Homework Help - Problem Set 1
Solutions

Capacity Calculation Help (2.15) This problem helps review some EE379A concepts on capacity, power-spectra (one and two-sided) and PAM/QAM as real/complex channels.

a. This part refreshes recollection that a flat (constant) power spectral density (2-sided) is equal to $\bar{\mathcal{E}}_{\boldsymbol{x}}$. Simply writing the two-sided and one-sided integrals for real baseband and complex baseband reinforces this concept easily. Recall that a bar over a quantity normalizes it to the number of **real** dimensions.

b. This follows by taking the same dimensionality in numerator and denominator for SNR; One can subtract the noise psd in dBm/Hz from the transmitted flat psd to obtain the SNR. They both must use the same number of dimensions.

c. The capacity in bits/real-dimension is $\frac{1}{2} \cdot \log_2(1 + SNR)$. The factor of 1/2 needs to be there for bits/REAL dimension.

d. This part asks you to work backwards from the 18 Mbps equal to capacity to find the power needed. This calculation follows more easily by determining first the symbol rate associated with PAM and QAM (QAM subsymbol rate is 1/2 that of PAM, but has twice as many dimensions/subsymbol). You should get the same power for both, emphasizing that PAM and QAM are essentially equivalent fundamentally.

e. This part also forces determination of a subsymbol rate for PAM and QAM, but now with a gap that is nonzero. You should get less than 72 Mbps because it is not a capacity-achieving code.

f. It is not Wi-Fi because the smallest Wi-Fi subsymbol rate is 20 MHz and your subsymbol-rate answer in the previous part is less than this. Understand that that real systems must attenuation near the edges of any channel, potentially by reducing the symbol rate so the spectrum is slightly narrowed. What kinds of wireless systems might have a symbol rate close to your answer in the previous part?

Gap Analysis (4.3) Before beginning with the problem itself, some general comments on Gap Analysis:

The gap Γ (in dB, $10 \cdot \log_{10}(\Gamma)$) is a design approximation that measures a specific AWGN design's proximity to best Shannon-Capacity performance:

$$\bar{b} = \frac{1}{2}\log_2(1 + \frac{SNR}{\Gamma}) \ , \ \ \bar{b} \geq 0.5 \ \ .$$

The gap depends on two design choices:

1. The code (class) used $C_{\boldsymbol{x}}$
2. The target error probability $\bar{P}_e$.

Thus, specifically, it does not depend on the data rate ($\bar{b}$) directly and is **constant** if the code and $\bar{P}_e$ are given design targets. The gap thus allows rate calculation, given the gap and SNR. Thus, a designer need not "redesign" a code for different data rates; instead the designer knows the same code will work for whatever data rate that the SNR and selected code/$\bar{P}_e$, equivalently Γ, support. The gap does not work for low data rates ($\bar{b} < 0.5$).

Smaller gap means better code with $\Gamma = 1$ (0 dB) as best, but only approachable bound with infinite implementation delay. For simple "uncoded" QAM/PAM (students EE379A will recognize), the gap is typically 8.8 dB (so that means $10^{.88}$ if inserted in the formula) if the usual error-probability target is 10^{-6}. The gap is 9.5 dB for 10^{-7} and roughly 5.8 dB for 10^{-3}. Some popular codes have gaps for $\bar{P}_e = 10^{-6}$:

1. Trellis codes, particularly 4D code, has coding gain $\gamma_c = 4.5$ dB, and reduce the gap to $\Gamma - \gamma_c = 8.8 - 4.5 = 4.3$ dB
2. Turbo Codes, two interleaved 8-state binary codes with BICM, $\gamma_c = 5.5$ dB, so then $\Gamma = 3.3$ dB.
3. LDPC codes with block lengths of roughly 1000 or more and BICM, $\gamma_c \to 7$ dB, so $\Gamma = 1.8$ dB.
4. "Polar Codes" do not have a constant gap and find successful use only for low SNR channels where $\bar{b} < 1$ and unfortunately do depend heavily on the selected $\bar{b}$.

When $\bar{b}$ is large, approach to capacity will also require a "shaping" code, which as in Chapters 1 and 2 can add another up to 1.5 dB to coding gain (or reduce gap by 1.5 dB). This shaping effect is often not used in practice, but would be necessary to approach the ideal of a 0 dB gap code.

