**ECE 233 Project Report**

**Hybrid Digital and Analog Beamforming Design**

**for Large-Scale Antenna Arrays**

**Ruoye Wang | 605625594**

**Yida Chen | 005852117**

**Abstract**

The massive multiple-input multiple-output (MIMO) concept has been widely studied and developed. It enables large-scale spatial multiplexing and highly directional beamforming. This project implements a two-stage hybrid beamforming architecture in MATLAB, utilizing the system model and algorithms described in the paper “Hybrid Digital and Analog Beamforming Design for Large-Scale Antenna Arrays” by Foad Sohrabi and Wei Yu.

In this project, the overall beamformer is composed of a digital beamformer, followed by an RF beamformer that uses analog phase shifters for implementation. The finite resolution effect of analog phase shifters is studied. Two scenarios are considered. Firstly, the number of RF chains at the base station (BS) and user end is equal to the number of data streams (). The second scenario is where the number of NF chains is larger than the number of data streams but smaller than twice the total number of data streams (). This report presents the simulation outcomes obtained using MATLAB. Our findings closely correspond to the results documented in the reference paper.

**System Model**

This project implements a model proposed by the reference paper. This narrowband downlink single-cell multi-user MIMO system model has a two-stage hybrid digital and analog beamforming architecture at the base station (BS) and the user terminals. A picture containing diagram, plan, text, technical drawing

Description automatically generated

Figure 1: Block diagram of the model proposed by the paper

As the figure above shows, the BS has antennas and RF chains and serves users. Each user is equipped with antennas and RF chains and requires data streams. The number of data streams required by each user; the total number of data streams .

This project assumes single-user scenario, i.e., . To simplify the notation while preserving the generality, it is assumed that . The project first implements the hybrid beamformer design for the case where to show that, according to Proposition 1, a fully digital beamformer architecture can be realized by a hybrid structure with at least RF chains using a proposed heuristic algorithm. Then the same algorithm is implemented for the case where .

The symbols represent the digital precoder at the BS (size ), the RF precoder at the BS (size ), the digital combiner at the user end (size ), and the RF combiner at the user end (size ). (size ) is the matrix of the complex channel gains from the transmit antennas of the BS to the user (note that since , all can be represented by a single ; the same can be applied to other user-specific quantities in the paper).

Then, the transmitted signal can be represented by:

A picture containing font, text, white, typography

Description automatically generated

Figure 2: Transmitted signal

Where is the vector of data symbols. And the final processed signals are obtained as:

A picture containing font, text, handwriting, typography

Description automatically generated

Figure 3: Final processed signal

Where represents the additive white Gaussian noise.

**Main Part**

This paper mainly focuses on maximizing the overall spectral efficiency with total transmit power constrained and fully known. This requires us to find the optimal solution for precoders at the transmitter end and the combiners at the receiver end, which can be represented by this formula:

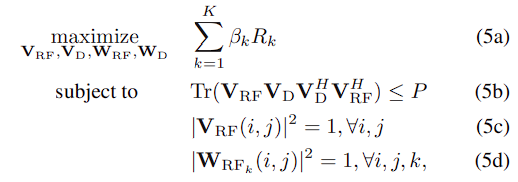


Figure 4: the formula representing the main problem (K=1, disregarding k)

This formula is calculated under the aforementioned cases and . It can be simplified for precoder design:

A picture containing text, font, line, white

Description automatically generated

Figure 5: Precoder design formula

**scenario**

When , can be calculated by:

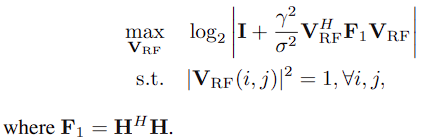


Figure 6: RF precoder design formula

A picture containing text, screenshot, font, number

Description automatically generatedWhich is a simplification of Figure. 5 assuming . It is summarized in Algorithm 1:

Figure 7: Algorithm 1 to calculate RF precoder, with infinite resolution phase shifter

The formula to calculate is under the assumption of infinite phase shifters. Since accurate phase shifters can be expensive to implement in real world, they are commonly replaced by cost effective low resolution phase shifters. To simulate such phase shifters, the design of is quantized:

A picture containing text, font, white, diagram

Description automatically generated

Figure 8: RF precoder design, with 1-bit resolution phase shifter

Then can be calculated by solving:

A picture containing text, font, handwriting, line

Description automatically generated

Figure 9: Digital precoder design formula

Where and .

