ECE 233 Wireless Communications System Design, Modeling, and Implementation.

MIMO, and RF Impairments

By:
Akshay Joshi, UCLA

MATLAB Code:

Spectral Regrowth function:

```
function [x,y] = spectral_regrowth(M,b)
    phi1=pi/6;
    phi2=pi/4.5;
    h1=exp(-1j*pi*(0:M-1)*sin(phi1));
    h2=exp(-1j*pi*(0:M-1)*sin(phi2));
    H=[h1' h2']';
    P=H'*inv(H*H');
    m_order=4;
    b_in_s1=randi([0 1], log2(m_order)*1000,1);
    b_in_s2=randi([0 1], log2(m_order)*1000,1);
    s1 = qammod(b_in_s1, m_order, 'InputType', 'bit', 'UnitAveragePower', true);
    s2 = zeros(1000,1);
    s1_upsampled=upsample(s1,4);
    filtr=rcosdesign(0.5,8,4);
    s1_tilde=filter(filtr,1,s1_upsampled);
    s2_upsampled=upsample(s2,4);
    s2_tilde=filter(filtr,1,s2_upsampled);
    s_tilde=[s1_tilde s2_tilde]';
    x=P*s_tilde;
    range=linspace(-1/M,1/M,2^b);
    y_real=[];
    y_imag=[];
    y=zeros(M,4000);
    for n=1:size(x,2)
        y_real=[];
        y_imag=[];
        for m=1:size(x,1)
            realpart=real(x(m,n));
            imagpart=imag(x(m,n));
            [valreal,realidx]=min(abs(range-realpart));
            [valimag,imagidx]=min(abs(range-imagpart));
            y_real=[y_real range(realidx)];
            y_imag=[y_imag range(imagidx)];
```

```
        end
        y(:,n)=complex(y_real,y_imag);
    end
end
```

