STANFORD

Lecture 14

MAC GDFEs and Design Measures

May 22, 2023

JOHN M. CIOFFI

Hitachi Professor Emeritus of Engineering

Instructor EE392AA – Spring 2023

Announcements & Agenda

- Announcements
  - PS7 – last homework – nominally due end of week before dead week, but accepted on Monday also
    - Basically have 2 weeks from today, counts double
  - Section 5.4
  - 2023: will skip L13's (ZF/MMSE Convergence p 31-35),

- Problem Set 7 = PS7 (due May 28)
  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
  - 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)

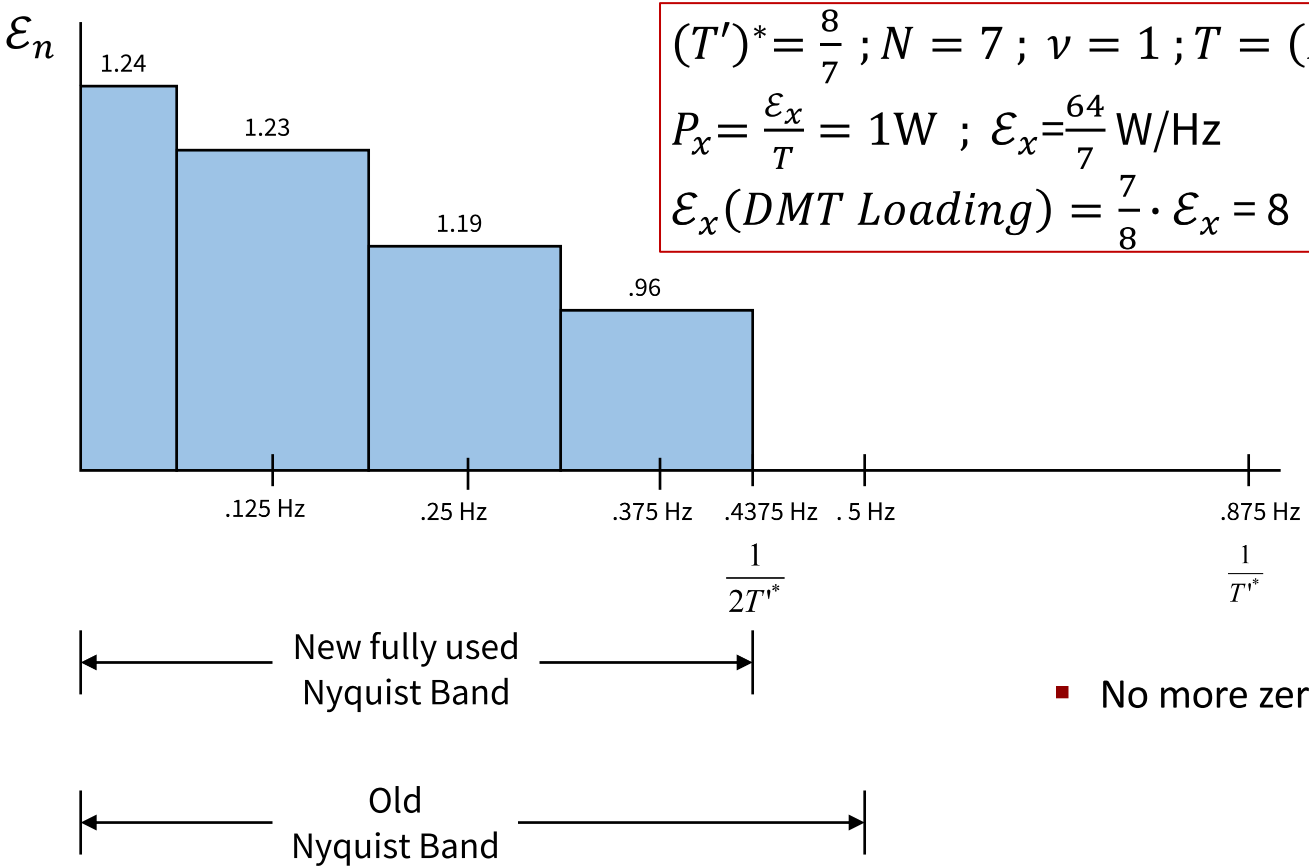
L13 conclusion

Resampled 1+.9D^{-1} Another way to match WF to singularity

$(T')^* = \frac{8}{7}\ ; N = 7\ ;\ \nu = 1\ ; T = (N+\nu)\cdot(T')^* = \frac{64}{7};$

$P_x = \frac{\mathcal{E}_x}{T} = 1\text{W}\ ;\ \mathcal{E}_x = \frac{64}{7}\ \text{W/Hz}$