We use a water-filling solution (Reference: [Waterfilling](https://zhuanlan.zhihu.com/p/502453127)) to allocate power to each channel so that the overall channel capacity is maximized. The sum of power satisfies:

**A picture containing sketch, diagram, white, font

Description automatically generated**

Figure 10: Constraint on power allocation

Where the power of the first most efficient channels sum up to use to the largest extend. Since power cannot be negative:

A picture containing font, diagram, white, line

Description automatically generated

Figure 11: Formula to calculate allocated power

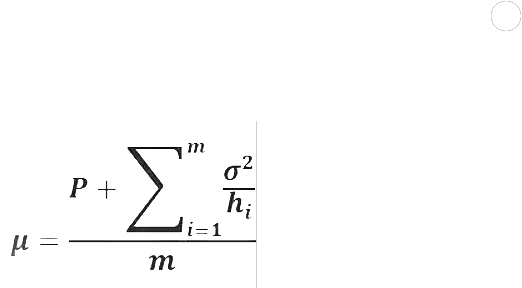
Combining the above two formulas, the water level chosen to satisfy the power sum constraints with equality can be represented by:

Figure 12: Formula to calculate water level

is obtained by iterating through the number of total channels . In each iteration, is calculated using the above formula, and compared with , where represents noise and is the square of right singular value of the SVD result of , since in Figure 9, the matrices are applied twice on both left and right. Similarly, the diagonal elements of are all square roots of . (Reference: Ruifu Donar Li)

The final value of is chosen so that satisfies the power constraint, as the following figure shows:

A picture containing diagram, line, technical drawing, rectangle

Description automatically generated

Figure 13: Water-filling solution

Then the matrix of digital precoder can be calculated using the formula proposed by the paper:

Where is the set of right singular vectors corresponding to the largest singular values of and is the diagonal matrix of allocated powers to each stream.

With designed, the formula in Figure 3 can be simplified to:

A picture containing text, font, line, white

Description automatically generated

Figure 14: RF combiner design formula

Due to its similarity to Figure 6, it can also be solved with Algorithm 1, with replaced by and since is large. The quantized is also calculated with the same method as .

Finally, the optimal digital combiner can be obtained from its MMSE solution:

A picture containing text, font, white, typography

Description automatically generated

Figure 15: Digital combiner design formula

After have been obtained, the performance of the model can be evaluated by its spectral efficiency:

A picture containing text, font, handwriting, white

Description automatically generatedThe above process is repeated for a range of SNR, and for each SNR value, spectral efficiency is averaged over 100 Monte Carlo trials.

Figure 16: Formula of spectral efficiency

**scenario**

As stated in the paper, this scenario can still be implemented using the aforementioned process. It is implemented for .

**Results and Discussion**

1. A picture containing line, diagram, plot, parallel

   Description automatically generatedPlot the spectral efficiency vs. SNR in the range −10 dB to 6 dB, assuming a 64 × 16 MIMO system and .

Figure 17: Spectral efficiency vs. SNR for result 1

The resulting figure is similar to Fig. 2 in the paper, with ~1.5dB difference on SNR values.