Examples:

1. A channel has SNR = 13.5 dB and uses uncoded PAM/QAM and $\bar{P}_e = 10^{-6}$.

   $$\bar{b} = 0.5 \cdot \log_2(1 + 10^{1.35 - .88}) = 1 \text{ bit/dimension.}$$

   If this channel uses a good code with $\Gamma = 7$ dB, then the data rate should increase

   $$\bar{b} = 0.5 \cdot \log_2(1 + 10^{1.35 - (.88 - .7)}) = \ 2\text{bits/dimension.}$$

With Trellis code and 4.5 dB

$$\bar{b} = 0.5 \cdot \log_2(1 + 10^{1.35-.43}) = 1.6 \text{ bit/dimension.}$$

The best performance would be

$$\bar{b} = \bar{\mathcal{C}} = 0.5 \cdot \log_2(1 + 10^{1.35-0}) = 2.3 \text{ bits/dimension.}$$

2. Repeating the previous example with SNR = 30 dB yields

```
>> GAP=[8.8 1.8 4.3 0];
>> SNR=30;
>> b=0.5*log2(1+(10.^(0.1*(SNR - GAP))) =

     3.5267    4.6850    4.2706    4.9836  bits/dimension
```

The gap formula can be solved for required gap in terms of a desired bit rate $\bar{b}$

$$\Gamma = \frac{SNR}{2^{2\bar{b}} - 1} \quad .$$

or necessary SNR to achieve the desired bit rate

$$SNR = \Gamma \cdot (2^{2\bar{b}} - 1) \quad .$$

resume So, thus the SNR required for 5 bits/dimension with uncoded 1024 QAM at $\Gamma = 5.8$ dB for $\bar{P}_e = 10^{-3}$ is (note here using dB directly since $2^b \to b{\cdot}3$ dB)

$$SNR = 10^{.58} \cdot (2^{2 \cdot 5} - 1) = 5.8 + 10 \cdot 3 = 35.9 \text{ dB} \ .$$

resume Or for an AWGN channel with 20 dB SNR and a desired 3 bits/dimension, the code, with whatever $\bar{P}_e$ is the target must have a gap

$$\Gamma = 20 - 6 \cdot 3 = 2 \text{ dB} \quad .$$

When the required Gamma is negative, the desired data rate is not achievable with any code at any (acceptable) error probability.

The margin γ_m is similar to the gap, but essentially reduces the gap by an amount that guards against unforeseen noise. The margin holds for the same $\bar{P}_e$ as the gap.

$$\bar{b} = \frac{1}{2} \log_2 \left(1 + \frac{SNR}{\Gamma \cdot \gamma_m} \right) \quad , \quad \bar{b} \geq 0.5 \quad .$$

$(\Gamma \cdot \gamma_m)$ could be viewed as one parameter. Negative margin means that the chosen code for the gap cannot achieve the data rate. We could repeat examples in items 1 - 2 with any margin desired. For instance, inserting $\gamma_m = 3$ dB will roughly reduce all the bits/dimension by 1/2. Often the margin is computed by

$$\gamma_m = \frac{SNR/\Gamma}{2^{2\bar{b}} - 1} \quad .$$

or the desired SNR with certain safety margin computed as

$$SNR = \Gamma \cdot \gamma_m \cdot (2^{2\bar{b}} - 1) \quad .$$

Thus the margin for a 24 dB SNR AWGN with $\Gamma = 3$ dB at $\bar{P}_e = 10^{-7}$ that attempts 3 bits/dimension is

$$\gamma_m = 24 - 3 - 6 \cdot 3 = 3 \text{ dB} .$$

Equivalently for this rate and a margin of 2 dB, the necessary channel SNR is

$$SNR = 3 + 2 + 6 \cdot 3 = 23 \text{ dB} .$$

Now to the problem:

a. The gap depends on P_e, and is constant over what other design choice?

b. What caused the gap to reduce from EE379A's various Chapter 8 studies? (This is a simple 1-2 word answer.)

c. Work with the formula:
$$\gamma_m = \frac{SNR/\Gamma}{2^b - 1} .$$

d. Use again the same formula as last part.

