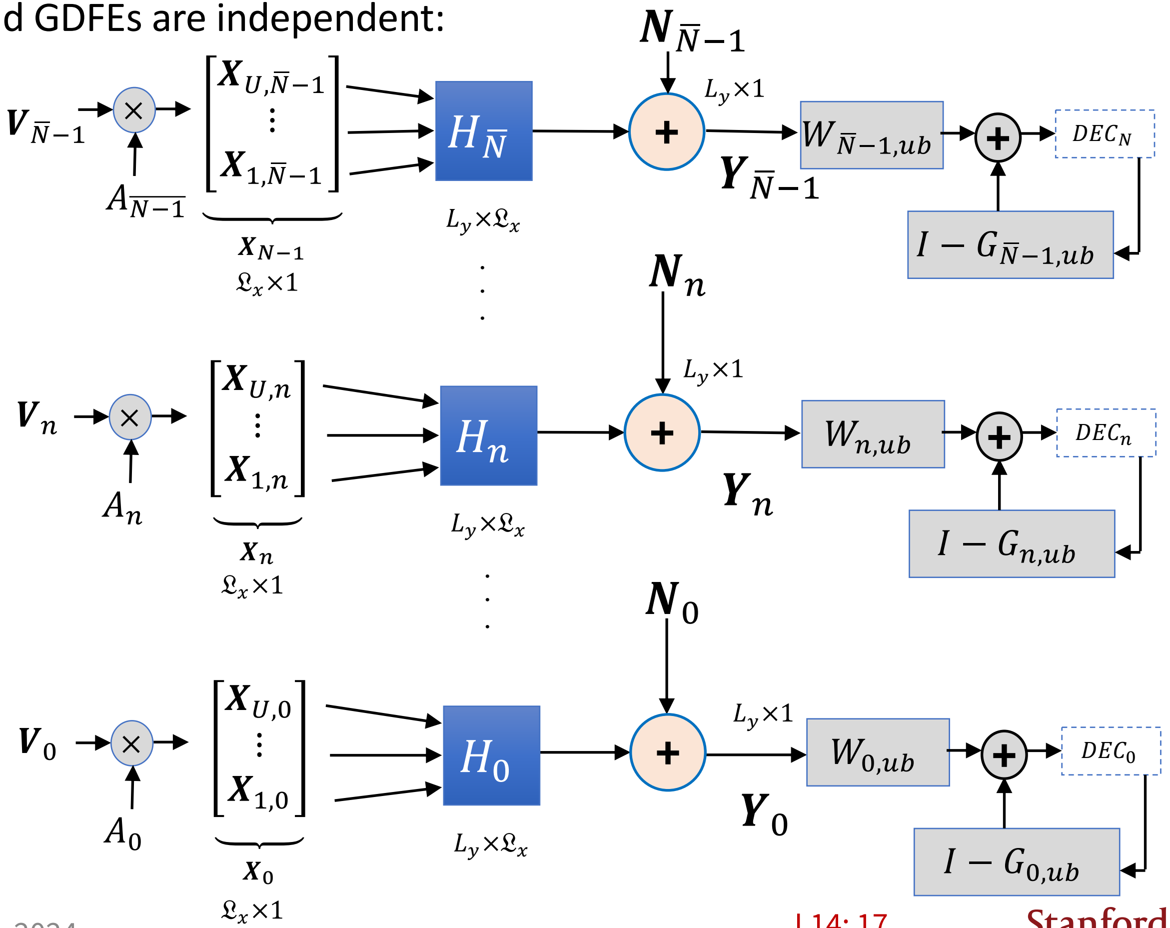
$$\sum_n trace\,\{R_{\boldsymbol{XX}}(u,n)\} \le \mathcal{E}_u \;\forall\, u = 1, \dots, U$$

$$\text{Esum–MAC: } \sum_{u=1}^{U} \sum_{n=0}^{\bar{N}} trace\,\{R_{\boldsymbol{XX}}(u,n)\} \le \mathcal{E}_{\boldsymbol{x}}$$

- Symbol boundaries align through cyclic-extension (guard period) uses (even with different channel delays, may increase $\nu$).
- Basically, an IFFT per transmit-antenna-user – symbol boundary delayed/advanced so that receiver FFT captures all $U$.
- Discrete=time MT MAC, indeed all SVD's, Cholesky's, and QR factorizations become "frequency-dependent" (finite $n$ replaces $\lim_{n\to\infty}[\;]$).