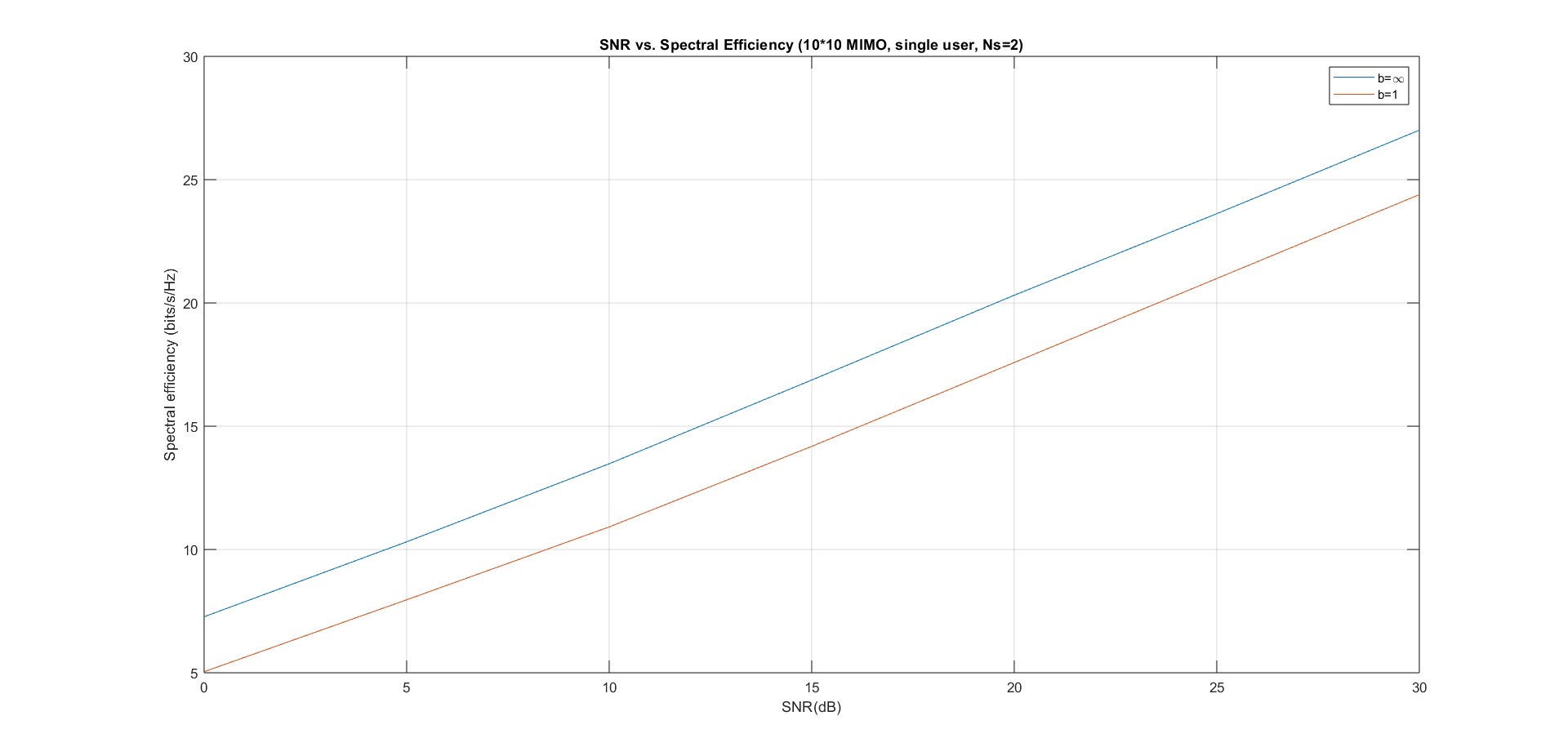
1. Plot the spectral efficiency vs. SNR in the range 0 dB to 30 dB, assuming a 10 × 10 MIMO system, , and phase shifters with 1-bit and infinite resolutions.

Figure 18: Spectral efficiency (infinite resolution and 1-bit phase shifters) vs. SNR for result 2

In paper Fig. 3, the spectral efficiency of 1-bit phase shifter is higher than that of infinite resolution phase shifter. However, this resulting figure shows the opposite. One possible reason is that, the paper’s beamformer is designed under the assumption of infinite resolution phase shifters and each entry of the RF beamformer is quantized to the nearest point of the set of possible phases. However, in this project, the beamformers for infinite resolution phase shifter is not quantized.

1. Plot the spectral efficiency vs. SNR in the range −10 dB to 6 dB, assuming a 64 × 16 MIMO system, , and phase shifters with 1-bit and infinite resolutions.

A picture containing line, plot, diagram, text

Description automatically generatedA picture containing line, plot, diagram

Description automatically generated

Figure 20: Spectral efficiency (1-bit phase shifter) vs. SNR, 100 Monte Carlo trials

Figure 19: Spectral efficiency (Infinite resolution phase shifter) vs. SNR, 100 Monte Carlo trials

The curves for 1-bit phase shifter are ~1.5dB lower than paper Fig. 4 which is considered acceptable error. Same as Fig. 4, the curves for infinite phase shifters (plotted separately for clarity) are significantly higher than that of 1-bit phase shifters, and the spectral efficiency increases as the number of RF chains increase, so that 1-bit phase shifter with more RF chains has a performance close to infinite phase shifters. This is in accordance with the paper’s conclusion that the number of RF chains can be used to trade off the accuracy of phase shifters in hybrid beamforming design.