$\mathcal{E}_x(DMT\ Loading) = \frac{7}{8}\cdot\mathcal{E}_x = 8\ \text{W/Hz}$



- No more zeroed bands

Resampled Design Matlab Commands

```
>> D=exp(j*[0:100]*(7/8)*.01*pi);
>> H7=sqrt(8/7)*(ones(1,101)+.9*D);
>> H7=[H7,conj(H7(101:-1:2))];
>> h7=real(ifft(H7));
>> h=[h7(200:201),h7(1:5)] =
  -0.1011   0.9393   1.1979  -0.0603   0.0394  -0.0292   0.0232
>> H=toeplitz([h(1),h(7:-1:2)]',h) =
  -0.1011   0.9393   1.1979  -0.0603   0.0394  -0.0292   0.0232
   0.0232  -0.1011   0.9393   1.1979  -0.0603   0.0394  -0.0292
  -0.0292   0.0232  -0.1011   0.9393   1.1979  -0.0603   0.0394
   0.0394  -0.0292   0.0232  -0.1011   0.9393   1.1979  -0.0603
  -0.0603   0.0394  -0.0292   0.0232  -0.1011   0.9393   1.1979
   1.1979  -0.0603   0.0394  -0.0292   0.0232  -0.1011   0.9393
   0.9393   1.1979  -0.0603   0.0394  -0.0292   0.0232  -0.1011
>> H=sqrt(1/.181)*H;
>> J7=hankel([zeros(1,6),1]');
>> Q7=(1/sqrt(7))*J7*fft(J7);
>> rXX=diag([1.23,1.19,.96,.96,1.19,1.23,1.24]);
>> rxx=real(Q7'*rXX*Q7);
>> Phibar=lohc(rxx);
>> A=Phibar;
```

```
>> [snrGDFEu, GU, WU, S0, MSWMFU, b, bbar] = computeGDFE(H, A, 2, 8)

snrGDFEu =   9.1416 dB (higher, but at lower symbol rate)
GU =
   1.0000   0.4783  -0.0492  -0.0074   0.0208  -0.0760   0.4583
        0   1.0000   0.5663  -0.0507  -0.0100   0.0385  -0.2577
        0        0   1.0000   0.5952  -0.0517  -0.0208   0.1470
        0        0        0   1.0000   0.6049  -0.0463  -0.0833
        0        0        0        0   1.0000   0.6042  -0.0105
        0        0        0        0        0   1.0000   0.6074
        0        0        0        0        0        0   1.0000
MSWMFU =
  -0.0173   0.0040  -0.0050   0.0068  -0.0103   0.2054   0.1611
   0.1971  -0.0222   0.0069  -0.0089   0.0125  -0.1032   0.1701
   0.1583   0.2097  -0.0250   0.0095  -0.0127   0.0677  -0.0866
  -0.0808   0.1539   0.2145  -0.0268   0.0118  -0.0445   0.0590
   0.0559  -0.0786   0.1520   0.2166  -0.0283   0.0302  -0.0397
  -0.0401   0.0562  -0.0788   0.1522   0.2163  -0.0258   0.0308
   0.0724  -0.0769   0.0956  -0.1281   0.2194   0.1136  -0.0825

>> b' =
   0.9313   0.8775   0.8700   0.8691   0.8690   0.8697   0.7970
>> bbar =   1.6013
>> R=bbar*(7/8) = 1.4718   (bits/sec) > 1.3814 bits/sec  (same as interp)
```

Also
7 Dimensions

Converges On GU

(and WU Not shown)

- See also the two-band example in Section 5.3
  - Tedious but could be helpful in following details for a multiband CDFE (e.g. – uplink carrier aggregation with multiple resource blocks in Cellular)

Some Final Comments

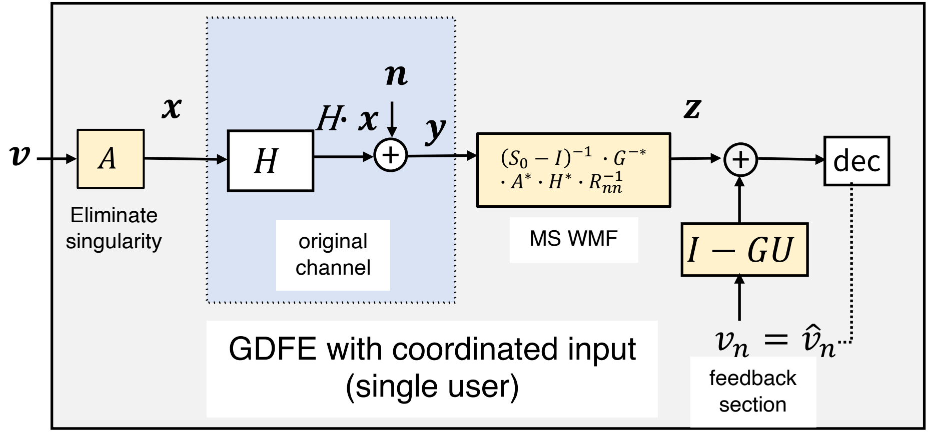
- The GDFE is canonical – capacity rate is reliably achievable with $\Gamma = 0$ (or capacity less shaping loss)
- GDFE can have error propagation (limited to $\bar{N}$) if $\Gamma > 0$ dB
  - Unless it is VC (~DMT) , which is ML decoder uniquely amoung all GDFEs
  - Other GDFE's becoming increasingly 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 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
  - 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