# Tonal GDFEs with MAC

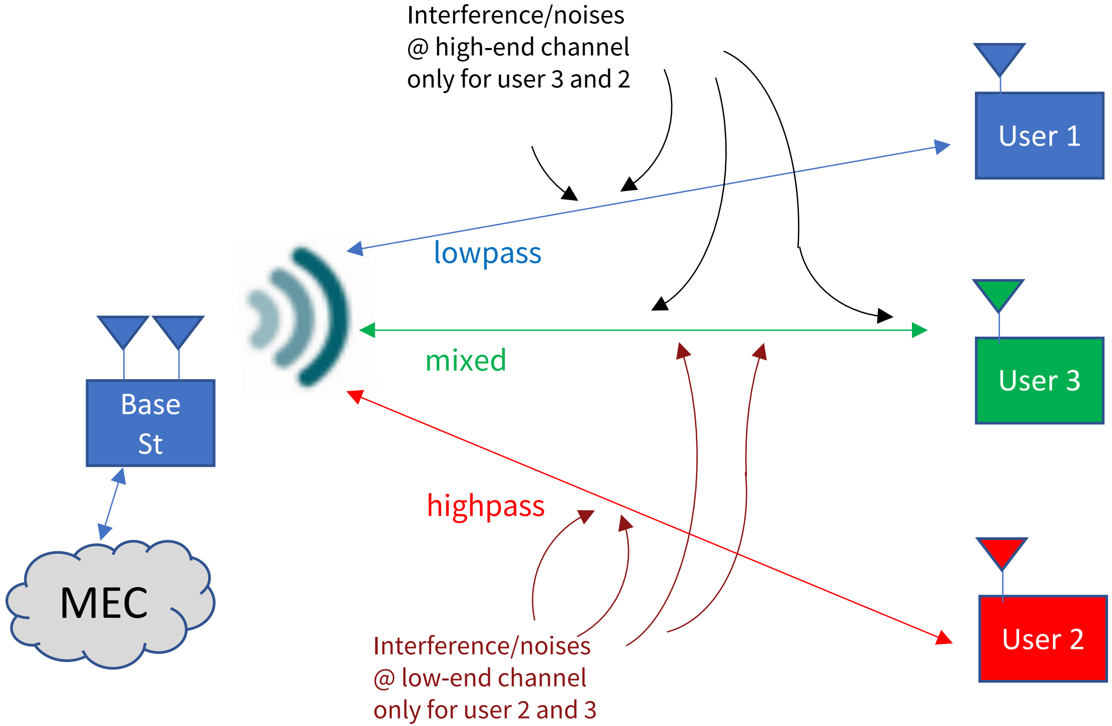
- Discrete tonal modulators and GDFEs are independent:




- Put an "$n$" index on all the GDFE design equations.

$$
\begin{aligned}
b_{u,\ell} &= \sum_{n=0}^{\bar{N}-1} b_{u,\ell,n} \\
b_{u,n} &= \sum_{\ell=1}^{L_{x,u}} b_{u,\ell,n} \\
b_{u} &= \sum_{\ell=1}^{L_{x,u}} \sum_{n=0}^{\bar{N}-1} b_{u,\ell,n} \\
b &= \sum_{u=1}^{U} \sum_{\ell=1}^{L_{x,u}} \sum_{n=0}^{\bar{N}-1} b_{u,\ell,n} \quad .
\end{aligned}
$$

# Example complex BB channel




**More users than antennas**

This allows up to 64 resource Blocks (tones) for each user, all in same channel.

- Illustrates many effects
  - ISI and crosstalk

$$H(D) = \begin{bmatrix} 1 + .9 \cdot D & -.3 \cdot D + .2 \cdot D^2 & .8 \\ .5 \cdot D - .4 \cdot D^2 & 1 - D - .63 \cdot D^2 + .648 \cdot D^3 & 1 - D \end{bmatrix}$$

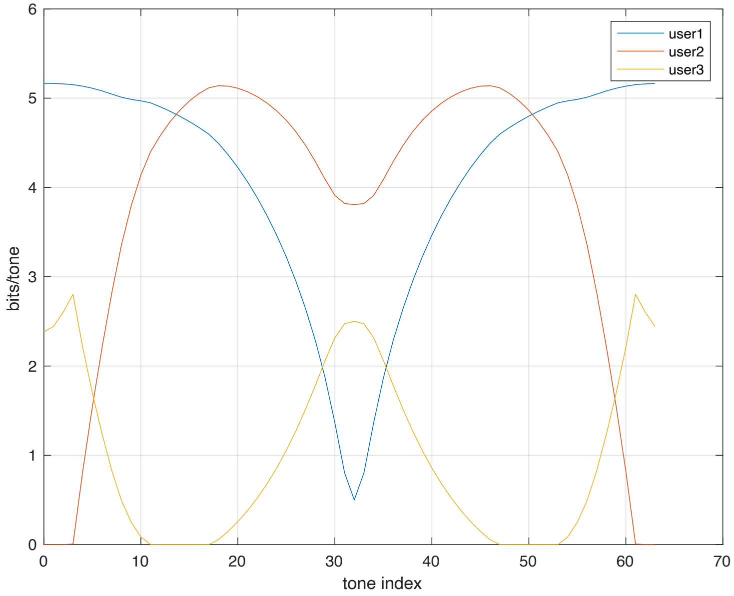


A bit oversimplified, as real only for cplex channel.

# Example has ISI and MIMO together

- There are 3 user channels .
- Any tone will maximally have rank $\wp_H$=2; $\mathcal{N}_0 = .01$.

$$H(D) = \begin{bmatrix} 1 + .9D & -.3D + .2D^2 & .8 \\ .5D - .4D^2 & 1 - D - .63D^2 + .648D^3 & 1 - D \end{bmatrix}$$




```
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(:,:,1) =
   10    0    8
    0   10   10
h(:,:,2) =
    9   -3    0
    5  -10  -10
h(:,:,3) =
         0    2.0000         0
    4.0000   -6.3000         0
h(:,:,4) =
         0         0         0
         0    6.4800         0
N=8;
H = fft(h, N, 3)
>> H = fft(h, N, 3)
H(:,:,1) =
  19.0000 + 0.0000i  -1.0000 + 0.0000i   8.0000 + 0.0000i
   9.0000 + 0.0000i   0.1800 + 0.0000i   0.0000 + 0.0000i
H(:,:,2) =
  16.3640 - 6.3640i  -2.1213 + 0.1213i   8.0000 + 0.0000i
   3.5355 - 7.5355i  -1.6531 + 8.7890i   2.9289 + 7.0711i
And 6 more values, see text
```

← Increase to 64 – look ahead at MAC with equal energy every dimension.