A picture containing line, plot, diagram, text

Description automatically generatedThe above curves are rather unsmooth because the number of Monte Carlo trials are small. After the number of trials is increased to 1000, the figures become:

Figure 21: Spectral efficiency (infinite resolution phase shifter) vs. SNR, 1000 Monte Carlo trials

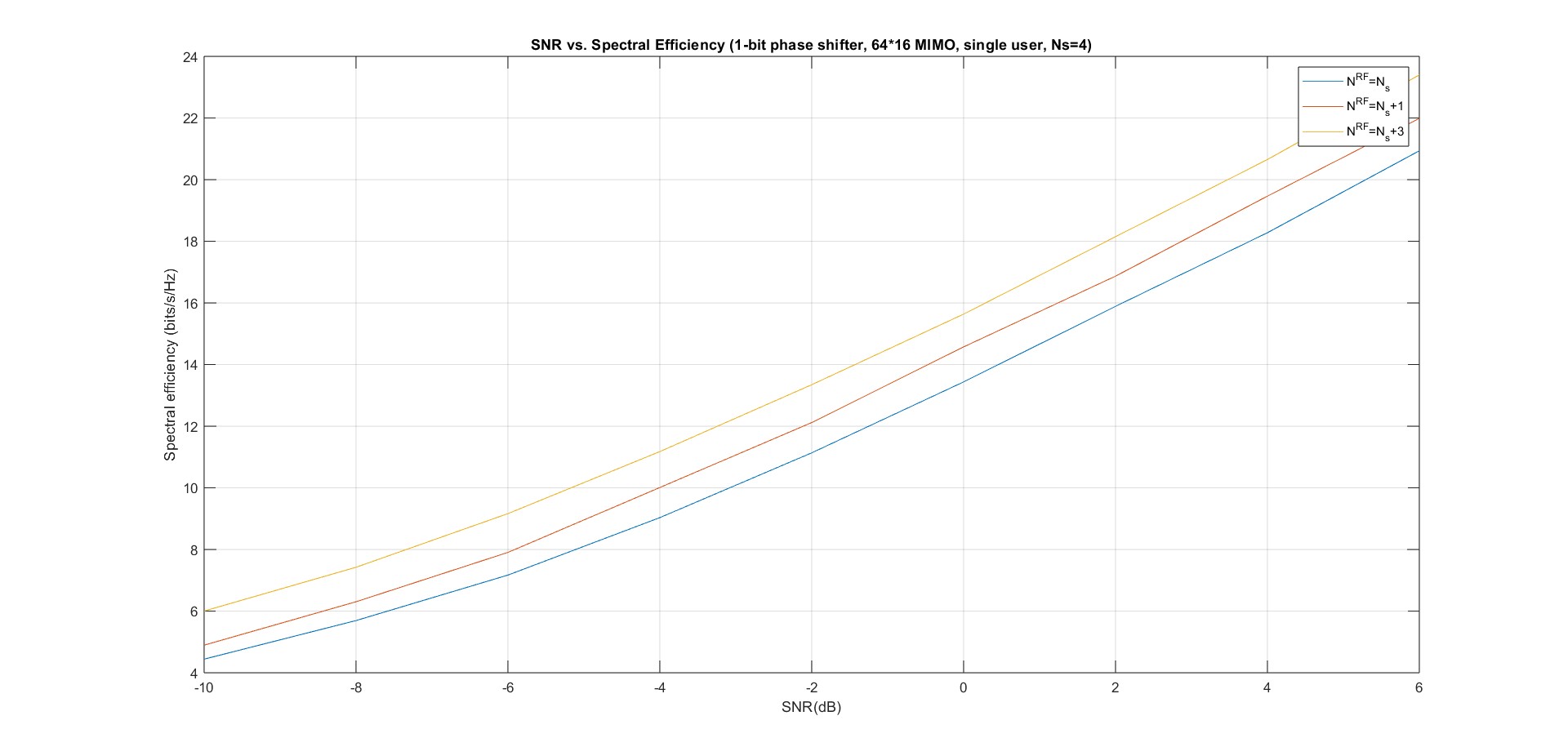


Figure 22: Spectral efficiency (1-bit phase shifter) vs. SNR, 1000 Monte Carlo trials

Which more clearly shows the paper’s discoveries.

