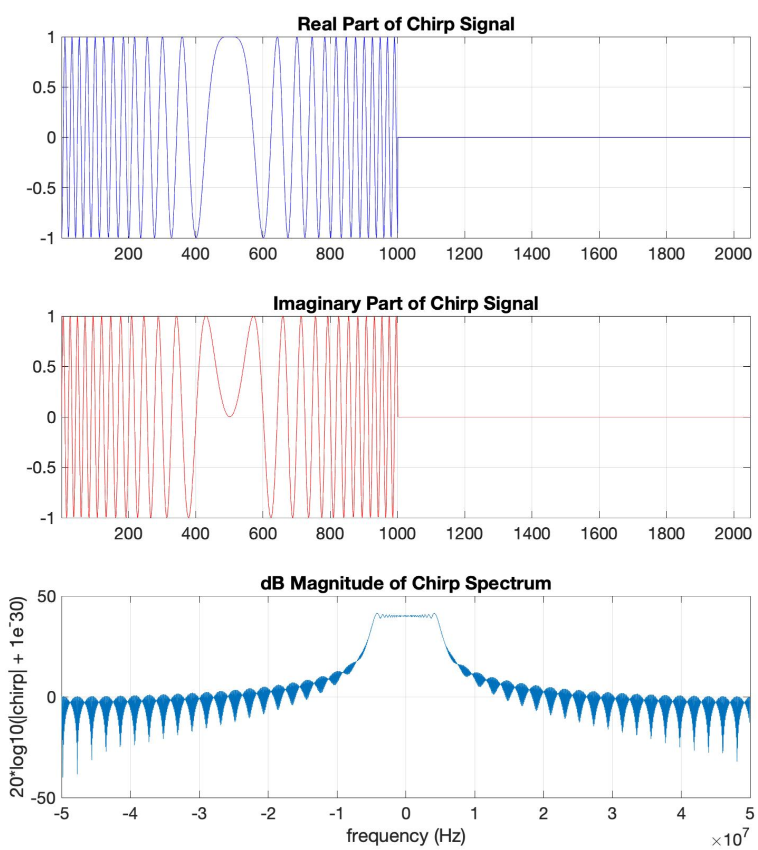
HW 2 Solutions
Imaging Radar Winter 2020

This and future homework will require making many reference chirps. So, we first create a function that makes the chirp based on known parameters.

```
function chirp = makechirp(s,tau,fs,fc,start,n)
	%Function to compute chirp - reused in all problems
	%s: slope
	%tau: pulse length
	%fs: sample rate
	%fc: center frequency
	%start: starting index of chirp
	%n: the length of the chirp including zero

	dt=1/fs;
	npts=tau*fs;
	t=[-npts/2:npts/2-1]*dt;
	phase=pi*s*t.^2+2*pi*fc*t;
	chirp=[zeros(1,start-1) exp(1i*phase) zeros(1,n-length(phase)-
start+1)];
end
```