# Actual MAC/GDFE calculations for L14:17

**White-Input Tonal GDFE**

```
Nmax=32;
U=3;
Ly=2;
cb=1;
Lxu=[1 1 1];
bsum=zeros(1,Nmax);
for index=1:Nmax
 i=2*index;
 H = fft(h, i, 3);
 GU=zeros(U,U,i);
 WU=zeros(U,U,i);
 S0=zeros(U,U,i);
 Bu=zeros(U,i);
 MSWMFU=zeros(U,Ly,i);
 AU=zeros(3,3,i);
 for n=1:i
  AU(:,:,n)=sqrt(i)/sqrt(i+3)*eye(3);
 end
 for n=1:i
  [Bu(:,n), GU(:,:,n) , WU(:,:,n),S0(:,:,n), MSWMFU(:,:,n)] = ...
  mu_mac(H(:,:,n), AU(:,:,n), Lxu, cb);
 end
bvec=sum(Bu');
Bsum(index) = sum(bvec);
End
bvec = 445.1264  412.8794  132.7477
sum(bvec) = 990.7535
```

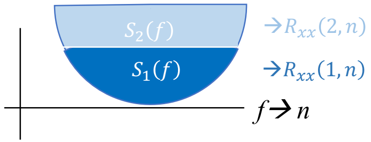
```
>> GU(:,:,23)  =
  1.0000 + 0.0000i  -1.3365 + 0.3447i  -0.3093 + 0.3130i
  0.0000 + 0.0000i   1.0000 + 0.0000i   0.8108 + 0.4218i
  0.0000 + 0.0000i   0.0000 + 0.0000i   1.0000 + 0.0000i
>> MSWMFU(:,:,15) =
  0.0454 + 0.0341i  -0.0105 + 0.0249i
  0.0183 + 0.0179i   0.0275 - 0.0468i
  0.0589 + 0.0199i   0.0179 - 0.0416i
>> SNRs = 10*log10(diag(S0(:,:,29)).^(64/67)-eye(3)) =

  12.1946
  20.7691
   8.8939
```

**Feedback is sizeable**

# Simultaneous Water Filling with DMT

- Frequency $f$ → tone index $n$.




See also Section 2.7.4.2 and Lecture 8.

1. Find all noise and crosstalk:
   - $R_{noise}(u,n) = \sum_{i \neq u} H_{i,n} \cdot R_{\boldsymbol{XX}}(i,n) \cdot H_{i,n}^{*} + R_{\boldsymbol{NN}}(n)$

2. Create a noise-equivalent that includes all other users as noise (no order, all):
   - $\widetilde{H}_{u,n} = R_{noise}^{-1/2}(u,n) \cdot H_{u,n} = F_{u,n} \cdot \Lambda_{u,n} \cdot M_{u,n}^{*}$

3. Water-fill each user:
   - $\mathcal{E}_{u,l,n} + \frac{1}{g_{u,l,n}} = K_u \;\; \forall \, n, l$ with $g_{u,l,n} = \lambda_{u,l,n}^{2}$

4. Form resulting input autocorrelation matrices (energy distribution with $L_{x,u} = 1$)
   - $R_{\boldsymbol{XX}}^{o}(u,n) = M_{u,n} \cdot \text{Diag}\{\mathcal{E}_{u,n}\} \cdot M_{u,n}^{*} \;\; \forall \, n = 0, \ldots, \overline{N} - 1$

**With MT and Ly=1, $n \to \infty$**
**There is always an FDM SWF solution.**

# SWF.m versus macmax.m

```
>> help SWF
 function [Rxx, bsum , bsum_lin] = SWF(Eu, H, user_ind, 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 = # tones
    (equally spaced over (0,1/T) at N/T.
    if time-domain h, H = 1/sqrt(N)*fft(h, N, 3);
 user_ind  The start index for each user, in the same order as Eu
    The Lxu vector of each user's number of antennas is computed
    internally.  %     U is determined from user_ind
 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

 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 is modified version of one originally supplied by student
 Chris Baca
```