**Conclusion**

This project implements the hybrid beamformer architecture proposed by the reference paper for wireless systems with large-scale antenna arrays. It focuses on single-user scenario and explores the performance of hybrid beamformers. It first implements a hybrid beamformer design in a 64\*16 MIMO system assuming the number of RF chains is equal to the number of data streams, then modifies the algorithm to compare the difference between infinite resolution and 1-bit phase shifters. The process is repeated for the scenario that the number of RF chains is less than twice but larger than once of the number of data streams. The simulation results are in accordance with the paper’s simulation figures with only minor difference on the range of spectral efficiency. Result 2 shows the performance of infinite resolution phase shifter is better than that of 1-bit phase shifter, which is opposite from paper Fig. 3, because the project does not quantize the infinite resolution phase shifter as in the paper. Result 3 shows that increasing the number of RF chains in hybrid beamformer architectures trades off its insufficiency in accuracy compared to fully digital ones and thus allows it to serve as a sub-optimal solution for massive MIMO systems.

**Appendix: code**

Note: to produce 3 results in the project specification separately, please refer to the instructions in the comment and change the parameter set accordingly.

clear;

%% Project 1

%% NOTE Change parameters according to the results in project spec!!!

N=10; % Number of antennas of the base station | 64 | 10

M=10; % Number of antennas of each user | 16 | 10

K=1; % Number of users

d=2; % Number data streams required by each user | 6 | 2 | 4

Ns=K\*d; % Number of total data streams

L=15; % Number of paths (paper suggests 15)

SNR=0:5:30; % -10:2:6 | 0:5:30

tolerance=1e-6;

num\_mc=100; % Number of Monte Carlo trials | 100 | 1000

phase\_list=[1 exp(1j\*pi)];

P=1;

% Complex channel

H=zeros(N,M);

% Uncomment for results 1, 2 ↓

N\_RF\_list=Ns; % Sec. IV assumes N\_RF=Nt\_RF=Nr\_RF preserves generality

R\_list=zeros(1,length(SNR));

R\_quant\_list=zeros(1,length(SNR));

% Uncomment for results 1, 2 ↑

% Uncomment for result 3 ↓

%N\_RF\_list=[Ns, Ns+1, Ns+3];

%R\_list=zeros(3,length(SNR));

%R\_quant\_list=zeros(3,length(SNR));

% Uncomment for result 3 ↑

for i\_nrf=1:length(N\_RF\_list)

N\_RF=N\_RF\_list(i\_nrf);

Nt\_RF=N\_RF; % Number of transmit RF chains

Nr\_RF=N\_RF; % Number of receive RF chains

% NOTE Nt\_RF=Nr\_RF=N\_RF, the total number of RF chains is 2\*N\_RF which

% satisfies "if the number of RF chains is twice the total number of data streams, the

% hybrid beamforming structure can realize any fully digital beamformer exactly"

% Initialize Tx precoder and combiner

V\_D=zeros(Nt\_RF,Ns); % (14)

V\_RF=zeros(N,Nt\_RF); % (11)

Vt=V\_RF\*V\_D;

for i\_snr=1:length(SNR)

%i\_snr

SNR\_lin=10^(SNR(i\_snr)/10);

sigma=sqrt(P/SNR\_lin);

R=0;

R\_quant=0;

for i\_mc=1:num\_mc

%i\_mc

H=zeros(M,N);

for l=1:L

alpha=sqrt(1/2)\*(randn(1,1)+1j\*randn(1,1));

phi\_r=2\*pi\*randn(1,1);

phi\_t=2\*pi\*randn(1,1);

% TODO rand

a\_r=transpose(exp(1j\*pi\*(0:M-1)\*sin(phi\_r))/sqrt(M));

a\_t=transpose(exp(1j\*pi\*(0:N-1)\*sin(phi\_t))/sqrt(N));

H=H+alpha\*a\_r\*a\_t';

end

H=sqrt(N\*M/L)\*H;

F1=H'\*H;

%% Part 1: IV.B RF Precoder Design for N\_RF=Ns

% Necessary parameters

V\_RF=ones(N,N\_RF);

V\_RF\_quant=ones(N,N\_RF);

% Translating algorithm 1 to design V\_RF

not\_converge\_inf=1;

not\_converge\_1bit=1;