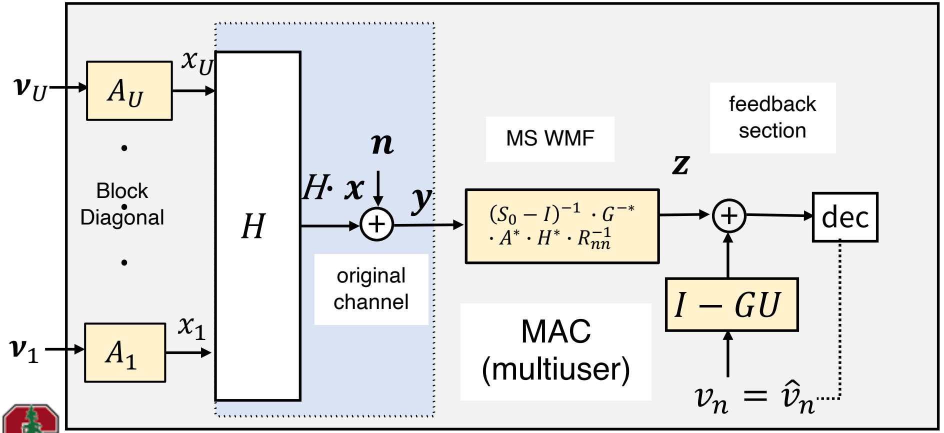
😅

MAC and GDFE Comparison

The MMSE MAC vs MMSE GDFE



- GDFE
  - Designed for "single-user" H ; $A = R_{\boldsymbol{xx}}^{1/2}$
  - MMSE: $R_b^{-1} = R_{\boldsymbol{xx}}^{-*/2}\cdot H^*\cdot R_{\boldsymbol{nn}}^{-1}\cdot H\cdot R_{\boldsymbol{xx}}^{-1/2} + I = G\cdot S_0\cdot G^*$
  - Canonical decision feedback (decisions correct)
  - Input restriction $trace\{R_{\boldsymbol{xx}}\} \le \mathcal{E}_x$
  - Rate does not depend on dimensional order (no err prop)



- MAC
  - Designed for block-diag $trace\{R_{\boldsymbol{xx}}\} \le \mathcal{E}_x$
    - only in energy-sum case
  - Usually has input energies $trace\{R_{\boldsymbol{xx}}(u)\} \le \mathcal{E}_u$; separation locations ; $A_u = R_{\boldsymbol{xx}}^{1/2}(u)$; $A = R_{\boldsymbol{xx}}^{1/2}$
  - MMSE: $R_b^{-1} = R_{\boldsymbol{xx}}^{-*/2}\cdot H^*\cdot R_{\boldsymbol{nn}}^{-1}\cdot H\cdot R_{\boldsymbol{xx}}^{-1/2} + I = G\cdot S_0\cdot G^*$
  - Canonical decision feedback (decisions correct)
    - by user – order is more important