**Energy-Vector MAC**

```
function [Rxx, bsum , bsum_lin] = macmax(Esum, 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:
 Esum The sum-user energy/SAMPLE scalar in time-domain.
     This will be increased by cb*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 computed 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 FFT size (equally spaced over (0,1/T) at 1/(NT).
 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
    FREQUENCY DOMAIN
 bsum the maximum rate sum, when cb=2, this is effectively over lower half
 of tones, or equivalently the 1/2*log2 form of data rates are summed
 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 MAC**

**Up to N on h**

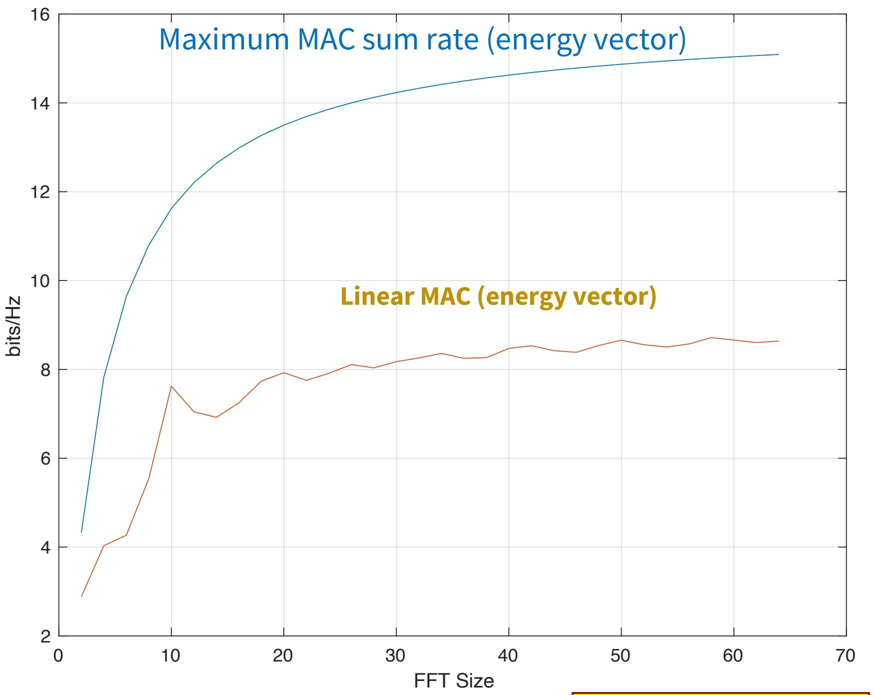
**This & bcmax updated at Website.**

- SWF is frequency domain input (useful with non-white noise psd), has separate user energies, and uses no CVX.
- macmax is time-domain (and uses Lxu instead of user_ind), has single sum energy, and uses CVX.

# Max rate-sum Example

```
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;
bsum=zeros(1,Nmax);
bsumlin=zeros(1,Nmax);

for index=1:Nmax
  i=2*index;   % (don't need to plot a point for every number of tones)
  H = fft(h, i, 3);
  Rnn=zeros(Ly,Ly,i);
    for n=1:i
    Rnn(:,:,n) = eye(2);
    end
  [Rxx, bsum(index), bsumlin(index)] = SWF(i/(i+3)*[1 1 1], H, [1  1 1], Rnn(:,:,:), 1);
  bsum(index)=bsum(index)/(i+3);
  bsumlin(index)=bsumlin(index)/(i+3);
end
bsum(32)*67= 1011.1 >  990.8
bsumlin(32)*67 = 578.8502
plot(2*[1:Nmax], bsum,2*[1:Nmax],bsumlin)
```




- Even with 3 users > 2 antennas, linear loses much.
  - ~ 20 dB (from "link budget)
- **Linear curve variation** is because $\overline{N}$ is finite and the simultaneous water filling is not necessarily best solution under linear restriction
- When would "linear receiver be best?"

**If vector-coding could be used, But not possible on MAC in general. Discrete slush-packing awaits…**

**The linear max-sum prob is not convex, See OSB in Sec 5.6**

# SWF energy/Rxx distribution

```
Rxx(:,:,1)
   1.4504   0       0
        0   0       0
        0   0  1.3050
Rxx(:,:,2) =
   1.4512   0       0
        0   0       0
        0   0  1.3120
Rxx(:,:,3) =
   1.4528   0       0
        0   0       0
        0   0  1.3274
.........
Rxx(:,:,9) =
   1.4419        0        0
        0   0.0670        0
        0        0   1.3303
Rxx(:,:,11) =
   1.3170        0        0
        0   1.0228        0
        0        0   0.5748
Rxx(:,:,15) =

   1.3889        0        0
        0   1.4116        0
        0        0   0.1513
Rxx(:,:,26) =
   0.1384        0        0
        0   1.4184        0
        0        0   1.3192
```

```
.........
Rxx(:,:,27) =
        0        0        0
        0   1.4939        0
        0        0   1.3767
Rxx(:,:,28) =
        0        0        0
        0   1.4885        0
        0        0   1.3761
Rxx(:,:,31) =

        0        0        0
        0   1.4229        0
        0        0   1.3689
Rxx(:,:,32) =
        0        0        0
        0   1.3394        0
        0        0   1.3606
Rxx(:,:,39) =
        0        0        0
        0   1.4939        0
        0        0   1.3767
Rxx(:,:,40) =
   0.1384        0        0
        0   1.4184        0
        0        0   1.3192
Rxx(:,:,51) =
   1.3889        0        0
        0   1.4116        0
        0        0   0.1513
Rxx(:,:,52) =
   1.3871        0        0
        0   1.3929        0
        0        0   0.1700
```

```
Rxx(:,:,53) =
   1.3678        0        0
        0   1.3329        0
        0        0    0.243
Rxx(:,:,57) =