DMT Program Help (4.18) First some general DMT program-use summary: The DMT programs basically implement water-filling for matrix AWGN channels describing intersymbol interference. They implement water-filling for the energy distribution. While SVD or DFT's would easily identify channel gains with any matrix, these programs take into account the cyclic-prefix/guard period automatically. The channel is thus viewed as a packet of transmitted samples through (convolved with) a discrete-time channel response, with the receiver discarding the corresponding guard-period samples.

The first input is the channel, specified as a discrete-time response in a row vector. The channel starts with it's left most sample and completes with it's rightmost sample. ν is the length of the channel minus 1. The AWGN noise parameter σ^2 is the second input, while the third input is the energy/dimension transmitted. If both these inputs have the same dimensionality (but not 1, for instance both are 2 dimensional), the program still works. The 4th input is the FFT size, or equivalently the symbol length in sampling periods (minus ν). The last input is the gap in dB. With these 5 inputs, the program will product the overall (unbiased) SNR for the set of channels and the data rate in bits/dimension.

There are actually 5 outputs total, which provide additional information: The first output are the channel gains, so basically $|H_n|^2/\sigma^2$. The 2nd output is the water-fill energy vector (which necessarily sums to product of the input energy and the number of dimensions N) $\mathcal{E}_n$; while the 3rd output is the corresponding water-fill bit distribution $\bar{b}_n$ bits/dimension. The 3rd output is the number of energized dimensions N^*. The last two outputs are the most useful for overall analysis in that they are the total bits/symbol and the SNR. The factor 1/2 is used in the program so the overall bits are divided by $\overline{N} + \nu$. For a real channel, that is just $N + \nu$, but for a complex channel, the N input is really assumed to be $\overline{N}$, so the number of complex dimensions. Because the program only uses the ratio of the input energy to noise, as long as those two have the same number of dimensions, it works. In the complex case, the bits and totals are per-complex-dimension.

An example might be the real-baseband channel $H(D) = 1 + .8D + .4D^2$ with noise .2 and energy/dimension 1:

```
>> [gn,en,bn_bar,Nstar,b_bar,SNRdmt]=DMTra([1 .8 .4],.2,1,8,0)

gn= 24.2000 16.9196  5.0000  1.0804  1.8000  1.0804  5.0000 16.9196
en=  1.3295  1.3117  1.1708  0.4452  0.8152  0.4452  1.1708  1.3117
bn_bar= 2.5260 2.2678 1.3885 0.2833  0.6515  0.2833  1.3885  2.2678
Nstar=  8
b_bar=  1.1057
SNRdmt=  5.6002 dB
```

Even with zero-dB gap, this channel does not have a very high data rate and uses all dimensions. Notice the channel appears low pass because bits and energies near Nyquist (the 5th dimension here, DC is the first on left) are smaller. Changing the gap to uncoded causes a large loss

```
>> [gn,en,bn_bar,Nstar,b_bar,SNRdmt]=DMTra([1 .8 .4],.2,1,8,8.8)
gn= 24.2000 16.9196  5.0000  1.0804  1.8000  1.0804  5.0000 16.9196
en=  2.1354  2.0005  0.9317       0       0       0  0.9317  2.0005
bn_bar= 1.4829 1.2247 0.3454      0       0       0  0.3454  1.2247
Nstar=  5
b_bar=  0.4623
SNRdmt=  8.3337 dB
```

The SNR increased (but recall this gets divided by the gap so the argument of the log is smaller). Only 5 dimensions are used by water-filling in this case. Bigger gap always leads to reduction (or staying the same) in number of used dimensions. Suppose the transmit energy doubles:

```
>> [gn,en,bn_bar,Nstar,b_bar,SNRdmt]=DMTra([1 .8 .4],.2,2,8,8.8)
gn= 24.2000 16.9196  5.0000  1.0804  1.8000  1.0804  5.0000 16.9196
en=  3.7354  3.6005  2.5317       0       0       0  2.5317  3.6005
bn_bar= 1.8456 1.5874 0.7081      0       0       0  0.7081  1.5874
Nstar= 5
b_bar=  0.6437
SNRdmt= 10.3859
```

Well, still not so great. How about more dimensions (use the semicolon or it will print a lot of numbers)

```
>> [gn,en,bn_bar,Nstar,b_bar,SNRdmt]=DMTra([1 .8 .4],.2,2,128,8.8);
>> Nstar =    81
>> b_bar =    0.7893
>> SNRdmt =   11.7818
```