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+ L(1)^2) =   6.64 > 6.144
```

- But 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 $R_{xx} = \mathbf{[1\ 0\ ;0\ 0]}$ 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 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

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

MAC Loss

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

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

- For the previous example $\gamma_{MAC} = \frac{2^{6.64}-1}{2^{6.144}-1} = 1.5 \text{ dB}$

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

See also *split-dimensionality* example in Section 5.4.1

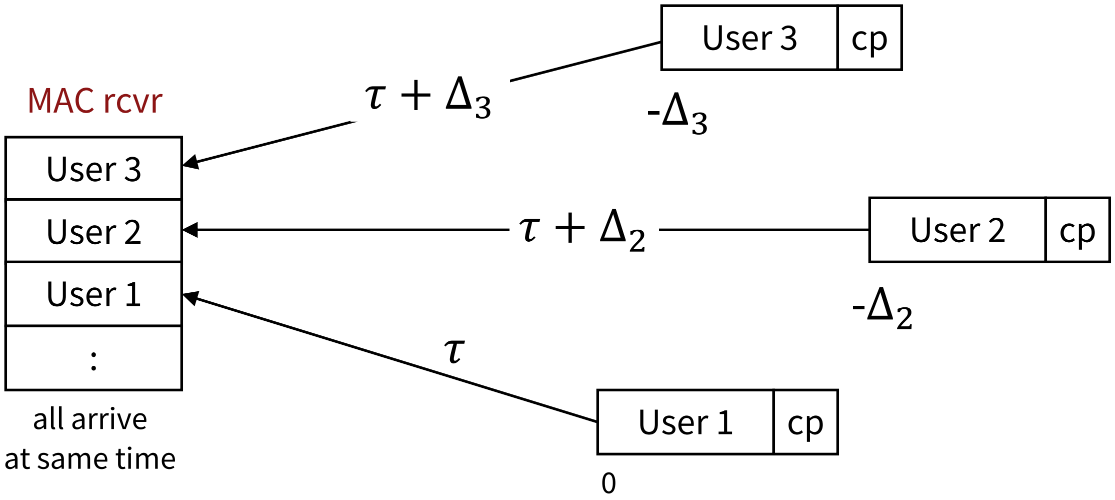
Tonal MAC (with DMT)

Section 5.4.2

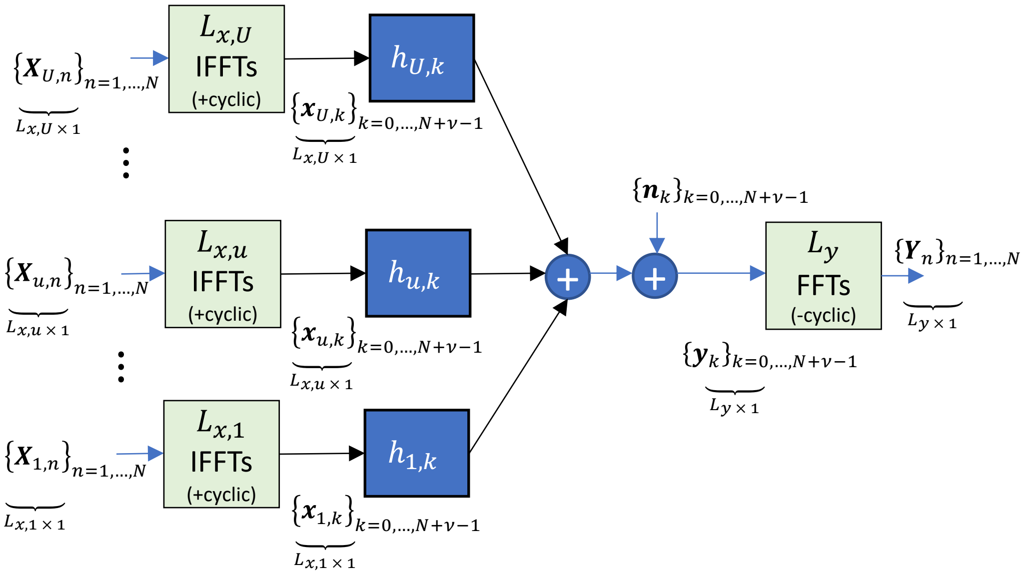
Note Section 5.4.1’s two-user ISI-GDFE is interesting, but largely becomes superfluous with the tonal vector-DMT system.

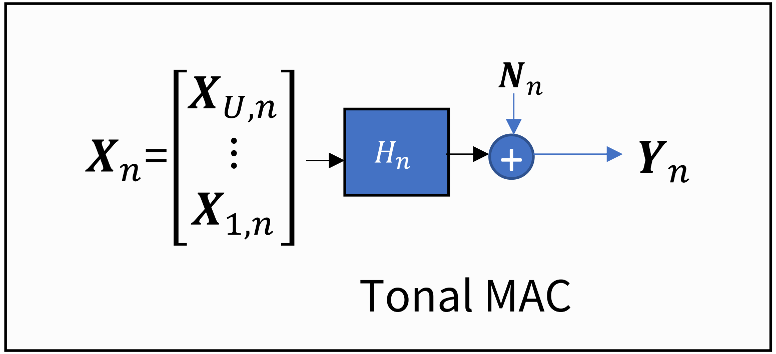
Align Receiver DMT Symbols for MAC

- 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





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

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

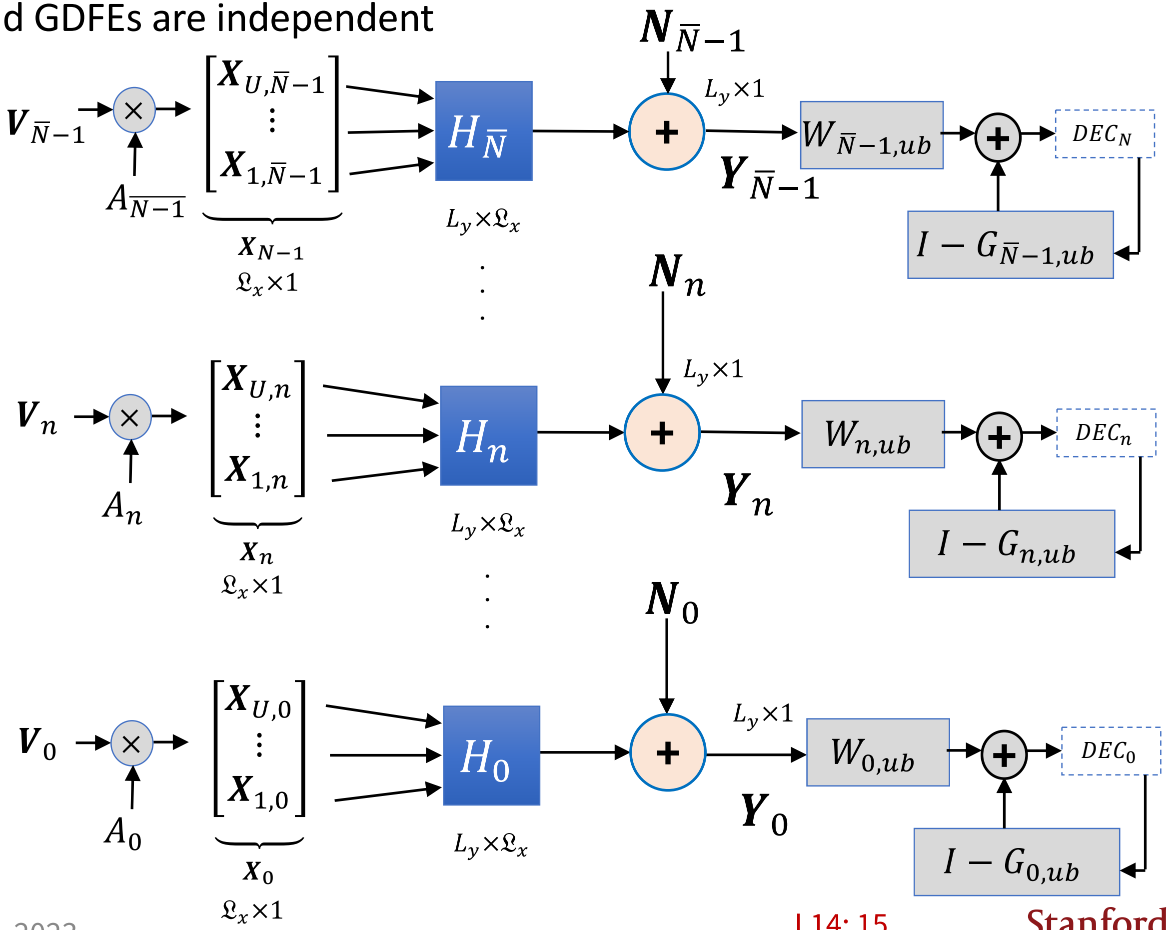
Esum−MAC: $\sum_{u=1}^{U} \sum_{n=0}^{\bar{N}} trace\,\{R_{XX}(u,n)\} \le \mathcal{E}_x$

- Symbol boundaries align through cyclic-extension (guard period) uses (even with different channel delays)