   1.4419        0        0
        0   0.0670        0
        0        0   1.3303
...........
Rxx(:,:,58) =
   1.4577        0        0
        0        0        0
        0        0   1.3736
Rxx(:,:,59) =
   1.4573   0       0
        0   0       0
        0   0  1.3693
Rxx(:,:,60) =
   1.4567        0        0
        0        0        0
        0        0   1.3633
Rxx(:,:,61) =
   1.4557        0        0
        0        0        0
        0        0   1.3548
Rxx(:,:,62) =
   1.4544        0        0
        0        0        0
        0        0   1.3428
Rxx(:,:,63) =
   1.4528        0        0
        0        0        0
        0        0   1.3274
```

$\wp_{H,n} < U$ ??

With $\bar{N} > 1$ , there can be some tones that use all 3 dimensions. **Sep Theorem** applies only over a single user's tones.

These are equivalent to time-shared (dimension shared) of 2-user-only tones,
but with same order on all tones.

Freq sharing cannot happen when $\bar{N} = 1$ & $\wp_H < U$.

**Also Ly > 1,**
**So FDM not assured.**

# macmax (new at website, also bcmax)

```
[RxxEsum, bsumEsum , bsum_linEsum] = macmax(3*64/67, h, [1 1 1], 64 , 1);

bsumEsum = 1011.3 > 1011.2                    (just slightly)
bsum_linEsum = 571.7289 < 578.85              (no guarantee that linear version is best)