Main Code:

```
Nt=32;
dt=1/2;
dr=1/2;
L=4;
Pt=0.001;
B=200*10^3;
No=-174;
No=10^((No-30)/10);
Nr=4:2:32;
c_r=zeros(length(Nr),1000);
c_s=zeros(length(Nr),1000);
c_r_fin=zeros(1,length(Nr));
c_s_fin=zeros(1,length(Nr));
for r=1:length(Nr)
    for m=1:1000
        Hs=zeros(Nr(r),Nt);
        Hr=randn(Nr(r),Nt);
        for l=1:L
            thetai=unifrnd(-pi/2,pi/2);
            phii=unifrnd(-pi/2,pi/2);
            alphai=normrnd(0,sqrt(max(Nt,Nr(r))));
            arx_thetai = exp(-1j*2*pi*(0:Nr(r)-1)'*dr*sin(alphai))/sqrt(Nr(r));
            atx_phii = exp(-1j*2*pi*(0:Nt-1)'*dt*sin(phii))/sqrt(Nt);
            Hs=Hs+alphai*arx_thetai*atx_phii';
        end
        kr=rank(Hr);
        ks=rank(Hs);
        [Us, Ss, Vs]=svd(Hs);
        [Ur, Sr, Vr]=svd(Hr);
        for k=1:kr
            c_r(r,m)=c_r(r,m)+log2(1+(Nr(r)*Nt*Pt*abs(Sr(k,k))^2/(kr*B*No)));
        end
        c_r(r,m)=B*c_r(r,m);
        for k=1:ks
```

```
            c_s(r,m)=c_s(r,m)+log2(1+(Nr(r)*Nt*Pt*abs(Ss(k,k))^2/(ks*B*No)));
        end
        c_s(r,m)=B*c_s(r,m);
    end
    c_r_fin(r)=mean(c_r(r,:));
    c_s_fin(r)=mean(c_s(r,:));
end
%%
figure
semilogy(Nr,c_r_fin,Nr,c_s_fin, 'linewidth',2)
legend('Rich Scattering Capacity','Sparse Scattering Capacity'), title('\fontsize{14}Channel
Capacity'), xlabel('\fontsize{12}Nr'), ylabel('\fontsize{12}Rate in bps')
%% Problem 2 Question 2
Nt=32;
dt=1/2;
dr=1/2;
L=4;
Pt=0.001;
B=200*10^3;
No=-174;
No=10^((No-30)/10);
Nr=4:2:32;
achieveable_r=zeros(length(Nr),1000);
achieveable_s=zeros(length(Nr),1000);
achieveable_r_fin=zeros(1,length(Nr));
achieveable_s_fin=zeros(1,length(Nr));
achieveable_r_unk=zeros(length(Nr),1000);
achieveable_s_unk=zeros(length(Nr),1000);
achieveable_r_unk_fin=zeros(1,length(Nr));
achieveable_s_unk_fin=zeros(1,length(Nr));
ftr_angle=-pi/2:pi/Nt:-(-pi/2+pi/Nt);
ftr=exp(-1j*2*pi*(0:Nt-1)'*dt*sin(ftr_angle))/sqrt(Nt);
for r=1:length(Nr)
    wtr_angle=-pi/2:pi/Nr(r):-(-pi/2+pi/Nr(r));
    wtr=exp(-1j*2*pi*(0:Nr(r)-1)'*dt*sin(wtr_angle))/sqrt(Nr(r));
    for m=1:1000
        Hs=zeros(Nr(r),Nt);
        Hr=randn(Nr(r),Nt);
        for l=1:L
```

```
            thetai=unifrnd(-pi/2,pi/2);
            phii=unifrnd(-pi/2,pi/2);
            alphai=normrnd(0,sqrt(max(Nt,Nr(r))));
            arx_thetai = exp(-1j*2*pi*(0:Nr(r)-1)'*dr*sin(alphai))/sqrt(Nr(r));
            atx_phii = exp(-1j*2*pi*(0:Nt-1)'*dt*sin(phii))/sqrt(Nt);
            Hs=Hs+alphai*arx_thetai*atx_phii';
        end
        kr=rank(Hr);
        ks=rank(Hs);
        [Us, Ss, Vs]=svd(Hs);
        [Ur, Sr, Vr]=svd(Hr);
        for k=1:2

achieveable_r(r,m)=achieveable_r(r,m)+log2(1+(Nr(r)*Nt*Pt*abs(Sr(k,k))^2/(2*B*No)));
        end
        achieveable_r(r,m)=B*achieveable_r(r,m);
        for k=1:2

achieveable_s(r,m)=achieveable_s(r,m)+log2(1+(Nr(r)*Nt*Pt*abs(Ss(k,k))^2/(2*B*No)));
        end
        achieveable_s(r,m)=B*achieveable_s(r,m);
        y_abs_r=[];
        y_abs_s=[];
        for i=1:size(ftr,2)
            for j=1:size(wtr,2)
                y_r=wtr(:,j)'*Hr*ftr(:,i);
                y_abs_r_temp=abs(y_r)^2;
                y_s=wtr(:,j)'*Hs*ftr(:,i);
                y_abs_s_temp=abs(y_s)^2;
                y_abs_r=[y_abs_r y_abs_r_temp];
                y_abs_s=[y_abs_s y_abs_s_temp];
            end
        end
        yrarray=sort(y_abs_r,'descend');
        ysarray=sort(y_abs_s,'descend');
        for k=1:2

achieveable_r_unk(r,m)=achieveable_r_unk(r,m)+log2(1+(Nr(r)*Nt*Pt*yrarray(k)/(2*B*No)));
        end
```