- Basically, an IFFT per transmit-antenna-user
- Discrete version of the MT MAC, indeed all SVD's, Cholesky's, and QR factorizations become "frequency-dependent" (in limit)

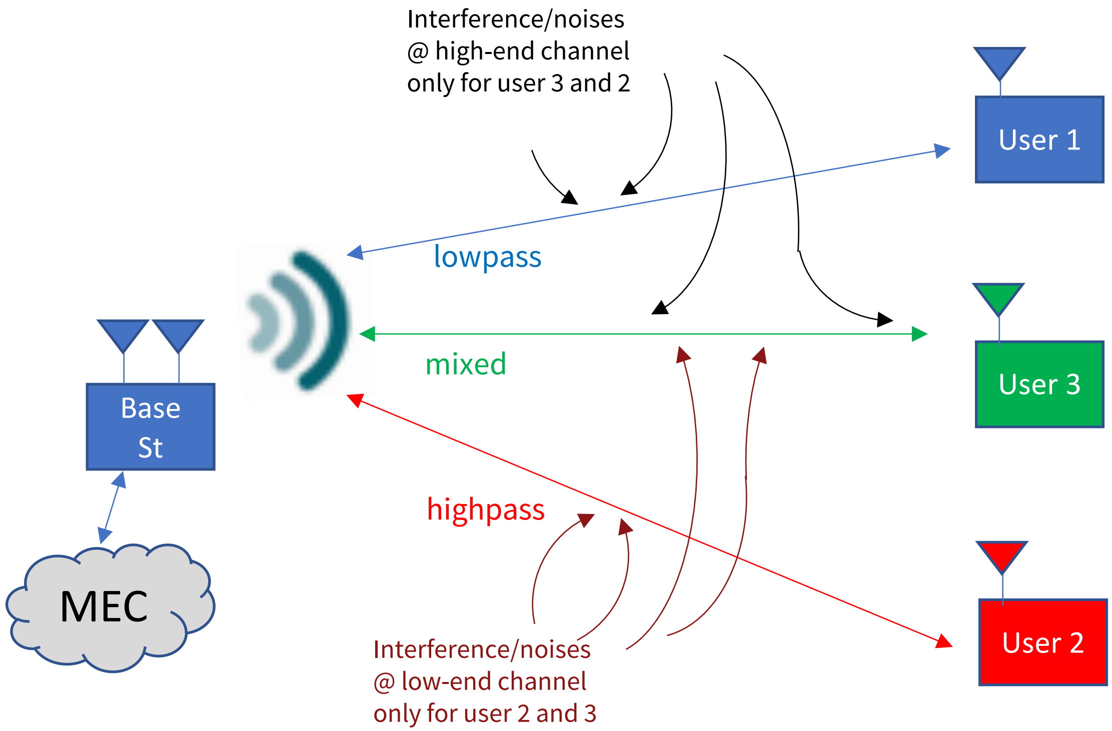
Tonal GDFEs with MAC

- Discrete tonal modulators and GDFEs are independent
- Put an "n" index on all the GDFE design equations

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



Example complex BB channel



More users than antennas

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

- Illustrates many effects

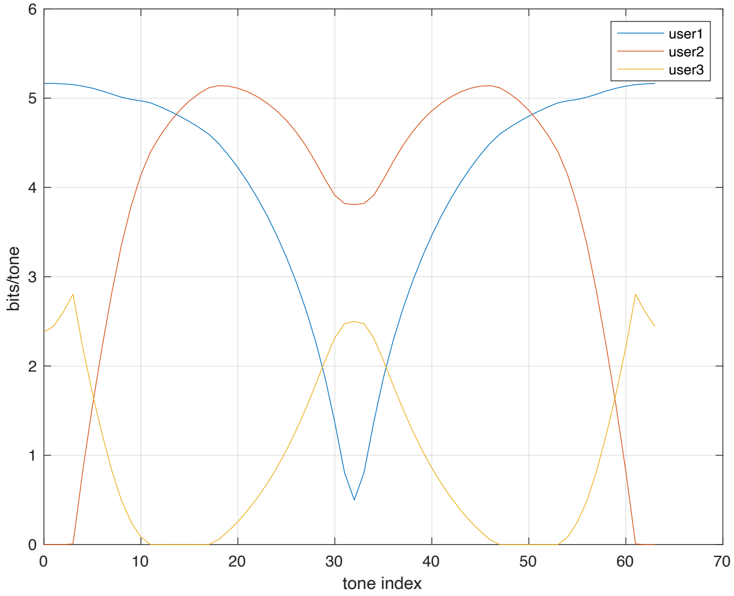
$$H(D) = \begin{bmatrix} 1+.9\cdot D & -.3\cdot D+.2\cdot D^2 & .8 \\ .5\cdot D-.4\cdot D^2 & 1-D-.63\cdot D^2+.648\cdot D^3 & 1-D \end{bmatrix}$$

A bit oversimplified, as real only for cplex channel

Example has ISI and MIMO together

- 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 cacluations for L14:17

White-Input Tonal GDFE

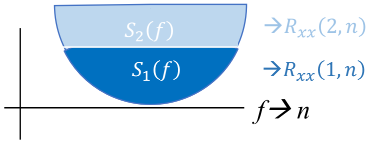
```
Nmax=32;
U=3;
Ly=2;
cb=1;
Lxu=[1 1 1];
bsum=zeros(1,Nmax);
for index=1:Nmax
 i=2*index;
 H = fft(h, i, 3);
 GU=zeros(U,U,i);
 WU=zeros(U,U,i);
 S0=zeros(U,U,i);
 Bu=zeros(U,i);
 MSWMFU=zeros(U,Ly,i);
 AU=zeros(3,3,i);
 for n=1:i
  AU(:,:,n)=sqrt(i)/sqrt(i+3)*eye(3);
 end
 for n=1:i
  [Bu(:,n), GU(:,:,n) , 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