>> sum(real(Rxx),3) =
 61.1343        0        0
       0   61.1343        0
       0        0   61.1343
>> 64^2/67 =   61.1343 checks on each dimension
>> trace(sum(real(Rxx),3)) =  183.4030 (clearly 3x single dimensional energy)
```

**Any order can be used for same rate sum (in this case maximum)**

**But, the same order needs to be used on ALL tones.**

**Secondary user COMPONENTS are 0.**

**(there is no additional time-sharing of two vertices for a SWF/macmax point)**

- The RxxEsum are very similar to those from SWF.m.

- While many tones individually zero one user (consistent with secondary-component concept).
  - It is not the same user for all such tones.
  - Some tones energize all 3 users.

- $\sum_n \wp_{H,n} > U$ , significantly so. This means there is effectively dimension-sharing occurring over the 64 tones, at least for the rate-sum max.

# Gain is larger when crosstalk is larger

- Binders of copper wires (think ethernet or your/neighbors' cable of telephone wires) crosstalk.
  - Highly variable with twisting (even measuring point can lead to 20dB or more variation if moved an inch or two).
  - Probabilistic models (like wireless' distributions) are also used.
  - Average xtalk is larger on SHORTER wires because xtalk coupling increases with frequency.
    - Shorter wires use higher frequencies that are less attenuated

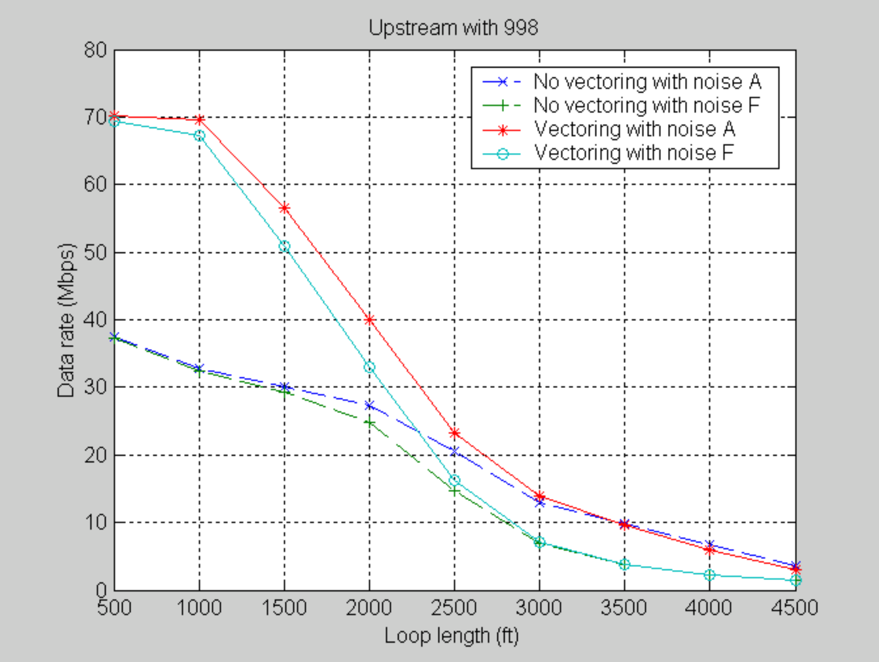
- Example is vectored VDSL (upstream MAC).

- Each user has its own "link" that terminates (upstream) on a common receiver – by default all primary users (no time-sharing needed).
  - "perfect massive MIMO" all (used) tones (plot is for 25 links)
  - Can see up to *U=384* links vectored (predates "massive MIMO" in invention and use by 10 years).

- The GDFE cancels the crosstalk.

- It exhibits diagonal dominance too. Why?
  - So typicaly no feedback section is used.

- Actually, some "mgfast" (ITU G.9711) to 5 Gbps multiuser and 10 Gbps "fastback" (ITU G.9702), 2 pairs – 3 channels, single user can use some GDFE's at lowest frequencies.



# Wireless – uplink Cellular or Wi-Fi

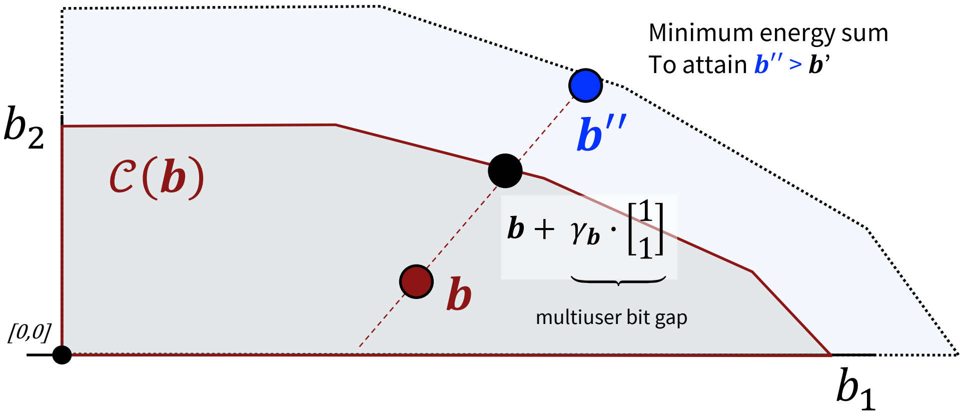
- The C-OFDM systems are as in Lecture 6 (Wi-Fi and cellular).
- A single MAC receiver uses common FFTs (one for each, max #, of spatial stream/dimension with MIMO).
- They share common frequencies.
- Usually, no feedback sections are used ... yet, so linear setting from computeGDFE provides performance.

**OK – All good, but what is $R_{xx}(u)$ when we don't maximize a rate sum??**

# Designs with weighted sums

## Section 5.4.3

# Capacity region(s)




- $\mathcal{C}(\boldsymbol{b})$ contains all possible weighted **rate** sums $\sum_{u=1}^{U} \theta_u \cdot b_u$ that meet **energy-vector** constraint $\boldsymbol{\mathcal{E}} \preccurlyeq \boldsymbol{\mathcal{E}}_x$.
- The max-b-sum point is "highest" (tangent to plane $\mathbf{1}^t \cdot \boldsymbol{b}$) with $\boldsymbol{b}$ in $\mathcal{C}(\boldsymbol{b})$, **but we want another $\boldsymbol{b}$**!
- If $\boldsymbol{\mathcal{E}} \preccurlyeq \boldsymbol{\mathcal{E}}_x$, **then $\boldsymbol{b}$ is admissible.**
- If not, a design might still target the minimum energy-sum that achieves $\boldsymbol{b}$.

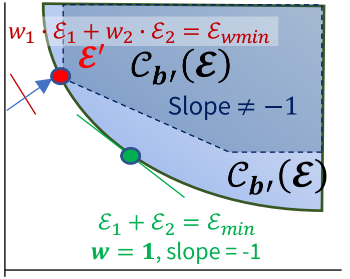
# minimum weighted energy sum

- **Minimize weighted energy** sum $\mathcal{E}_{\boldsymbol{x}}$ , given $\boldsymbol{b}$