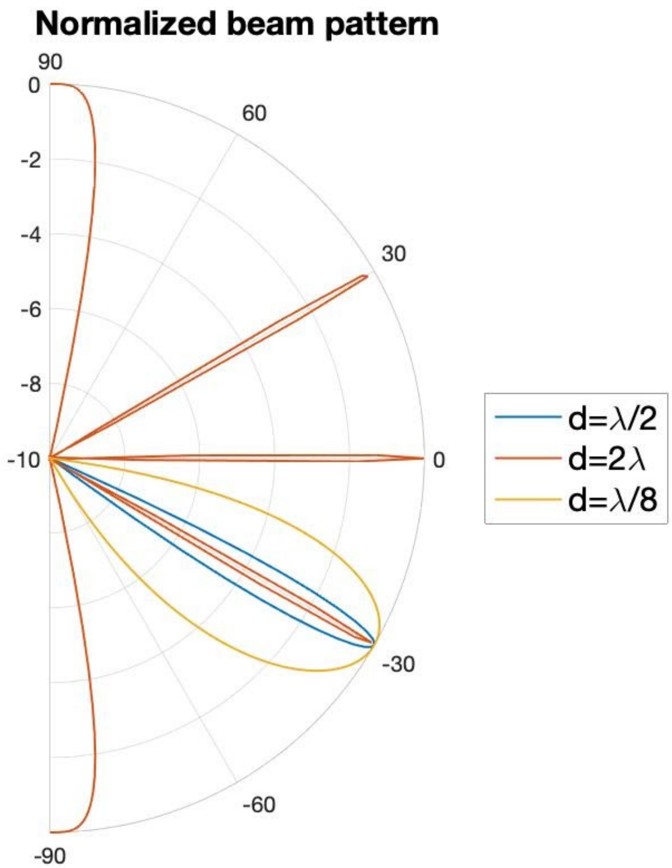
```
        achieveable_r_unk(r,m)=B*achieveable_r_unk(r,m);
        for k=1:2

achieveable_s_unk(r,m)=achieveable_s_unk(r,m)+log2(1+(Nr(r)*Nt*Pt*ysarray(k)/(2*B*No)));
        end
        achieveable_s_unk(r,m)=B*achieveable_s_unk(r,m);
    end
    achieveable_r_fin(r)=mean(achieveable_r(r,:));
    achieveable_s_fin(r)=mean(achieveable_s(r,:));
    achieveable_r_unk_fin(r)=mean(achieveable_r_unk(r,:));
    achieveable_s_unk_fin(r)=mean(achieveable_s_unk(r,:));
end
%%
ytick1=10^7:5*10^7:5*10^8;
yticklab=({'0.6x10^8','1.1x10^8','1.6x10^8','2.1x10^8','2.6x10^8','3.1x10^8','3.6x10^8'});
figure
semilogy(Nr,c_r_fin, 'linewidth',2)
hold on
semilogy(Nr,c_s_fin,'linewidth',2)
hold on
semilogy(Nr,achieveable_r_fin,'linewidth',2)
hold on
semilogy(Nr,achieveable_s_fin,'linewidth',2)
hold on
semilogy(Nr,achieveable_r_unk_fin,'linewidth',2)
hold on
semilogy(Nr,achieveable_s_unk_fin,'linewidth',2)
% ylim('auto')
yticks(ytick1)
yticklabels(yticklab)
legend('Rich Scattering Capacity','Sparse Scattering Capacity','Rich Scattering Achievable (Ch
known)','Sparse Scattering Achievable (Ch known)','Rich Scattering Achievable (Ch
unknown)','Sparse Scattering Achievable (Ch unknown)', 'FontSize',12)
title('Capacity, Channel known/unknown achievable rate', 'FontSize', 12)
xlabel('\fontsize{12}Nr'), ylabel('\fontsize{12}Rate in bps')
%% Problem 3 question 1,2
M_values=[8,32];
b_values=[4,12];
colors=['b','r';'k','m'];
```

```
phi_range=-pi/2:0.01:pi/2;
for m=1:2
    for b=1:2
        [x,y]=spectral_regrowth(M_values(m),b_values(b));
        figure(1)
        [pxx,ww]=periodogram(y(1,:));
        plot(ww/pi,10*log10(pxx),colors(m,b), 'linewidth',1.5)
        hold on
        legend('M=8 b=4','M=8 b=12','M=32 b=4','M=32 b=12', 'FontSize',15)
        title('Impact of DAC quantization in the frequency domain','FontSize',15)
        xlabel('\fontsize{13}Normalized Frequency'), ylabel('\fontsize{13}Power Spectral Density')
        phi_range=-pi/2:0.01:pi/2;
        a=zeros(M_values(m),length(phi_range));
        for p=1:length(phi_range)
            a(:,p)=exp(-1j*pi*(0:M_values(m)-1)'*sin(phi_range(p)));
        end
        z=y;
%         clear g
        for p=1:length(phi_range
            G=[];
            for n=1:4000
                G=[G abs(a(:,p)'*z(:,n))^2];
            end
            g(p)=rms(G);
        end
        figure(2)
        plot(phi_range*180/pi,10*log10(g), 'linewidth',2)
        hold on
        legend('M=8 b=4','M=8 b=12','M=32 b=4','M=32 b=12', 'FontSize',15)
        title('Impact of DAC quantization in the angular domain','FontSize',15)
        xlabel('\fontsize{13}Angle in degrees'), ylabel('\fontsize{13}Angular response')
    end
end
%% Problem 3 question 3,4
beta=[0,-133];
for m=1:2
    for be=1:2
        [x,y]=spectral_regrowth(M_values(m),b_values(b));
        z=zeros(M_values(m),4000);
```