Question 1: Chirp Compression

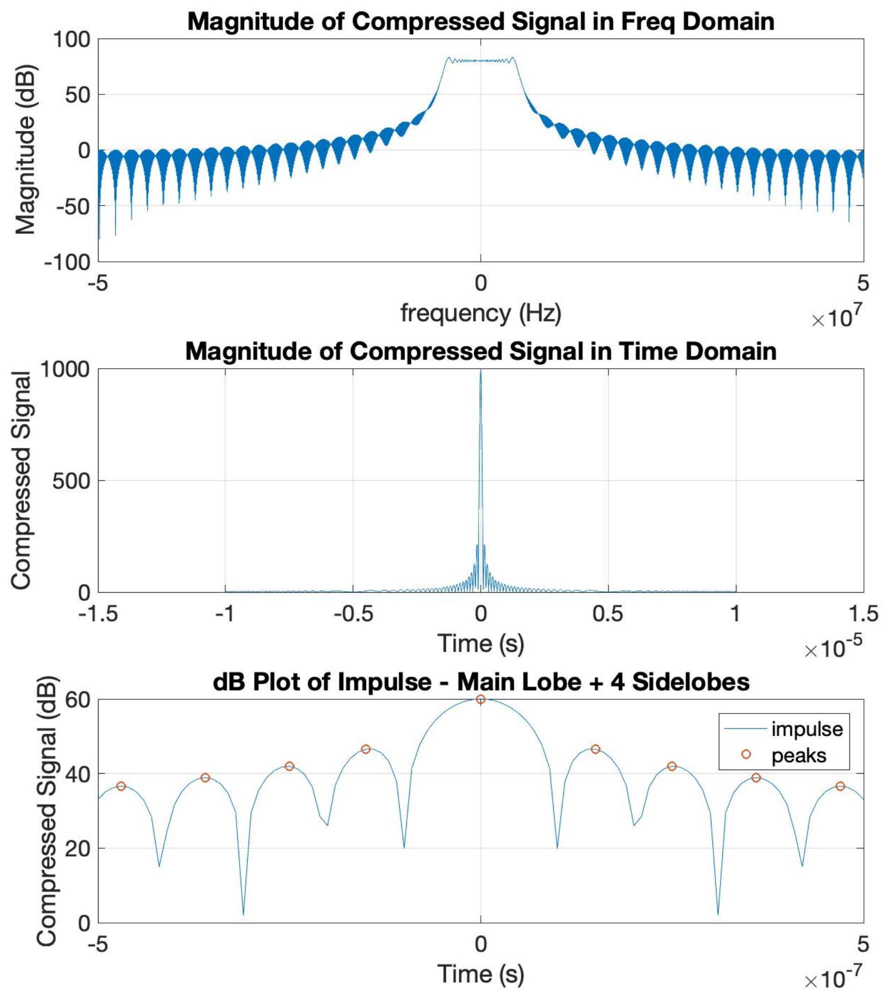
a) Plot the chirp spectrum in the form 20 log(|S(f)|).



The theoretical bandwidth is BW = $s\tau = 10^{12}$ Hz/s * 10^{-5}s = 10^{7}Hz = 10MHz. From the plot, we can see that the bandwidth is close to 10MHz, since the width of the spectrum at 3dB down from the peak is about -0.475MHz to 0.475MHz, or 9.5MHz width.

```
%% Question 1 - Part A
clc; clearvars; close all;
set(0,'defaultAxesFontSize',16)

% Create a reference chirp signal with the given parameters
num = 2048; % number of complex samples
s = 1e12; % Slope in (Hz/s)
tau = 1e-05; % Pulse length (s)
fs = 1e8; % Sample rate (Hz)
fc = 0; % Center Frequency

chirp = makechirp(s,tau,fs,fc,1,num);

chp_fft = fft(chirp); % transform to frequency domain

% calculate the dB spectrum of the signal
magnitude = 20*log10(abs(chp_fft) + 1e-30);

freq = -fs/2:fs/num:fs/2-(fs/num); % Frequency axis

t_ind = linspace(-1024,1023,num); % Time Axis
t = t_ind/fs;
% Time Steps to plot
ts1 = 975;
ts2 = 1075;

% Plot the signals and spectrum
figure(1);
subplot(3,1,1)
plot(real(chirp),'b'); title('Real Part of Chirp Signal');
grid on
xlim([1 num])
subplot(3,1,2)
plot(imag(chirp),'r'); title('Imaginary Part of Chirp Signal');
grid on
xlim([1 num])
subplot(3,1,3)
plot(freq,fftshift(magnitude)); title('dB Magnitude of Chirp Spectrum');
ylabel('20*log10(|chirp| + 1e^-30)'); xlabel('frequency (Hz)');
grid on
```

b) Compress the chirp with a perfect reference signal. Plot the main lobe of the response plus several sidelobes on a dB plot.