% Convergence loop for infinite phase shifter

while not\_converge\_inf

for j=1:N\_RF

V\_RF\_noj=V\_RF;

V\_RF\_noj(:,j)=[];

C\_j=eye(Nr\_RF-1,Nr\_RF-1)+(SNR\_lin/N/N\_RF)\*V\_RF\_noj'\*F1\*V\_RF\_noj;

G\_j=(SNR\_lin/N/N\_RF)\*F1-(SNR\_lin/N/N\_RF)^2\*F1\*V\_RF\_noj\*pinv(C\_j)\*V\_RF\_noj'\*F1;

for i=1:N

eta\_ij=0;

for l=1:N

if l~=i

eta\_ij=eta\_ij+G\_j(i,l)\*V\_RF(l,j);

end

end

% Infinite phase shifter

if eta\_ij==0

V\_RF(i,j)=1;

else

V\_RF(i,j)=eta\_ij/abs(eta\_ij);

end

end

end

% Check convergence

if abs(abs(V\_RF).^2-ones(N,Nr\_RF)) < tolerance\*ones(N,Nr\_RF)

not\_converge\_inf=0;

end

end

% TODO for each entry of precoder, quantize it to the nearest

% in F = {1, w, ...}

% Convergence loop for 1-bit phase shifter

while not\_converge\_1bit

for j=1:N\_RF

V\_RF\_noj=V\_RF;

V\_RF\_noj(:,j)=[];

C\_j=eye(Nr\_RF-1,Nr\_RF-1)+(SNR\_lin/N/N\_RF)\*V\_RF\_noj'\*F1\*V\_RF\_noj;

G\_j=(SNR\_lin/N/N\_RF)\*F1-(SNR\_lin/N/N\_RF)^2\*F1\*V\_RF\_noj\*pinv(C\_j)\*V\_RF\_noj'\*F1;

for i=1:N

eta\_ij\_quant=0;

for l=1:N

if l~=i

eta\_ij\_quant=eta\_ij\_quant+G\_j(i,l)\*V\_RF(l,j);

end

end

% 1-bit resolution phase shifter

min\_abs\_sq=10000;

V\_RF\_quant\_min=V\_RF(i,j);

for i\_phase=1:length(phase\_list)

candidate=abs(phase\_list(i\_phase)-eta\_ij\_quant/abs(eta\_ij\_quant))^2;

if candidate<min\_abs\_sq

min\_abs\_sq=candidate;

V\_RF\_quant\_min=phase\_list(i\_phase);

end

end

V\_RF\_quant(i,j)=V\_RF\_quant\_min;

end

end

% Check convergence

if abs(abs(V\_RF\_quant).^2-ones(N,Nr\_RF)) < tolerance\*ones(N,Nr\_RF)

not\_converge\_1bit=0;

end

end

%% Part 2: IV.A. Digital Precoder Design for N\_RF=Ns

Q=V\_RF'\*V\_RF;

Q\_quant=V\_RF\_quant'\*V\_RF\_quant;

Heff=H\*V\_RF;

Heff\_quant=H\*V\_RF\_quant;

[Left,Delta,U\_e]=svd(Heff\*pinv(sqrtm(Q))); % TODO U\_e' ?

[Left\_quant,Delta\_quant,U\_e\_quant]=svd(Heff\_quant\*pinv(sqrtm(Q\_quant)));

% REF https://scicoding.com/water-filling-algorithm-in-depth-explanation/#:~:text=Water%2Dfilling%20is%20a%20generic,in%20a%20technical%20sense%2C%20orthogonal

% REF https://zhuanlan.zhihu.com/p/502453127

Delta(N\_RF+1:M,:)=[];

Delta\_quant(N\_RF+1:M,:)=[];

N0=sigma^2; % NOTE MIMO.pdf one line below formula 7.12

lambda\_list=diag(Delta);

lambda\_list(Ns+1:end,:)=[];

lambda\_quant\_list=diag(Delta\_quant);

lambda\_quant\_list(Ns+1:end,:)=[];

lambda\_sq\_list=lambda\_list.^2;

lambda\_quant\_sq\_list=lambda\_quant\_list.^2;

% NOTE water-filling solution: refer to MIMO.pdf formula 7.11 and above

% NOTE assume Nmin (mu?) interval (0, /2, /2, /2, ...)