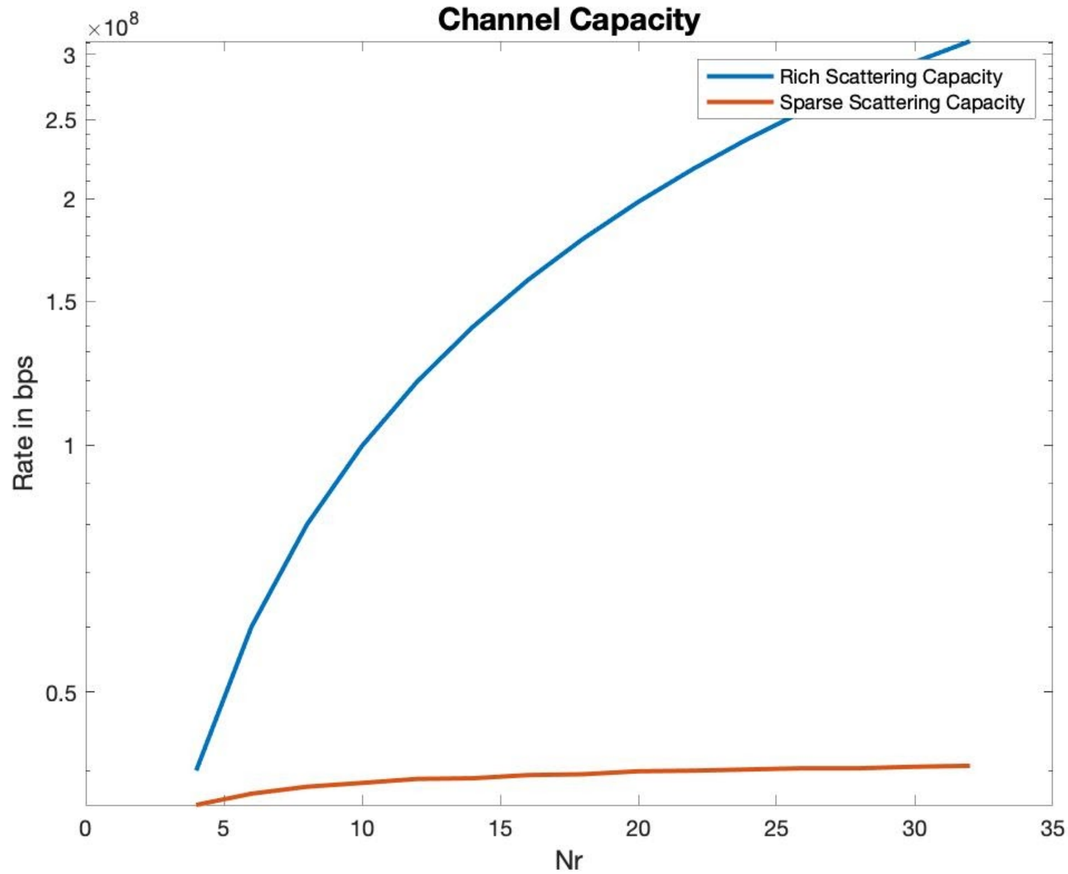
```
        for n=1:4000
            z(:,n)=x(:,n)+beta(be)*x(:,n).*abs(x(:,n)).^2;
        end
        figure(3)
        [pxx,ww]=periodogram(z(1,:));
        plot(ww/pi,10*log10(pxx),colors(m,be))
        hold on
        legend('M=8 \beta_3=0','M=8 \beta_3=-133','M=32 \beta_3=0','M=32 \beta_3=-133',
'FontSize',15)
        xlabel('\fontsize{13}Normalized Frequency'), ylabel('\fontsize{13}Power Spectral Density')
        title('Impact of PA non-linearity in the frequency domain','FontSize',15)
        a=zeros(M_values(m),length(phi_range));
        for p=1:length(phi_range)
            a(:,p)=exp(-1j*pi*(0:M_values(m)-1)'*sin(phi_range(p)));
        end
        for p=1:length(phi_range)
            G=[];
            for n=1:4000
                G=[G abs((a(:,p)'*z(:,n)))^2];
            end
            g(p)=rms(G);
        end
        figure(4)
        plot(phi_range*(180/pi),10*log10(g), 'linewidth',2)
        hold on
        legend('M=8 \beta_3=0','M=8 \beta_3=-133','M=32 \beta_3=0','M=32 \beta_3=-133',
'FontSize',15)
        title('Impact of PA non-linearity in the angular domain','FontSize',15)
        xlabel('\fontsize{13}Angle in degrees'), ylabel('\fontsize{13}Angular response')
    end
end
```