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

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

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

- 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
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(Eu, h, Lxu, N , cb)

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

 This program uses the CVX package

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

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

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

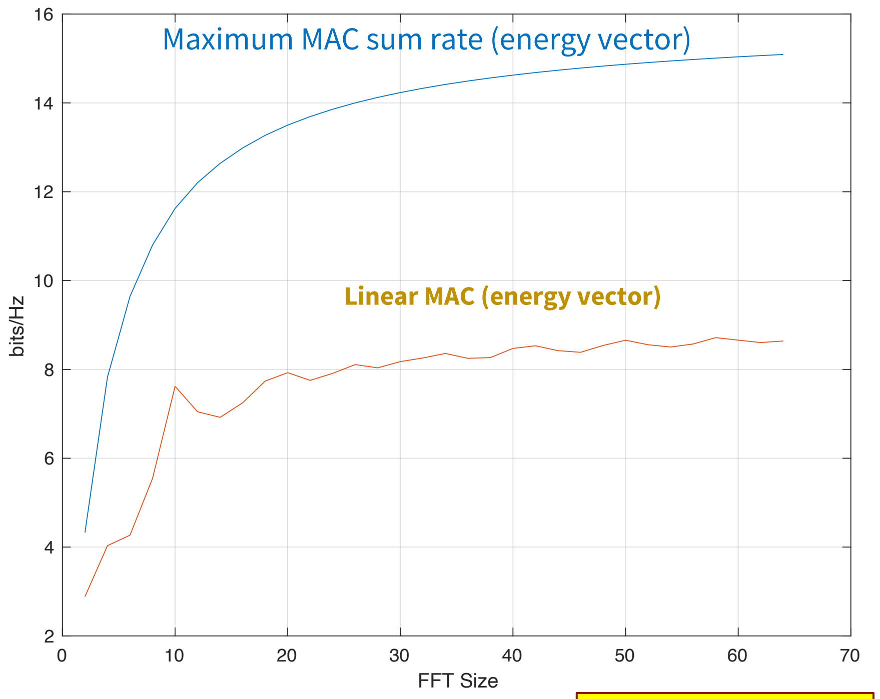
Energy-Sum MAC

Up to N on h

- SWF is frequency domain input (useful with non-white noise psd) and uses no CVX
- while macmax is time-domain (and uses Lxu instead of user_ind) 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

