Table of Contents

..... 1
params 1
precoding 1
zero forcing 2
MMSE 3
plot 3


```
%Vishnu Kaimal
%DongKyu Kim
%MIMO/OFDM
%MIMO part
%ECE408 Wireless Communications
%Professor Hoerning

clc; clear all; close all;
```

params

```
numIter = 3;  % The number of iterations of the simulation
nSym = 1e4;    % The number of symbols per packet
M = 16;        % Modulation order
Mt = 2;     %number of transmitters
Mr = 2;     %number of receivers

EbNo = -10:1:30; %EbNo range to iterate over for plot
SNR_Vec = EbNo + 10*log10(log2(M)); %SNR conversion from EbNo
lenSNR = length(SNR_Vec);
ber = zeros(3,numIter,lenSNR); %ber store
H =
 sqrt(1/2)*(randn(Mr,Mt,nSym*log2(M),numIter)+1j*randn(Mr,Mt,nSym*log2(M),numIter)
 for each iteration
```

precoding

```
U = zeros(Mr,Mt,nSym*log2(M));
V = U;
S = U;
tx_chan = zeros(Mr,1,nSym*log2(M));
tx_process = tx_chan;

for i = 1:numIter

    bits = randi([0 M-1],Mt, 1, nSym*log2(M));     % Generate random
 bits
    tx = qammod(bits,M);  % modulate the signal

   %construction of flat fading channel
```

```
    for k=1:nSym*log2(M)
        [U(:,:,k),S(:,:,k),V(:,:,k)] = svd(H(:,:,k,i));
        tx_chan(:,:,k) = H(:,:,k,i)*V(:,:,k)*tx(:,:,k);
        %Receiver chooses M = V' (Goldsmith Notation) MATLAB svd is
        %actually H = U*S*V'
    end

    for j = 1:lenSNR
         noise =
 sqrt(1/2)*(randn(Mr,1,nSym*log2(M))+1j*randn(Mr,1,nSym*log2(M)));
        txNoisy = tx_chan + 10^(-1*SNR_Vec(j)/20)*noise;
        for k=1:nSym*log2(M)
            tx_process(:,:,k) = S(:,:,k)^-1*U(:,:,k)'*txNoisy(:,:,k);
            % Receiver receives with F = U' (Goldsmith Notation)
        end

        rx = qamdemod(tx_process,M);

        % Compute and store the BER for this iteration
        [~, ber(1,i,j)] = biterr(bits, rx);

   end
end
```

zero forcing

```
for i = 1:numIter

    bits = randi([0 M-1],Mt, 1, nSym*log2(M));     % Generate random
 bits
    tx = qammod(bits,M);  % modulate the signal

    %construction of flat fading channel

    for k=1:nSym*log2(M)
        tx_chan(:,:,k) = H(:,:,k,i)*tx(:,:,k);
    end


    for j = 1:lenSNR

        noise =
 sqrt(1/2)*(randn(Mr,1,nSym*log2(M))+1j*randn(Mr,1,nSym*log2(M)));
        txNoisy = tx_chan + 10^(-1*SNR_Vec(j)/20)*noise;

        for k=1:nSym*log2(M)
            W(:,:,k) = (H(:,:,k,i)'*H(:,:,k,i))^-1*H(:,:,k,i)';
            tx_process(:,:,k) = W(:,:,k)*txNoisy(:,:,k);
        end


        rx = qamdemod(tx_process,M);
```

```
        % Compute and store the BER for this iteration
        [~, ber(2,i,j)] = biterr(bits, rx);

    end
end
```