Well, reduce the gap to 3 dB?

```
>> [gn,en,bn_bar,Nstar,b_bar,SNRdmt]=DMTra([1 .8 .4],.2,2,128,3);
>> Nstar =    128
>> b_bar =    1.3594
>> SNRdmt =   10.4685
```

Ok, we'll live with this design at 1.3 bits/dimension (or almost 11 bits/symbol). Best possible for this channel is about 1.8 bits/dimension or 14 bits/symbol, but hard to realize.

```
>> [gn,en,bn_bar,Nstar,b_bar,SNRdmt]=DMTra([1 .8 .4],.2,2,1024,0);
>> Nstar=    1024
>> b_bar=    1.7749
>> 8*b_bar=  14.1993
```

The use of this program with a complex channel is illustrated in Lecture. Dimensions then are "complex" and the symmetry in the bit/energy distributions about Nyquist (or equivalently about DC if one cyclically shifts the distributions) can be lost if the channel is not symmetric about its chosen carrier frequency. The input channel response is the complex baseband-equivalent channel (see EE379 or Chapter 1).

The student is encouraged to experiment with the MA version of this program, and an example is provided in class. The DMTLCRA program accepts the same inputs and outputs and produces bit distributions that are integers on all dimensions.

Now to PS1.3:

a. This part basically has you repeat and understand the class' and text's $1+.9D^{-1}$ channel with a different gap and $.9 \to .5$, so run the DMTra.m program with these input changes.

b. Form the proper Toeplitz convolution matrix to describe the channel and do svd in matlab on it. The singular values are the channel gains (square roots). They are not repeated, while the DMT values. How do magnitudes compare?

c. The idea here is to see that the singular values start to converge on to the FFT values for large N for the stationary Toeplitz channel.

Vector Coding/MIMO Help First some general Vector Coding material summary: Vector Coding uses SVD, providing both the transmitter (M matrix) and the receiver (F^*) for a matrix AWGN, which becomes a set of independent scalar AWGN channels with amplitude multipliers λ_ℓ. The data rate (or bits/symbol b_ℓ) is then the sum of the scalar channels's data rates, $\sum_\ell b_\ell = \log_2(1+SNR_\ell)$. (Put a 1/2 in front of log if a purely real baseband channel.)

The transmitter M does not change the energy sum over the scalar channels, nor does the F^* receiver change the noise's "white-ness." Prior to vector coding (which single-user MIMO systems use), the other antennas' (dimensions') signals were viewed as "part of the noise." Thus, it is instructive to appreciate the large difference between vector coding and those earlier systems. So perhaps there is a 3×3 matrix

$$H = \begin{bmatrix} H_{3,3} & H_{3,2} & H_{3,1} \\ H_{2,3} & H_{2,2} & H_{2,1} \\ H_{1,3} & H_{1,2} & H_{1,1} \end{bmatrix} =$$

```
>> H =

    10    -8     6
     5     9    -7
     4    -6     8
```

Maybe with $\bar{\mathcal{E}}_i = 1$ and independent on each input (so this means $\mathcal{E}_i = 2$ if the channel is complex baseband). Further suppose $\sigma^2 = 1$ on each real dimension for white noise.

To compute the data rate with the other signals as noise, say for output 2 computes

$$b_2 = \log_2 \left(1 + \frac{|H_{2,2}|^2 \cdot \bar{\mathcal{E}}_2}{\sigma^2 + |H_{2,1}|^2 \cdot \bar{\mathcal{E}}_1 + |H_{2,3}|^2 \cdot \bar{\mathcal{E}}_3} \right) =$$

```
>> log2(1 + H(2,2)^2 / (1+H(2,1)^2+H(2,3)^2))  =  1.0566
```

while the signals are well above the noise, the crosstalk dominates and lowers this number, which is poor if computed for the other two dimensions, each treating 2 different dimensions as noise, .9928 for the top dimension and 1.1424 for the bottom dimension. (Note the index reversal from text and matlab.). A linear MMSE estimator will help, but won't remove all the crosstalk. That is subject of another section in PS2.