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

These are equivalent to time-shared (dimension shared) of 2-user-only tones

Cannot happen when $\bar{N} = 1$ & $\wp_H < U$

Also Ly > 1,
So FDM not assured

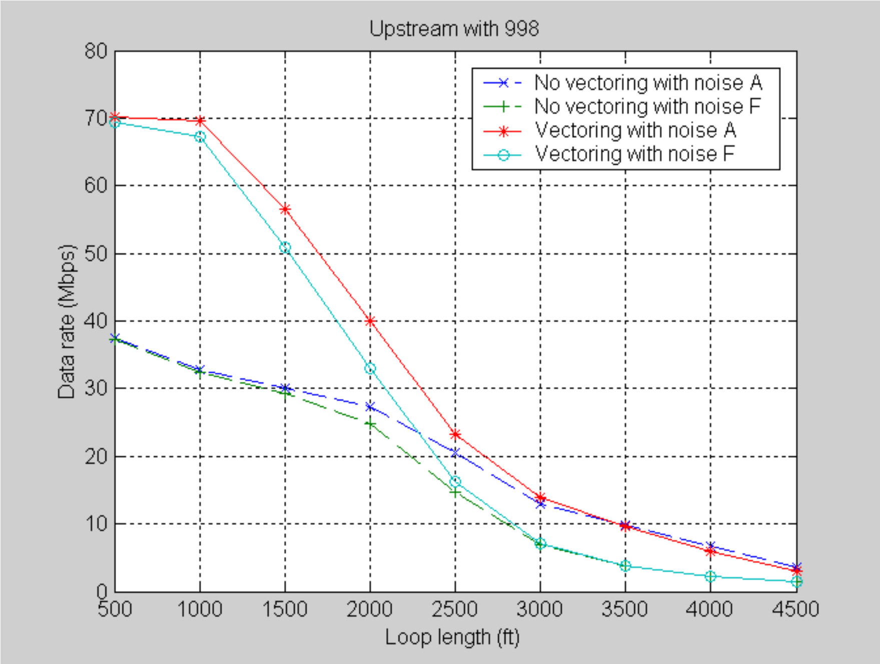
Macmax

```
[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)
```

- The RxxEsum are very similar to those from SWF.m
- While many tones individually zero one user (consistent with secondary-component concept)
  - Not the same user for all such tones
  - Some 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 neighbor 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)
  - Probabalistic models (like wireless' distributions) used
  - Average xtalk is larger on SHORTER wires
    - Why?
    - They 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 typically no feedback section 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

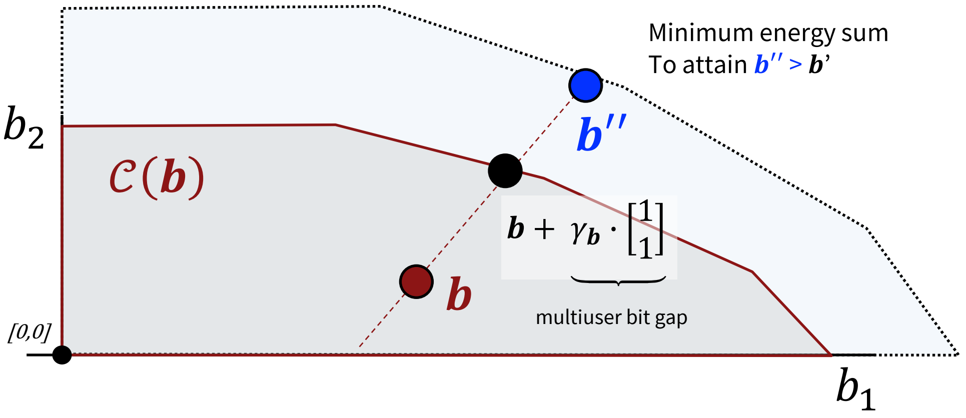
- The C-OFDM systems are as in Lecture 6
- Common FFTs in the single MAC receiver (one for each spatial stream/dimension with MIMO)
- They share common frequencies
- Usually no feedback sections used …. Yet, so linear setting from computeGDFE provides performance