MMSE

```
for i = 1:numIter

    bits = randi([0 M-1],Mt, 1, nSym*log2(M));     % Generate random
 bits
    tx = qammod(bits,M);  % modulate the signal

    %construction of flat fading channel

    for k=1:nSym*log2(M)
        tx_chan(:,:,k) = H(:,:,k,i)*tx(:,:,k);
    end

    for j = 1:lenSNR

        noise =
 sqrt(1/2)*(randn(Mr,1,nSym*log2(M))+1j*randn(Mr,1,nSym*log2(M)));
        txNoisy = tx_chan + 10^(-1*SNR_Vec(j)/20)*noise;

        for k=1:nSym*log2(M)
            W(:,:,k) =
 (H(:,:,k,i)'*H(:,:,k,i)+eye(Mt)*10^(-1*SNR_Vec(j)/20))^-1*H(:,:,k,i)';
            tx_process(:,:,k) = W(:,:,k)*txNoisy(:,:,k);
        end

        rx = qamdemod(tx_process,M);

        % Compute and store the BER for this iteration
        [~, ber(3,i,j)] = biterr(bits, rx);

    end
end
```

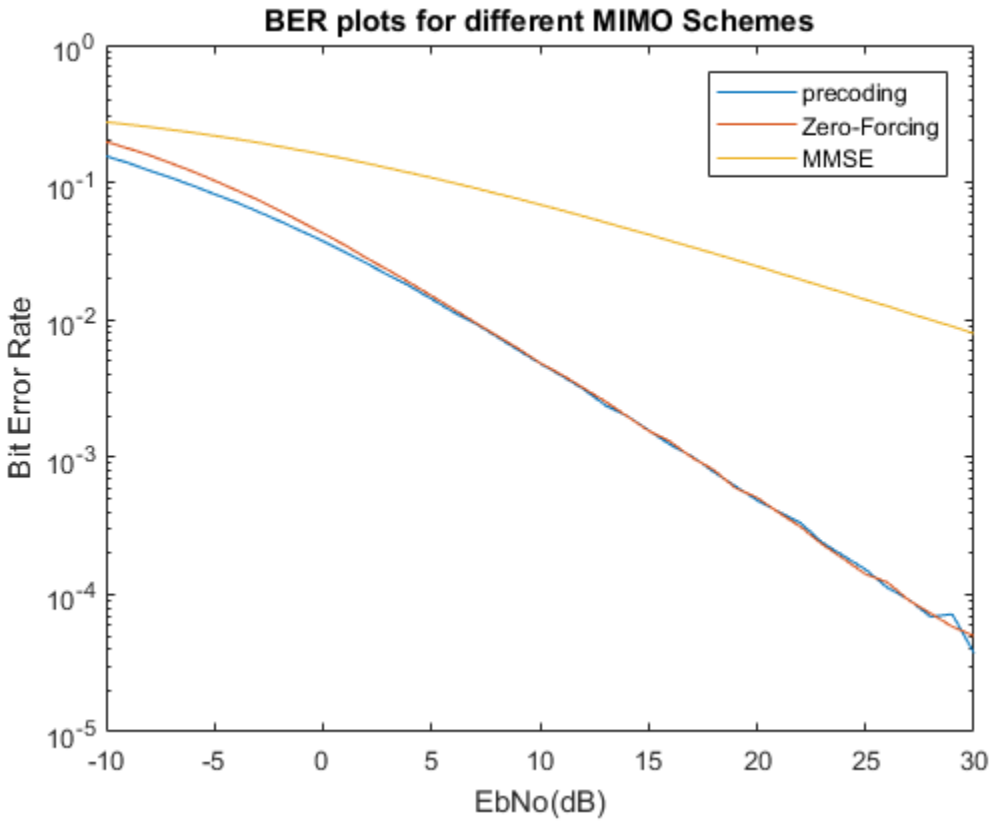
plot

```
ber_m = mean(ber,2); %take mean across all iterations

%plotting ber for different equalization techniques
semilogy(EbNo,ber_m(1,:),'DisplayName','precoding');
hold on;
semilogy(EbNo,ber_m(2,:),'DisplayName','Zero-Forcing');
hold on;
semilogy(EbNo,ber_m(3,:),'DisplayName','MMSE');

title('BER plots for different MIMO Schemes');
```

```
xlabel('EbNo(dB)');
ylabel('Bit Error Rate');
legend('show');
```



Published with MATLAB® R2017a