Table of Contents

1
params 1
ofdm zero forcing 1
ofdm mmse 2
plot 3

```
%Vishnu Kaimal
%DongKyu Kim
%MIMO/OFDM
%OFDM part

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;        % Binary Modulation

EbNo = -10:1:30; %EbNo range to iterate over for plot
SNR_Vec = EbNo + 10*log10(64/80); %SNR conversion from EbNo
lenSNR = length(SNR_Vec);

index = [1:5 7:19 21:26 28:33 35:47 49:53]+5; %frame parameters
index_pilot = [6 20 34 48]+5; % frame parameters

%params for rayleigh frequency selective channel
Ts = 1e-3;
Fd = 0;
tau = [0 1e-5 3.5e-5 12e-5];
pdb = [0 -1 -1 -3];

%ber store
ber = zeros(2,lenSNR,numIter);
```

ofdm zero forcing

```
for i = 1:numIter

    bits = randi([0,M-1],1, nSym*48);     % Generate random bits
    mod_data = qammod(bits,M);  % modulate the signal

    ofdm_data = reshape(mod_data,48,[]);
    ofdm_frame = zeros(64,nSym);
    ofdm_frame(index,:) = ofdm_data;
    ofdm_frame(index_pilot,:) = 1;

    ifft_ofdm = ifft(ofdm_frame,64);
    ofdm_trans = [ifft_ofdm(49:64,:); ifft_ofdm]; %guard

    %construction of frequency selective channel
    h = rayleighchan(Ts, Fd, tau, pdb);
    chan = zeros(80,nSym);
    ofdm_chan = zeros(80,nSym);
    for k=1:nSym
        chan(:,k) = filter(h,ones(80,1));
        ofdm_chan(:,k) = chan(:,k).*ofdm_trans(:,k); %apply channel to
 signal
    end

    for j = 1:lenSNR
        noise = sqrt(1/2)*(randn(80,nSym)+1j*randn(80,nSym));
        ofdm_noisy = ofdm_chan + 10^(-1*SNR_Vec(j)/20)*noise;


       ofdm_no_guard = ofdm_noisy(17:end,:);
       ofdm_orig_frame = fft(ofdm_no_guard,64);

       ofdm_zf = ofdm_orig_frame./chan(17:end,:);
       ofdm_rcv_data = ofdm_zf(index,:);
       mod_rcv_data = reshape(ofdm_rcv_data,1,[]);

       rx = qamdemod(mod_rcv_data,M);
       [~, ber(1,j,i)] = biterr(bits, rx);

    end

end
```

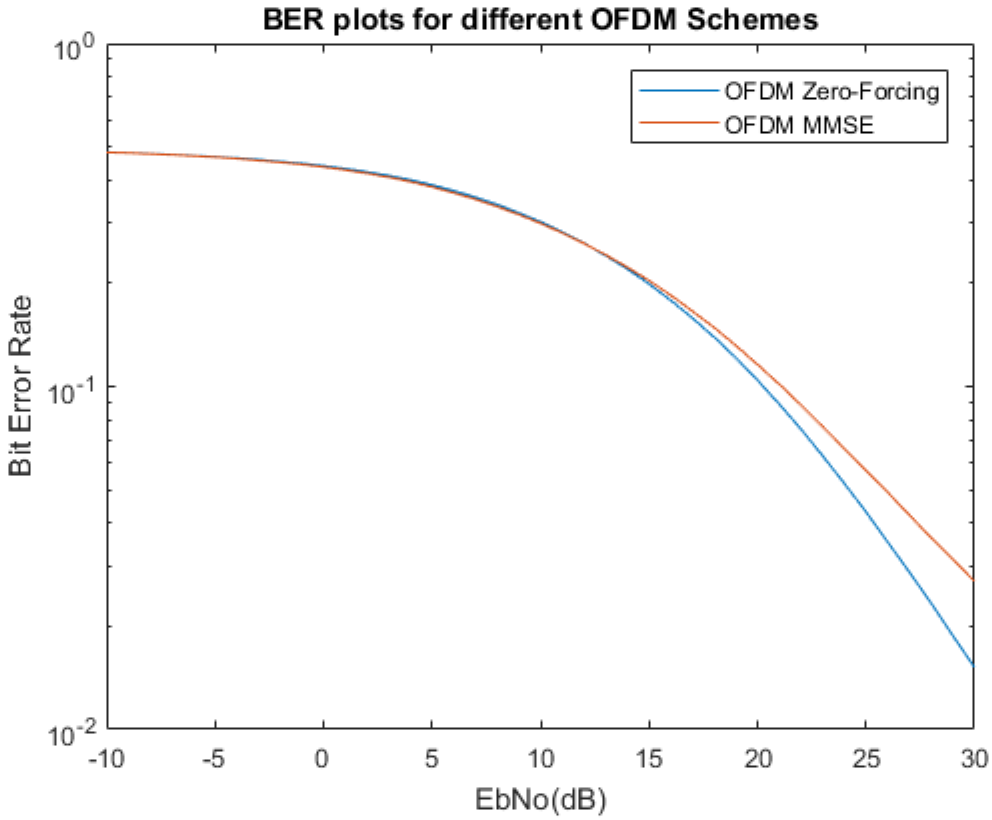
ofdm mmse

```
for i = 1:numIter

    bits = randi([0,M-1],1, nSym*48);     % Generate random bits
    mod_data = qammod(bits,M);  % modulate the signal

    ofdm_data = reshape(mod_data,48,[]);
    ofdm_frame = zeros(64,nSym);
    ofdm_frame(index,:) = ofdm_data;
    ofdm_frame(index_pilot,:) = 1;

    ifft_ofdm = ifft(ofdm_frame,64);
    ofdm_trans = [ifft_ofdm(49:64,:); ifft_ofdm]; %guard


    %construction of frequency selective channel
    h = rayleighchan(Ts, Fd, tau, pdb);
    for k=1:nSym
        chan(:,k) = filter(h,ones(80,1));
        ofdm_chan(:,k) = chan(:,k).*ofdm_trans(:,k); %apply channel to
 signal
```

```
    end

    for j = 1:lenSNR
        noise = sqrt(1/2)*(randn(80,nSym)+1j*randn(80,nSym));
        snr = 10^(-1*SNR_Vec(j)/20);
        ofdm_noisy = ofdm_chan + snr*noise;


       ofdm_no_guard = ofdm_noisy(17:end,:);
       ofdm_orig_frame = fft(ofdm_no_guard,64);

       norm = conj(chan(17:end,:)).*chan(17:end,:) + snr;
       ofdm_zf = ofdm_orig_frame.*conj(chan(17:end,:))./norm;
       ofdm_rcv_data = ofdm_zf(index,:);
       mod_rcv_data = reshape(ofdm_rcv_data,1,[]);

       rx = qamdemod(mod_rcv_data,M);
       [~, ber(2,j,i)] = biterr(bits, rx);

    end

end
```

plot

```
ber = mean(ber,3); %take mean across all iterations

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


title('BER plots for different OFDM Schemes');
xlabel('EbNo(dB)');
ylabel('Bit Error Rate');
legend('show');
```



Published with MATLAB® R2017a