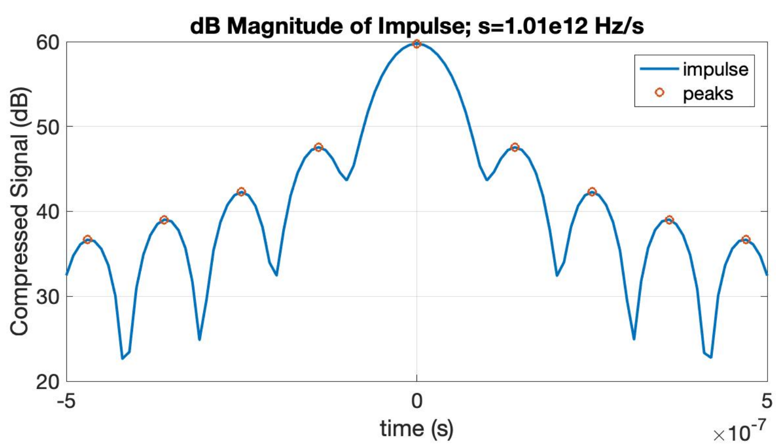


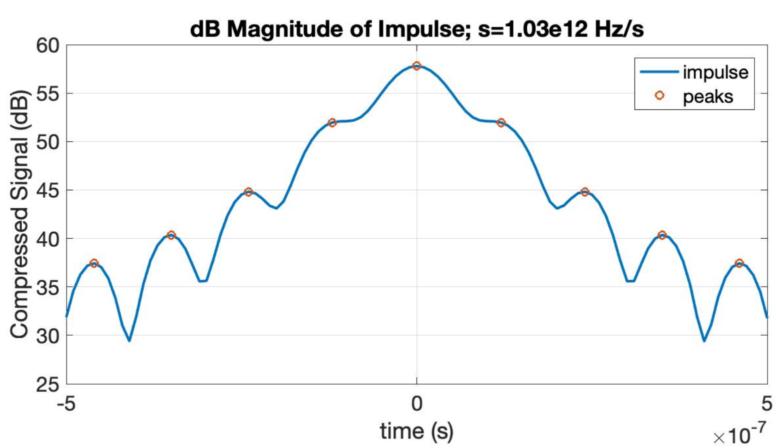
The peak ratio is -13.5dB.

```
%% Question 1 - Part B
% Compress
compress = chp_fft.*conj(chp_fft);
comp_mag = 20*log10(abs(compress) + 1e-30); % Get the dB magnitude of the
compressed chirp

signal_comp = ifft(compress);

mag_sig = fftshift(20*log10(abs(signal_comp) + 1e-30));
[val,ind] = findpeaks(mag_sig(ts1:ts2));

figure(2);
subplot(3,1,1);
plot(freq,fftshift(comp_mag));
title('Magnitude of Compressed Signal in Freq Domain');
xlabel('frequency (Hz)'); ylabel('Magnitude (dB)');
grid on
```

```
subplot(3,1,2);
plot(t,fftshift(abs(signal_comp))); title('Magnitude of Compressed Signal in
Time Domain');
grid on
xlabel('Time (s)'); ylabel('Compressed Signal');
subplot(3,1,3)
plot(t(ts1:ts2),mag_sig(ts1:ts2)); title('dB Plot of Impulse - Main Lobe + 4
Sidelobes');
hold on
plot((ind+t_ind(ts1-1))/fs,val,'o');
xlabel('Time (s)'); ylabel('Compressed Signal (dB)');
legend('impulse','peaks');
grid on

sorted_peaks = sort(val,'descend');
rel_height = sorted_peaks(1)-sorted_peaks(2);

disp(['Sidelobe Magnitude Difference for Perfect Reference = '
num2str(rel_height) ' dB']);
```

c) Repeat (b) for a reference chirp with s=1.01e12 and s=1.03e12





In the first plot (1% error giving s=1.01e12) the relative height of the main peak to the first side lobe is only 12.2dB. The SNR has effectively decreased.