If an SVD is instead done, then

```
>> [F,Lambda,M]=svd(H)


F =
   -0.6818   -0.5508   -0.4815
    0.4685   -0.8342    0.2909
   -0.5619   -0.0273    0.8268
Lambda =
   18.8697         0         0
         0   10.4786         0
         0         0    2.2657
M =
   -0.3563   -0.9341   -0.0235
    0.6912   -0.2804    0.6661
   -0.6288    0.2211    0.7455
```

and the 2nd dimension's data rate is instead, with one unit of energy on each dimension

$$b_2 = \log_2\left(1 + \frac{|\Lambda_{2,2}|^2 \cdot \bar{\mathcal{E}}_2}{\sigma^2}\right) = 6.8 \quad .$$

A big improvement (and similarly so on the other two channels with top dimension as 8.48 and bottom dimension as 2.62. The sum is 17.8. Water-filling will make this yet better sum of rates. The K for rate-adaptive case with energy 6 is (all dimensions are used for this water-fall)

```
>> K=(1/3)*(6+sum(diag(inv(Lambda)^2))) =    2.0689
```

with then energies and consequent bits

```
>> E3=K-1/((Lambda(1,1))^2) =    2.0661
>> E2=K-1/((Lambda(2,2))^2) =    2.0598
>> E1=K-1/((Lambda(3,3))^2) =    1.8741
>> sum([E1,E2,E3]) =     6 (good way to check)

>> b3=log2(1+E3*Lambda(1,1)^2) =    9.5249
>> b2=log2(1+E2*Lambda(2,2)^2) =    7.8276
>> b1=log2(1+E1*Lambda(3,3)^2) =    3.4088
>> sum([b1,b2,b3]) =   20.7613 > 17.8
```

The gain in overall sum is the water-filling on top of the vector coding (vector coding can use any energy distribution, not just water-filling).

For the problem PS1.3:

1. The solution can use the DMTra.m program. This follows the $1 + .9D^{-1}$ inverse in class and text, just .5 going where .9 was in terms of affected program inputs, and the gap is different.
2. The target error probability $\bar{P}_e$.

Thus, specifically, it does not depend on the data rate ($\bar{b}$) directly and is **constant** if the code and $\bar{P}_e$ are given design targets. The gap thus allows rate calculation, given the gap and SNR. Thus, a designer need not "redesign" a code for different data rates; instead the designer knows the same code will work for whatever data rate that the SNR and selected code/$\bar{P}_e$, equivalently Γ, support. The gap does not work for low data rates ($\bar{b} < 0.5$).

Smaller gap means better code with $\Gamma = 1$ (0 dB) as best, but only approachable bound with infinite implementation delay. For simple "uncoded" QAM/PAM (students EE379A will recognize), the gap is typically 8.8 dB (so that means $10^{.88}$ if inserted in the formula) if the usual error-probability target is 10^{-6}. The gap is 9.5 dB for 10^{-7} and roughly 5.8 dB for 10^{-3}. Some popular codes have gaps for $\bar{P}_e = 10^{-6}$:

1. Trellis codes, particularly 4D code, has coding gain $\gamma_c = 4.5$ dB, and reduce the gap to $\Gamma - \gamma_c = 8.8 - 4.5 = 4.3$ dB

2. Turbo Codes, two interleaved 8-state binary codes with BICM, $\gamma_c = 5.5$ dB, so then $\Gamma = 3.3$ dB.
3. LDPC codes with block lengths of roughly 1000 or more and BICM, $\gamma_c \to 7$ dB, so $\Gamma = 1.8$ dB.
4. "Polar Codes" do not have a constant gap and find successful use only for low SNR channels where $\bar{b} < 1$ and unfortunately do depend heavily on the selected $\bar{b}$.

When $\bar{b}$ is large, approach to capacity will also require a "shaping" code, which as in Chapters 1 and 2 can add another up to 1.5 dB to coding gain (or reduce gap by 1.5 dB). This shaping effect is often not used in practice, but would be necessary to approach the ideal of a 0 dB gap code.