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 $\mathcal{E} \preccurlyeq \mathcal{E}_x$
- 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 $\mathcal{E} \preccurlyeq \mathcal{E}_x$, **then $\boldsymbol{b}$ is admissible**
- If not, we might still desire the minimum energy-sum that achieves $\boldsymbol{b}$

minimum weighted energy sum

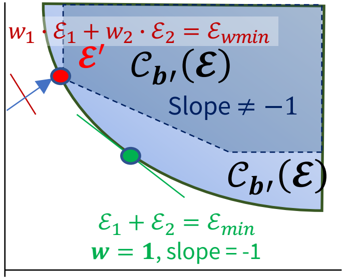
- **Minimize weighted energy** sum $\mathcal{E}_{\boldsymbol{x}}$, given $\boldsymbol{b}$
  - No energy limit (beyond minimum)
  - Energy weights $\boldsymbol{w}$ given, non-negative

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

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

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

$\mathcal{E}_2$

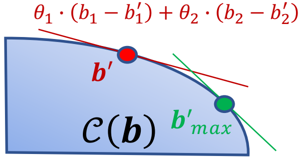


- **Maximize weighted rate** sum b, given $\boldsymbol{\mathcal{E}}$
  - No rate limit
  - rate weights $\boldsymbol{\theta}$ given, non-negative

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

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

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



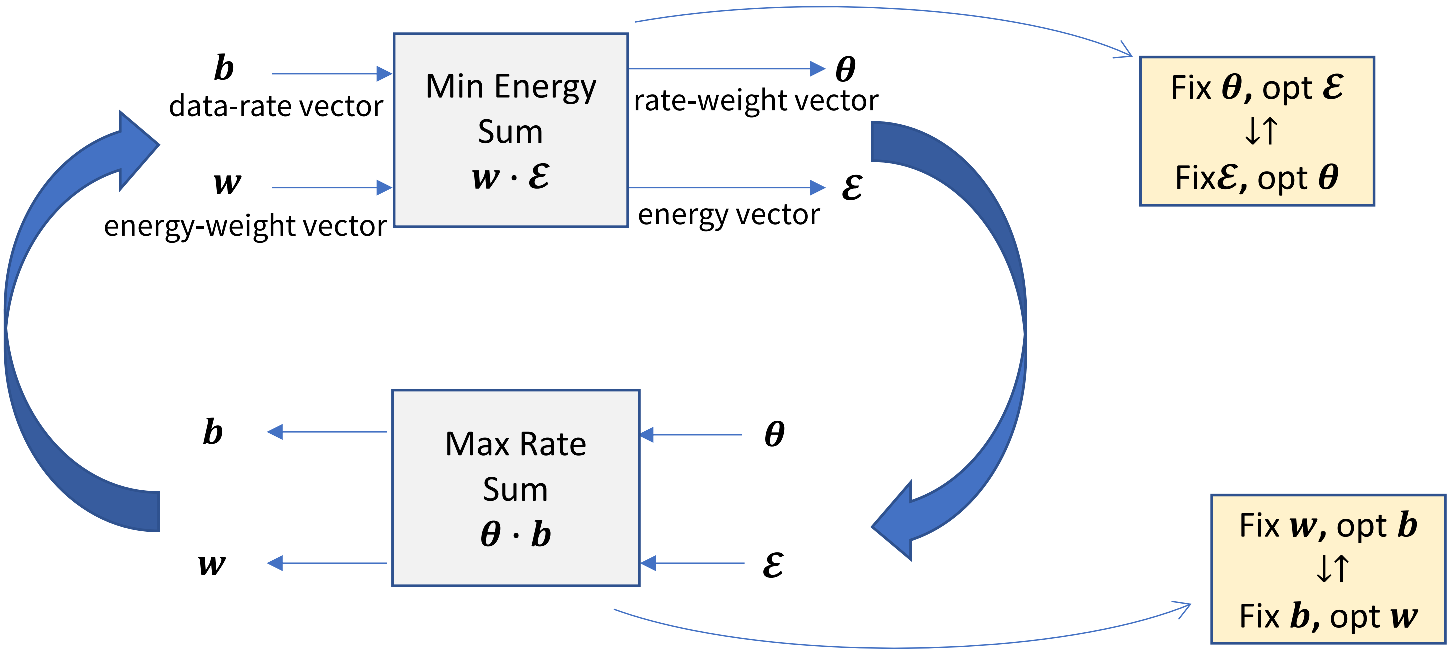
max rate sum; $\boldsymbol{\theta} = \boldsymbol{1}$

- 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 $\boldsymbol{b} \in \mathcal{C}(\boldsymbol{b})$

Potential Project – approximate this with ML/AI method?

Tonal Lagrangian

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

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

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

- Which produces then for tone n

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

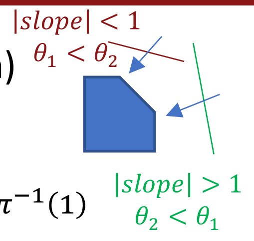
- Then max over $\boldsymbol{\theta}$

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

- and satisfy tonal GDFE (achievable region) constraint

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

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 $R_{\boldsymbol{XX}}(u,n) \rightarrow R_{\boldsymbol{XX}}(u)$; 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 θ_u , that 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)
  - **Don't need to test all orders – optimum order is inferred from the (converged) real vector θ !!**

- 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
  - This can happen often
  - 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)

STANFORD

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