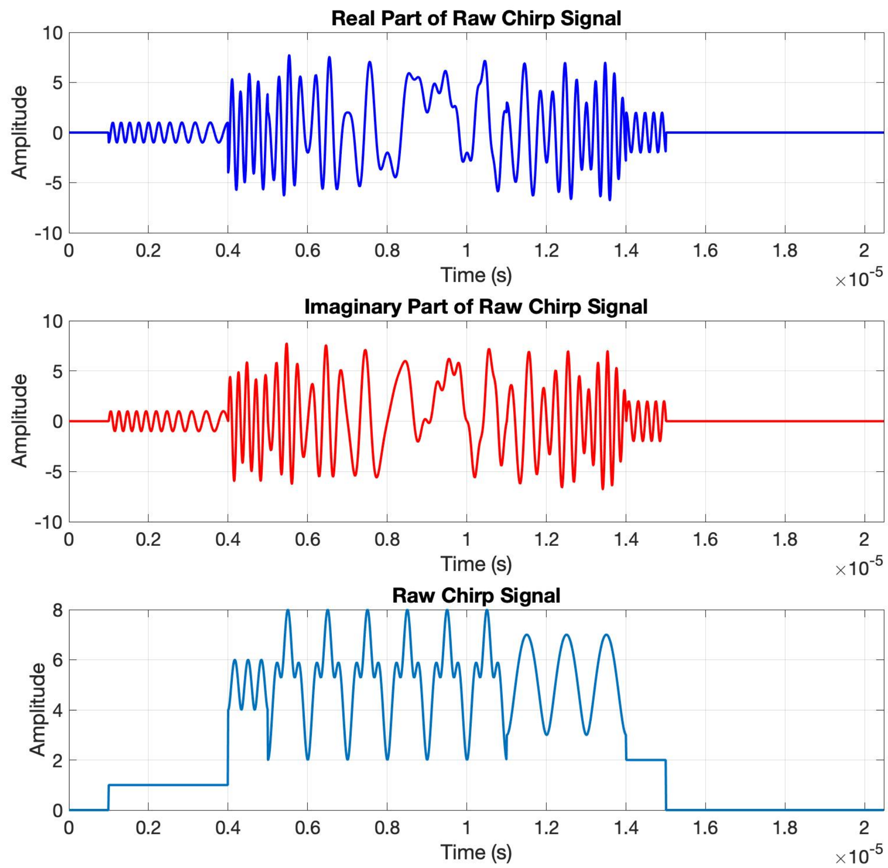
In the second plot (3% error giving s=1.03e12) the difference is -5.7dB, making an even further decrease in SNR.

```
%% Question 1 - Part C
% Create a reference chirp signal with the given paramters

s1 = 1.01*10^12; % Slope in (Hz/s)
s2 = 1.03*10^12;

str1 = '1.01e12';
str2 = '1.03e12';


chirp1 = makechirp(s1,tau,fs,fc,1,num); % Reference Chirp 1
chirp2 = makechirp(s2,tau,fs,fc,1,num); % Reference Chirp 2

% References in Frequency Domain
rfft = fft(chirp1); % transform to frequency domain
rfft_2 = fft(chirp2);

compress = chp_fft.*conj(rfft);
compress_2 = chp_fft.*conj(rfft_2);

signal_comp = ifft(compress);
signal_comp_2 = ifft(compress_2);

mag_sig_1 = fftshift(20*log10(abs(signal_comp) + 1e-30));
[val1,ind1] = findpeaks(mag_sig_1(ts1:ts2));

mag_sig_2 = fftshift(20*log10(abs(signal_comp_2) + 1e-30));
[val2,ind2] = findpeaks(mag_sig_2(ts1:ts2));
temp = mag_sig_2(ts1:ts2); % findpeaks didn't find the peak of the closest
side lobe
temp_ind = [ind1(1:3)+1 ind1(4)+2 ind1(5) ind1(6)-2 ind1(7:end)-1];
val3 = temp(temp_ind);

figure;
subplot(2,1,1)
plot(t(ts1:ts2),mag_sig_1(ts1:ts2),'LineWidth',2);
title(['dB Magnitude of Impulse; s=' str1 ' Hz/s']);
xlabel('time (s)'); ylabel('Compressed Signal (dB)');
hold on
plot((ind1+t_ind(ts1-1))/fs,val1,'o','LineWidth',1.5);
hold off
grid on
legend('impulse','peaks');
subplot(2,1,2)
plot(t(ts1:ts2),mag_sig_2(ts1:ts2),'LineWidth',2);
title(['dB Magnitude of Impulse; s=' str2 ' Hz/s']);
xlabel('time (s)'); ylabel('Compressed Signal (dB)');
hold on
plot((temp_ind+t_ind(ts1-1))/fs,val3,'o','LineWidth',1.5);
hold off
grid on
legend('impulse','peaks');
```

```
sorted_peaks1 = sort(val1,'descend');
rel_height1 = sorted_peaks1(1)-sorted_peaks1(2);

sorted_peaks2 = sort(val3,'descend');
rel_height2 = sorted_peaks2(1)-sorted_peaks2(2);

disp(['Sidelobe Magnitude Difference for s1 = ' num2str(rel_height1) ' dB']);
disp(['Sidelobe Magnitude Difference for s2 = ' num2str(rel_height2) ' dB']);
```

Question 2: Separating Multiple Chirps

a) Plot the raw signal amplitude vs. time and comment on the separability of the chirps.