% to obtain Nmin

Nmin=1;

mu=0;

while Nmin<Ns && ~(mu>=sigma^2/lambda\_sq\_list(Nmin) && mu<=sigma^2/lambda\_sq\_list(Nmin+1)) % TODO >= <= or > < ?

Nmin=Nmin+1;

mu=(P+sum(sigma^2./lambda\_sq\_list(1:Nmin)))/Nmin; % TODO ./!!!

end

Nmin\_quant=1;

mu\_quant=0;

while Nmin\_quant<Ns && ~(mu\_quant>=sigma^2/lambda\_sq\_list(Nmin\_quant) && mu\_quant<=sigma^2/lambda\_sq\_list(Nmin\_quant+1)) % TODO >= <= or > < ?

Nmin\_quant=Nmin\_quant+1;

mu\_quant=(P+sum(sigma^2./lambda\_quant\_sq\_list(1:Nmin\_quant)))/Nmin\_quant; % TODO ./!!!

end

P\_star\_list=max(mu-N0./lambda\_sq\_list,0);

P\_star\_list(Nmin+1:end)=[];

P\_star\_quant\_list=max(mu\_quant-N0./lambda\_quant\_sq\_list,0);

P\_star\_quant\_list(Nmin\_quant+1:end)=[];

Gamma\_e=diag([transpose(sqrt(P\_star\_list)) zeros(1,N\_RF-Nmin)]);

Gamma\_e\_quant=diag([transpose(sqrt(P\_star\_quant\_list)) zeros(1,N\_RF-Nmin\_quant)]);

V\_D=pinv(sqrtm(Q))\*U\_e\*Gamma\_e;

V\_D\_quant=pinv(sqrtm(Q\_quant))\*U\_e\_quant\*Gamma\_e\_quant;

%% Part 3: IV.C Hybrid Combining Design for N\_RF=NS

% Necessary parameters

W\_RF=ones(M,N\_RF);

W\_RF\_quant=ones(M,N\_RF);

Vt=V\_RF\*V\_D;

Vt\_quant=V\_RF\_quant\*V\_D\_quant;

F2=H\*(Vt)\*Vt'\*H';

F2\_quant=H\*(Vt\_quant)\*Vt\_quant'\*H';

% Translating algorithm 1 to design W\_RF for infinite phase shifter

not\_converge\_inf=1;

while not\_converge\_inf

for j=1:N\_RF

W\_RF\_noj=W\_RF;

W\_RF\_noj(:,j)=[];

big=W\_RF\_noj'\*F2\*W\_RF\_noj;

C\_j=eye(size(big))+(1/M/sigma^2)\*big;

G\_j=(1/M/sigma^2)\*F2-(1/M/sigma^2)^2\*F2\*W\_RF\_noj\*pinv(C\_j)\*W\_RF\_noj'\*F2;

for i=1:M

eta\_ij=0;

for l=1:M

if l~=i

eta\_ij=eta\_ij+G\_j(i,l)\*W\_RF(l,j);

end

end

if eta\_ij==0

W\_RF(i,j)=1;

else

W\_RF(i,j)=eta\_ij/abs(eta\_ij);

end

end

end

% Check convergence

if abs(abs(W\_RF).^2-ones(M,N\_RF)) < tolerance\*ones(M,N\_RF)

not\_converge\_inf=0;

end

end

% Translating algorithm 1 to design W\_RF for 1-bit resolution phase shifter

not\_converge\_1bit=1;

while not\_converge\_1bit

for j=1:N\_RF

W\_RF\_noj\_quant=W\_RF\_quant;

W\_RF\_noj\_quant(:,j)=[];

big\_quant=W\_RF\_noj\_quant'\*F2\_quant\*W\_RF\_noj\_quant;

C\_j\_quant=eye(size(big\_quant))+(1/M/sigma^2)\*big\_quant;

G\_j\_quant=(1/M/sigma^2)\*F2\_quant-(1/M/sigma^2)^2\*F2\_quant\*W\_RF\_noj\_quant\*pinv(C\_j\_quant)\*W\_RF\_noj\_quant'\*F2\_quant;

for i=1:M

eta\_ij\_quant=0;

for l=1:M

if l~=i

eta\_ij\_quant=eta\_ij\_quant+G\_j\_quant(i,l)\*W\_RF\_quant(l,j);

end

end

% 1-bit resolution phase shifter

min\_abs\_sq=10000;