Problem 1.1



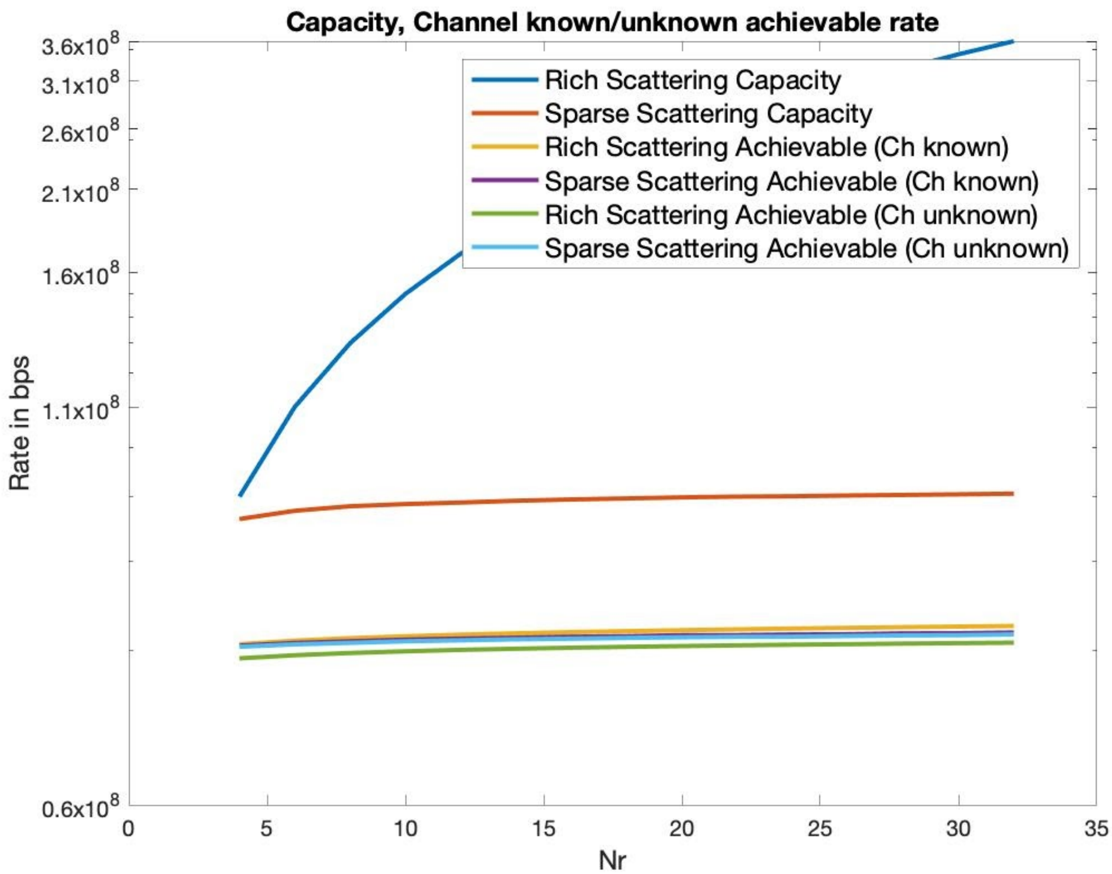
As evident from the plot, when d=λ/8 (in yellow), there are no secondary lobes but the directivity is lower, i.e., the main lobe is wider. When d=λ/2 (in blue), there are no sidelobes, but the directivity is increased i.e. the mainlobe width is decreased than when d was equal to λ/8. When d is further increased to 2λ (in red), the directivity is increased even further but more sidelobes appear. Thus, it can be concluded from the plot that as the spacing between antennas increases, there is a decrease in mainlobe width at an expense of secondary lobes.

Problem 2.1



Rich scattering environment contains a large number of scatterers whereas a sparse scattering environment contains a smaller number of scatterers. The capacity of the channel actually depends on the rank of the channel matrix. Higher the rank, higher is the capacity. In rich scattering, as the number of receivers increases, so does the channel matrix rank. This means that there are high numbers of positive singular values which lead to increase in capacity. The sparse scattering channel matrix has a lower rank because of lesser number of scatterers even if the number of receivers increases. Thus even when Nr increases, the less number of positive singular values due to low rank, doesn't allow the capacity to increase significantly.