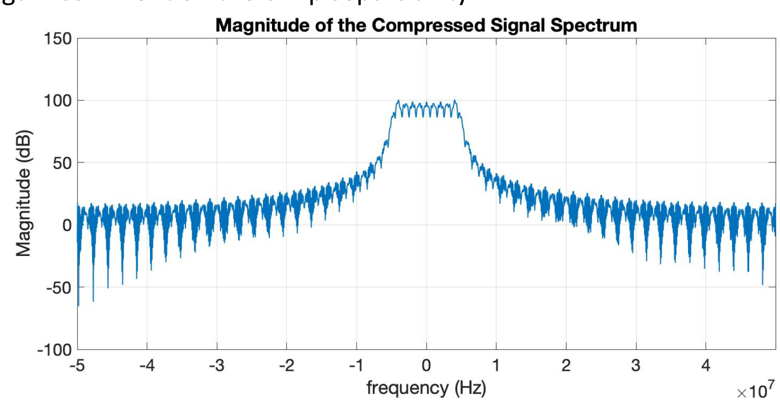


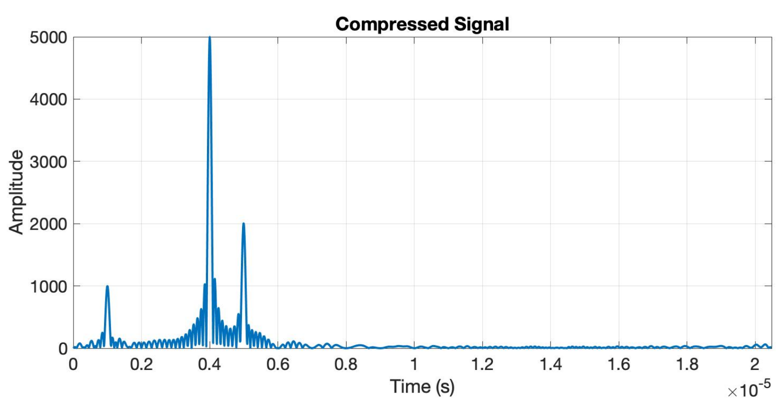
We can definitely tell that we do not have a standard, single chirp, but it would be very difficult to distinguish between the individual signals in the time domain because the signals overlap. Chirps starting at location 100, 400, and 500 will interfere with each other.

```
%% Question 2 - Part A
clc; clearvars; close all;

% Create a reference chirp signal with the given parameters
num = 2048; % number of complex samples
s = 1e12; % Slope in (Hz/s)
tau = 1e-05; % Pulse length (s)
fs = 1e8; % Sample rate (Hz)
fc = 0; % Center Frequency
dt = 1/fs;
```

```
t = linspace(0,num*dt,num);

chp1 = makechirp(s,tau,fs,fc,101,num);
chp2 = 5*makechirp(s,tau,fs,fc,401,num);
chp3 = 2*makechirp(s,tau,fs,fc,501,num);

chirp = chp1 + chp2 + chp3;

figure
subplot(3,1,1)
plot(t,real(chirp),'b','LineWidth',2);
title('Real Part of Raw Chirp Signal');
xlabel('Time (s)'); ylabel('Amplitude');
xlim([t(1) t(end)]);
grid on
subplot(3,1,2)
plot(t,imag(chirp),'r','LineWidth',2);
title('Imaginary Part of Raw Chirp Signal');
xlabel('Time (s)'); ylabel('Amplitude');
xlim([t(1) t(end)]);
grid on
subplot(3,1,3)
plot(t,abs(chirp),'LineWidth',2);
title('Raw Chirp Signal');
xlabel('Time (s)'); ylabel('Amplitude');
xlim([t(1) t(end)]);
grid on
```

b) Compress the signal by correlation with an ideal reference function. Plot the result on a linear scale and again comment on the chirp separability.