  - No energy limit (beyond minimum)
  - Energy weights $\boldsymbol{w}$ given, non-negative

$$\min_{\{R_{\boldsymbol{xx}}(u)\}} \sum_{u=1}^{U} w_u \cdot \text{trace}\underbrace{\{R_{\boldsymbol{xx}}(u)\}}_{\mathcal{E}_u}$$

$$ST: \quad \boldsymbol{b} \succeq [b_{1,min}\ b_{2,min}\ \ldots b_{U,min}]^* = \boldsymbol{b}^*_{min} \succeq \boldsymbol{0}$$

$$\boldsymbol{\mathcal{E}} \succeq \boldsymbol{0} \quad .$$

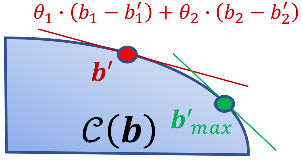



- **Maximize weighted rate** sum $b$, given $\boldsymbol{\mathcal{E}}$
  - No rate limit
  - rate weights $\boldsymbol{\theta}$ given, non-negative

$$\max_{\{R_{\boldsymbol{xx}}(u)\}} \sum_{u=1}^{U} \theta_u \cdot b_u$$

$$ST: \quad \boldsymbol{\mathcal{E}}_{\boldsymbol{x}} \preceq [\mathcal{E}_{1,max}\ \mathcal{E}_{2,max}\ \ldots \mathcal{E}_{U,max}]^* = \boldsymbol{\mathcal{E}}^*_{max} \succeq \boldsymbol{0}$$

$$\boldsymbol{b} \succeq \boldsymbol{0} \quad .$$



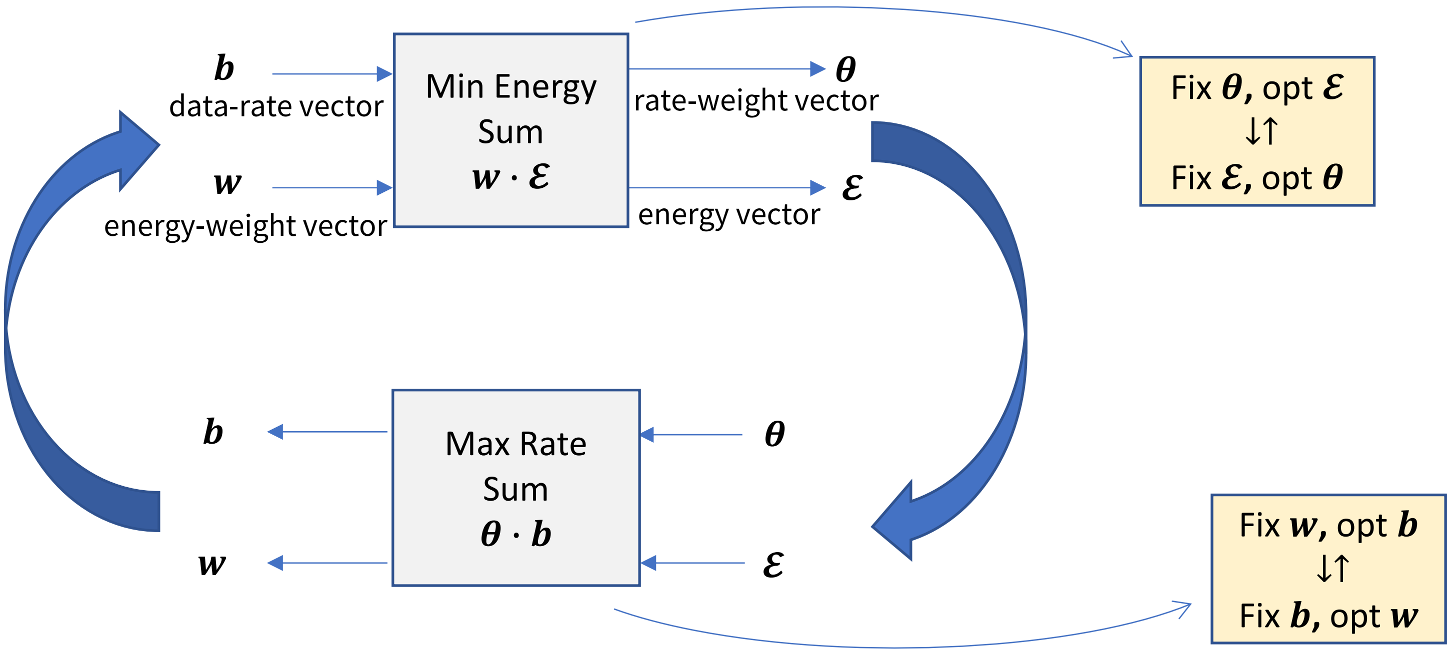

max rate sum; $\boldsymbol{\theta} = \boldsymbol{1}$

- These are "dual" problems

$$L_{minE}(R_{\boldsymbol{xx}}, \boldsymbol{b}, \boldsymbol{w}, \boldsymbol{\theta}) = \max_{\boldsymbol{\theta}} \min_{R_{\boldsymbol{xx}}} \underbrace{\sum_{u=1}^{U} [w_u \cdot trace\{R_{\boldsymbol{xx}}(u)\} + \theta_u \cdot b_u}_{\text{common term}} \quad - \theta_u \cdot b_{min,u}]$$

$$L_{maxR}(R_{\boldsymbol{xx}}, \boldsymbol{b}, \boldsymbol{w}, \boldsymbol{\theta}) = \min_{\boldsymbol{w}} \max_{R_{\boldsymbol{xx}}} \underbrace{\sum_{u=1}^{U} [w_u \cdot trace\{R_{\boldsymbol{xx}}(u)\} + \theta_u \cdot b_u}_{\text{common term}} \quad - w_u \cdot \mathcal{E}_{max,u}]$$

# Basic Solution Cycles




- Each of these "boxes" (subnetworks) can be intense calculation, but (~, see L14: 31) convex and convergent.
- The overall recursive cycling also converges if $\boldsymbol{b} \in \mathcal{C}(\boldsymbol{b})$.

**Use Convex, ML/AI methods ...**

# Tonal Lagrangian

- Minimize (over $R_{\boldsymbol{xx}}(u)$) weighted sum at any given (think temporary) $\boldsymbol{\theta}$ where $\boldsymbol{b}$ and $\boldsymbol{w}$ are the specified values:

$$L(R_{\boldsymbol{XX}}, \boldsymbol{b}, \boldsymbol{w}, \boldsymbol{\theta}) = \sum_{n=0}^{\bar{N}-1} \left\{ \underbrace{\sum_{u=1}^{U} \left[ w_u \cdot \text{trace}\left\{ R_{\boldsymbol{XX}}(u,n) \right\} - \sum_{u=1}^{U} \theta_u \cdot b_{u,n} \right]}_{L_n(R_{\boldsymbol{XX}}(n), \boldsymbol{b}_n, \boldsymbol{w}, \boldsymbol{\theta})} \right\} + \theta_u \cdot b_u \quad ,$$

With fixed $\boldsymbol{\theta} \succcurlyeq \boldsymbol{0}$ each tone can be individually minimized

- which produces then for tone $n$:

$$L_{min}(\boldsymbol{\theta}, n) \triangleq \min_{\{R_{\boldsymbol{XX}}(u,n)\}, b_{u,n}} L_n(R_{\boldsymbol{XX}}(n), \boldsymbol{b}_n, \boldsymbol{w}, \boldsymbol{\theta}) \quad .$$

- Then, max over $\boldsymbol{\theta}$

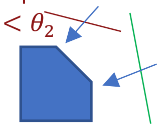
$$L^* = \max_{\boldsymbol{\theta}} \sum_{n=0}^{\bar{N}-1} L_{min}(\boldsymbol{\theta}, n) \triangleq \max_{\boldsymbol{\theta}} L_{min}(\boldsymbol{\theta}) \quad ,$$

- and satisfy tonal GDFE (achievable region) constraint

$$\boldsymbol{b}_n \in \left\{ \boldsymbol{b}_n \mid 0 \le \sum_{\boldsymbol{u} \subseteq \boldsymbol{U}} b_{u,n} \le \log_2 \left| \left( \sum_{u=1}^{U} \tilde{H}_{u,n} \cdot R_{\boldsymbol{XX}}(u,n) \cdot \tilde{H}^*_{u,n} \right) + I \right| \right\} = \mathcal{A}_n\left( \{R_{\boldsymbol{XX}}(n)\}, \bar{H}_n \right) .$$