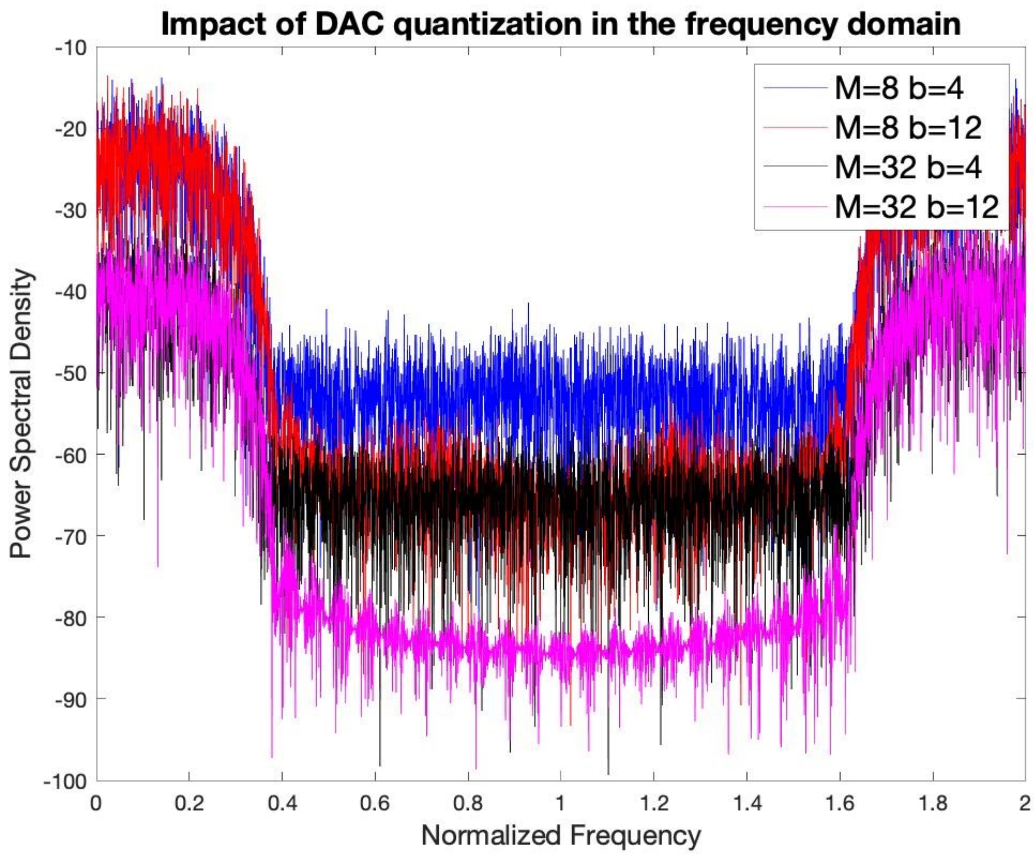
Problem 2.2



The achievable rates are fairly lesser than the capacity of the channels as expected. The SVD method i.e. when the channel matrix is known gives better achievable rate than beamforming method (when the channel matrix is unknown). In the SVD method we can be sure of choosing our 2 best vectors since the singular values are arranged in descending order in the sigma matrix and thus we pick top two beamforming pairs (U and V^H matrices). When the channel matrix is not known, we sweep through the angles from $-\frac{\pi}{2}$ to $+\frac{\pi}{2}$ in steps of $\frac{\pi}{Number\ of\ antennas}$ to find the best pairs of precoding vectors (f_{tr}) and combining vectors (w_{tr}) corresponding to each angle. This leads to finding the optimal pairs **given the angles**.

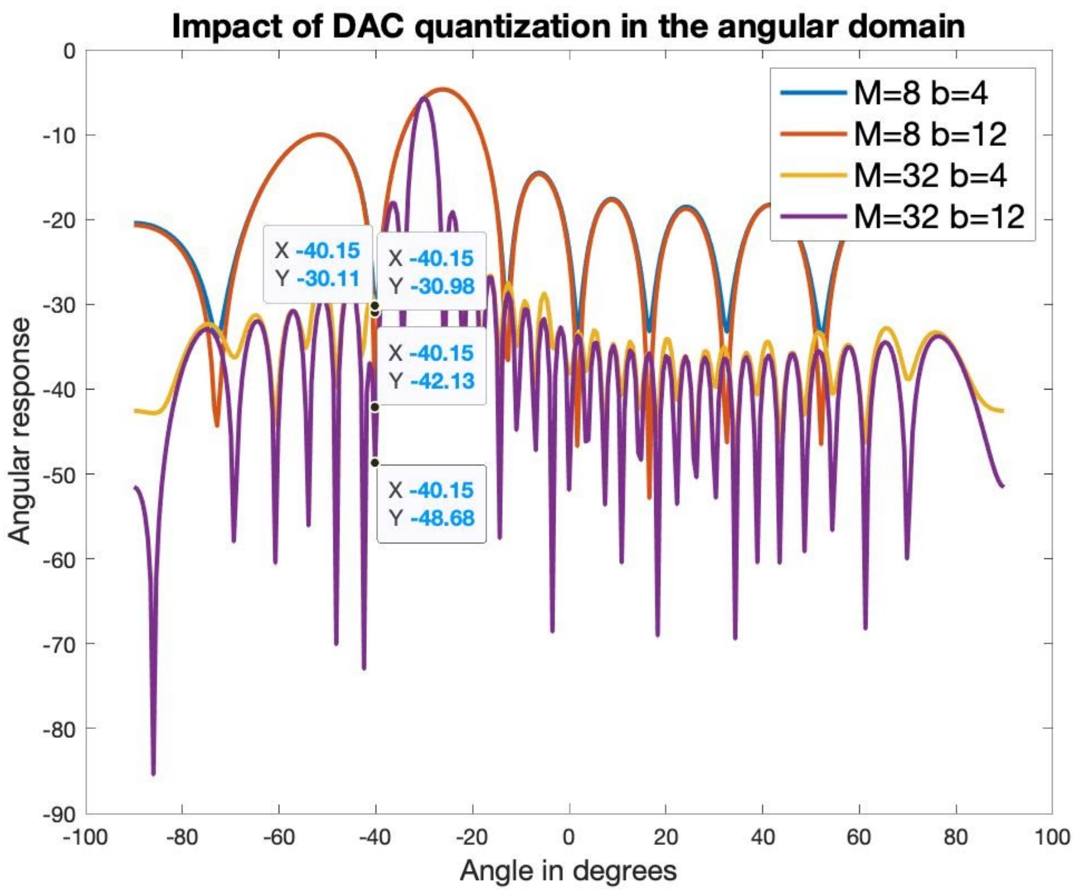
Beamforming gives a lower rate since SVD gives us the best possible combination of these vectors but beamforming sweeps through the angles to find those pairs which obviously isn't as optimal as those obtained using SVD.