Examples:

1. A channel has SNR = 13.5 dB and uses uncoded PAM/QAM and $\bar{P}_e = 10^{-6}$.

   $$\bar{b} = 0.5 \cdot \log_2(1 + 10^{1.35-.88}) = 1 \text{ bit/dimension.}$$

   If this channel uses a good code with $\Gamma = 7$ dB, then the data rate should increase

   $$\bar{b} = 0.5 \cdot \log_2(1 + 10^{1.35-(.88-.7)}) = \text{ 2bits/dimension.}$$

   With Trellis code and 4.5 dB

   $$\bar{b} = 0.5 \cdot \log_2(1 + 10^{1.35-.43}) = 1.6 \text{ bit/dimension.}$$

   The best performance would be

   $$\bar{b} = \bar{\mathcal{C}} = 0.5 \cdot \log_2(1 + 10^{1.35-0}) = 2.3 \text{ bits/dimension.}$$

2. Repeating the previous example with SNR = 30 dB yields

```
>> GAP=[8.8 1.8 4.3 0];
>> SNR=30;
>> b=0.5*log2(1+(10.^(0.1*(SNR - GAP))) =

    3.5267    4.6850    4.2706    4.9836  bits/dimension
```

The gap formula can be solved for required gap in terms of a desired bit rate $\bar{b}$

$$\Gamma = \frac{SNR}{2^{2\bar{b}} - 1} \quad .$$

or necessary SNR to achieve the desired bit rate

$$SNR = \Gamma \cdot (2^{2\bar{b}} - 1) \quad .$$

resume So, thus the SNR required for 5 bits/dimension with uncoded 1024 QAM at $\Gamma = 5.8$ dB for $\bar{P}_e = 10^{-3}$ is (note here using dB directly since $2^b \to b{\cdot}3$ dB)

$$SNR = 10^{.58} \cdot (2^{2\cdot 5} - 1) = 5.8 + 10 \cdot 3 = 35.9 \text{ dB} .$$

resume Or for an AWGN channel with 20 dB SNR and a desired 3 bits/dimension, the code, with whatever $\bar{P}_e$ is the target must have a gap

$$\Gamma = 20 - 6 \cdot 3 = 2 \text{ dB} \ .$$

When the required Gamma is negative, the desired data rate is not achievable with any code at any (acceptable) error probability.

The margin γ_m is similar to the gap, but essentially reduces the gap by an amount that guards against unforeseen noise. The margin holds for the same $\bar{P}_e$ as the gap.

$$\bar{b} = \frac{1}{2} \log_2 \left(1 + \frac{SNR}{\Gamma \cdot \gamma_m} \right) \ \ , \ \ \bar{b} \geq 0.5 \ .$$

$(\Gamma \cdot \gamma_m)$ could be viewed as one parameter. Negative margin means that the chosen code for the gap cannot achieve the data rate. We could repeat examples in items 1 - 2 with any margin desired. For instance, inserting $\gamma_m = 3$ dB will roughly reduce all the bits/dimension by $1/2$. Often the margin is computed by

$$\gamma_m = \frac{SNR/\Gamma}{2^{2\bar{b}} - 1} \ .$$

or the desired SNR with certain safety margin computed as

$$SNR = \Gamma \cdot \gamma_m \cdot (2^{2\bar{b}} - 1) \ .$$

Thus the margin for a 24 dB SNR AWGN with $\Gamma = 3$ dB at $\bar{P}_e = 10^{-7}$ that attempts 3 bits/dimension is

$$\gamma_m = 24 - 3 - 6 \cdot 3 = 3 \text{ dB} \ .$$

Equivalently for this rate and a margin of 2 dB, the necessary channel SNR is

$$SNR = 3 + 2 + 6 \cdot 3 = 23 \text{ dB} \ .$$

Now to PS1.4

a. The gap depends on P_e, and is constant over what other design choice?

b. What caused the gap to reduce from EE379A's various Chapter 8 studies? (This is a simple 1-2 word answer.)

c. Work with the formula:

$$\gamma_m = \frac{SNR/\Gamma}{2^b - 1}.$$

d. Use again the same formula as last part.

Partitioning For PS1.5

a. This channel is highly symmetric so all the users should look the same. This simplifies the amount of computation to respond.

b. You may want to review what a circulant matrix is, even though this is not a temporal convolution situation.

c. This is pretty easy recognizing that each user (since wireless) is using a QAM signal necessarily.

d. The spatial matrix can sometimes be an FFT matrix or have that character, even if spatial patterns may not always conform to this pattern. The special structure makes this problem similar to a DMT problem, even though the crosstalk is spatial.

e. You can analyze 3x3 water filling pretty easily by hand, but you could use the programs also - however, there is NO cyclic prefix penalty here in space time, so you would need to adjust those programs accordingly.

f. The answer to this part is a lot larger data rate than in Part a, demonstrating basic MIMO gain again.