W\_RF\_quant\_min=W\_RF(i,j);

for i\_phase=1:length(phase\_list)

candidate=abs(phase\_list(i\_phase)-eta\_ij\_quant/abs(eta\_ij\_quant))^2;

if candidate<min\_abs\_sq

min\_abs\_sq=candidate;

W\_RF\_quant\_min=phase\_list(i\_phase);

end

end

W\_RF\_quant(i,j)=W\_RF\_quant\_min;

end

end

% Check convergence

if abs(abs(W\_RF\_quant).^2-ones(M,N\_RF)) < tolerance\*ones(M,N\_RF)

not\_converge\_1bit=0;

end

end

% Design W\_D

J=W\_RF'\*H\*(Vt)\*Vt'\*H'\*W\_RF+sigma^2\*(W\_RF')\*W\_RF;

J\_quant=W\_RF\_quant'\*H\*(Vt\_quant)\*Vt\_quant'\*H'\*W\_RF\_quant+sigma^2\*(W\_RF\_quant')\*W\_RF\_quant;

W\_D=pinv(J)\*W\_RF'\*H\*Vt;

W\_D\_quant=pinv(J\_quant)\*W\_RF\_quant'\*H\*Vt\_quant;

Wt=W\_RF\*W\_D;

Wt\_quant=W\_RF\_quant\*W\_D\_quant;

% Spectral efficiency

R=R+log2(det(eye(M,M)+(Wt\*pinv(Wt'\*Wt)\*Wt'\*H\*(Vt)\*Vt'\*H')/sigma^2));

R\_quant=R\_quant+log2(det(eye(M,M)+(Wt\_quant\*pinv(Wt\_quant'\*Wt\_quant)\*Wt\_quant'\*H\*(Vt\_quant)\*Vt\_quant'\*H')/sigma^2));

end

% Uncomment for results 1, 2 ↓

R\_list(i\_snr)=R/num\_mc;

R\_quant\_list(i\_snr)=R\_quant/num\_mc;

% Uncomment for results 1, 2 ↑

% Uncomment for result 3 ↓

%R\_list(i\_nrf,i\_snr)=R/num\_mc;

%R\_quant\_list(i\_nrf,i\_snr)=R\_quant/num\_mc;

% Uncomment for result 3 ↑

end

end

% Uncomment for result 1 ↓

%figure;

%plot(SNR,R\_list);

%grid on;

%xlabel("SNR(dB)");

%ylabel("Spectral efficiency (bits/s/Hz)");

% NOTE spectral efficiency unit means: bit rate that this frequency can afford

%legend("N^{RF}=N\_{s}","N^{RF}=N\_{s}+1","N^{RF}=N\_{s}+3");

%title("SNR vs. Spectral Efficiency (Infinite phase shifter, "+N+"\*"+M+" MIMO, single user, Ns="+Ns+")");

% Uncomment for result 2 ↑

% Uncomment for result 2 ↓

figure;

plot(SNR,R\_list,SNR,R\_quant\_list);

grid on;

legend("b=\infty","b=1");

xlabel("SNR(dB)");

ylabel("Spectral efficiency (bits/s/Hz)");

% NOTE spectral efficiency unit means: bit rate that this frequency can afford

title("SNR vs. Spectral Efficiency ("+N+"\*"+M+" MIMO, single user, Ns="+Ns+")");

% Uncomment for result 2 ↑

% Uncomment for result 3 ↓

%figure;

%plot(SNR,R\_list);

%grid on;

%xlabel("SNR(dB)");

%ylabel("Spectral efficiency (bits/s/Hz)");

%legend("N^{RF}=N\_{s}","N^{RF}=N\_{s}+1","N^{RF}=N\_{s}+3");

%title("SNR vs. Spectral Efficiency (Infinite phase shifter, "+N+"\*"+M+" MIMO, single user, Ns="+Ns+")");

%figure;

%plot(SNR,R\_quant\_list);

%grid on;

%xlabel("SNR(dB)");

%ylabel("Spectral efficiency (bits/s/Hz)");

%legend("N^{RF}=N\_{s}","N^{RF}=N\_{s}+1","N^{RF}=N\_{s}+3");

%title("SNR vs. Spectral Efficiency (1-bit phase shifter, "+N+"\*"+M+" MIMO, single user, Ns="+Ns+")");

% Uncomment for result 3 ↑