(5.2

# The tonal achievable-region constraint




- Maximum $\sum_{u=1}^{U} \theta_u \cdot b_{u,n}$ occurs at $\mathcal{A}_n(R_{\boldsymbol{XX}}(u,n), \boldsymbol{b}_n)$ vertex (think slope -1 line and pentagon),
  - Given $R_{\boldsymbol{XX}}(u,n) \to R_{\boldsymbol{XX}}(u)$ ; equivalently max $\sum_{u=1}^{U} \theta_u \cdot b_u$ occurs at $\mathcal{A}\ (R_{\boldsymbol{XX}}(u), \boldsymbol{b}\ )$ vertex.
- That max-weighted-sum vertex has specific $\theta_u$, which must satisfy $\theta_{\pi^{-1}(U)} \geq \theta_{\pi^{-1}(U-1)} \geq \ldots \geq \theta_{\pi^{-1}(1)}$.
  - Proof see Theorem 5.4.1, and also discussion in Sec 5.4.4.1. (Same order all tones)
  - Alternative to testing all orders – optimum order is inferred from the (converged) real vector $\theta$.
- The user data rates in $\mathcal{A}_n(R_{\boldsymbol{XX}}(u,n), \boldsymbol{b}_n)$ must satisfy the (sum of) **tonal-GDFE constraint(s):**

$$b_{u,n} = \log_2\left\{\frac{|R_{yy}(u,n)|}{|R_{yy}(u-1,n)|}\right\} = \log_2\left|\sum_{i=1}^{u} \tilde{H}_{\pi^{-1}(i),n} \cdot R_{\boldsymbol{XX}}(\pi^{-1}(i),n) \cdot \tilde{H}^*_{\pi^{-1}(i),n} + I\right| - \log_2\left|\sum_{i=1}^{u-1} \tilde{H}_{\pi^{-1}(i),n} \cdot R_{\boldsymbol{XX}}(\pi^{-1}(i),n) \cdot \tilde{H}^*_{\pi^{-1}(i),n} + I\right|$$

- For given $\boldsymbol{\theta}$, min weighted rate sum over $R_{\boldsymbol{XX}}(u,n)$ minimizes convex sum

$$\sum_{u=1}^{U} \theta_u \cdot b_{u,n} = \sum_{u=1}^{U}\left\{\underbrace{\left[\theta_{\pi^{-1}(u)} - \theta_{\pi^{-1}(u+1)}\right]}_{\delta_{\pi^{-1}(u)} \leq 0} \cdot \log_2\left|\sum_{i=u}^{U} \tilde{H}_{\pi^{-1}(i),n} \cdot R_{\boldsymbol{XX}}(\pi^{-1}(i),n) \cdot \tilde{H}^*_{\pi^{-1}(i),n} + I\right|\right\} .$$

# Equal Theta

- Successive equal theta values
  - can happen, often!
  - This usually happens when there are secondary user components.
- The  corresponding rate-sum difference term(s) is (are) zero.
- Only the sum rate of the corresponding users can be varied $b_{\pi^{-1}(u)} + b_{\pi^{-1}(u)+1}$ is optimized.
- The corresponding vertices for swapping the order (more generally varying when 3 or more) need to be "vertex-shared" in a proportion that causes the desired data rate to be achieved.
- Complicated program in matlab to do this, although concept is straightforward.

**Coming Attraction: The Stanford minPMAC program(s)**

# End Lecture 14

# Two iterated steps

- $R_{XX}(u,n)$ step: With the given (current) $\boldsymbol{\theta}$ , $\boldsymbol{w}$, $\{b_{u,n}\}$ , minimize the (neg) weighted rate sum over $R_{XX}(u,n)$
  - Each tone separately and sum

$$L_{min}(\boldsymbol{\theta}\,, n) = \underbrace{\sum_{u=1}^{U}\left[w_u \cdot \text{trace}\left\{R_{\boldsymbol{XX}}(u,n)\right\} - \sum_{u=1}^{U}\theta_u \cdot b_{u,n}\right]}_{L_n(R_{\boldsymbol{XX}}(n),\boldsymbol{b}_n,\boldsymbol{w},\boldsymbol{\theta})}$$

e.g. $$L_{k+1} = L_k - \mu \cdot \left(\nabla^2 L_k\right)^{-1} \cdot \nabla L_k$$ Weighted steepest descent ("Newton")

- Order step: With the given (current) $R_{XX}(u,n)$, $\boldsymbol{w}$, $\{b_{u,n}\}$ , maximize the Lagrangian over $\boldsymbol{\theta}$

$$L\,(\boldsymbol{\theta}) = \sum_{n=1}^{\bar{N}} L_{min}(\boldsymbol{\theta}\,, n)$$

Initialize (first time only) with FM SWF for given $\boldsymbol{b}$

This is the "find the vertex set" – elliptic algorithm, see text