Problem 3.1



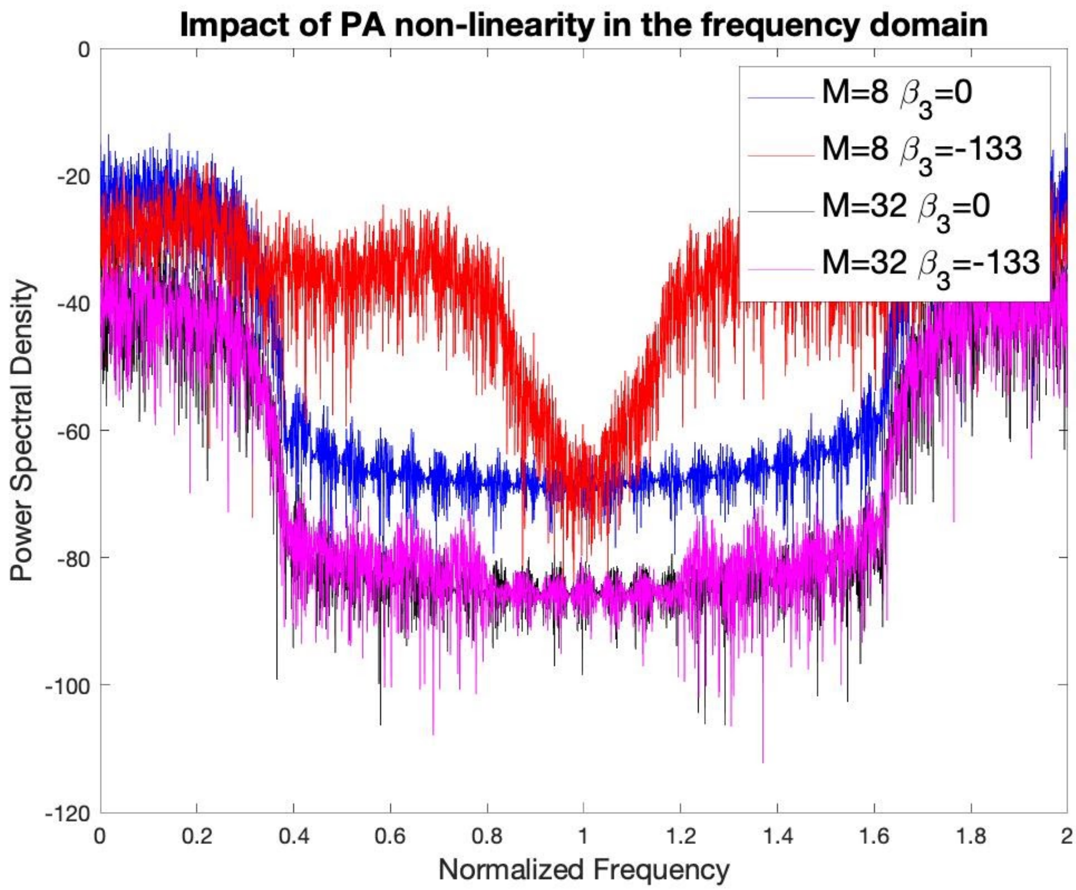
We see maximum spectral regrowth when M=8, b=4 (blue) and minimum spectral regrowth when M=32, b=12 (magenta). This happens due to the finite resolution of the DAC. We can say it happens because when b is low, the resolution of the DAC quantization is low and that coupled with a lesser number of antennas leads to out-of-band transmission.

Problem 3.2



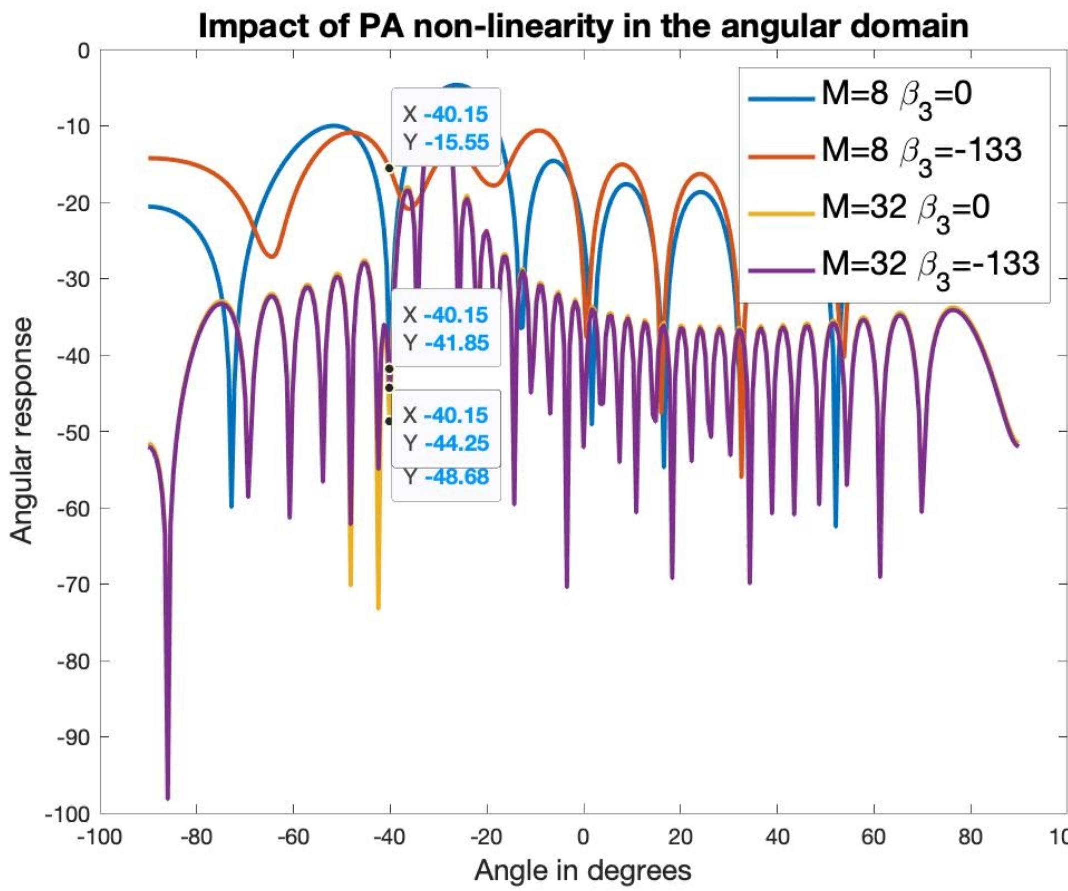
We see maximum interference to UE-2 i.e. $g(40°)$ is maximum when M=8, b=4 and minimum interference i.e. $g(40°)$ is minimum when M=32, b=12. Here as well, the finite resolution of the DAC has limitations. When b is low, the resolution of the DAC quantization is low and that coupled with a lesser number of antennas leads to higher interference than when M and b have high values.

Problem 3.3



We see maximum spectral regrowth when M=8, $\beta_3 = -133$ (red) and minimum spectral regrowth when M=32, $\beta_3 = -133$ (black) as well as when M=32, $\beta_3 = -133$ (magenta). It happens because when there is non-linearity in PA, intermodulation distortion occurs which causes the signal to be replicated at other frequencies. Thus, when β_3 is not zero and the number of antennas is less, we see out-of-band transmission.

Problem 3.4



We see maximum interference to UE-2 i.e. $g(40°)$ is maximum when M=8, $\beta_3 = -133$ and minimum interference i.e. $g(40°)$ is minimum when M=32, $\beta_3 = 0$. Here as well, this happens because when β_3 is not zero, i.e., when there is non-linearity in the PA, intermodulation distortion occurs and that leads to interference in the unintended direction.