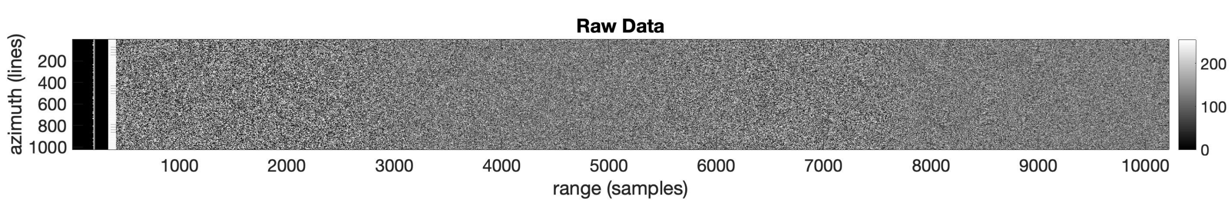


Now, we can distinguish between the chirps in the time domain. There is a peak at each chirp location (100, 400, and 500 samples_, scaled according to each chirp's amplitude (1, 5, and 2, respectively).

```
%% Question 2 - Part B
% Make the Reference Chirp
ref_chp = makechirp(s,tau,fs,fc,1,num);

% Transform chirps into frequency domain
chp_fft = fft(chirp);
ref_chp_fft = fft(ref_chp);

% Compress the signal
compress_fft = chp_fft.*conj(ref_chp_fft);
mag_comp = fftshift(20*log10(abs(compress_fft) + 1e-30));
signal_comp = ifft(compress_fft);

freq = -fs/2:fs/num:fs/2-(fs/num); % Frequency axis

figure
subplot(2,1,1)
plot(freq,mag_comp,'LineWidth',1);
title('Magnitude of the Compressed Signal Spectrum');
xlabel('frequency (Hz)'); ylabel('Magnitude (dB)');
xlim([freq(1) freq(end)]);
grid on
subplot(2,1,2)
plot(t,abs(signal_comp),'LineWidth',2);
title('Compressed Signal');
xlabel('Time (s)'); ylabel('Amplitude');
xlim([t(1) t(end)]);
grid on
```

Question 3: Actual Data

a) Display the byte file on your screen and observe the 412 bytes preceding the actual data.

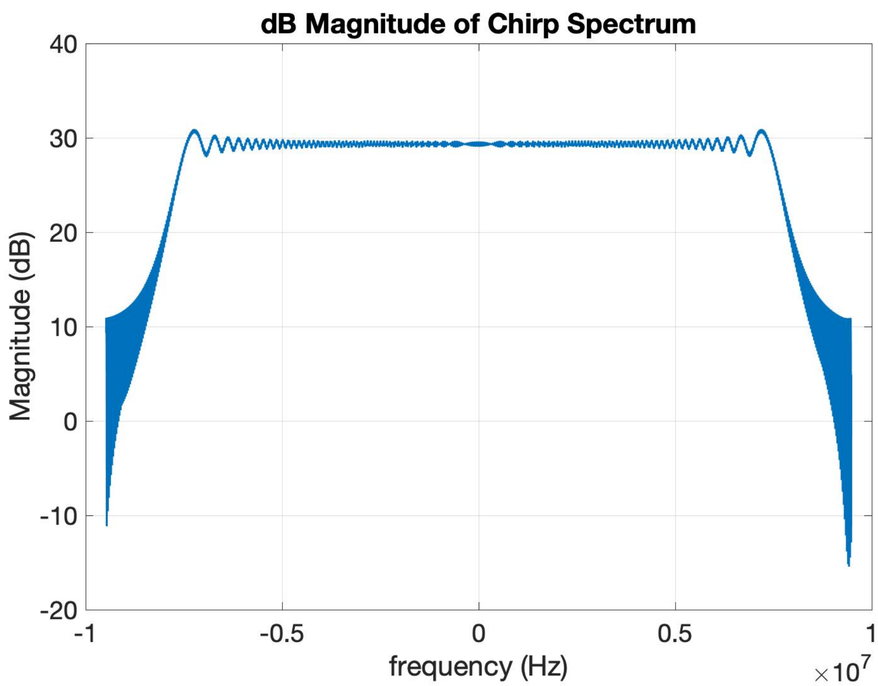


The raw data was multiplied by 8 to fill the colormap.

```
%% Question 3 - Part A
clc; clearvars; close all;
set(0,'DefaultAxesFontSize',20)

nhdr = 412;
nsamp = 10218;
nlines = 1024;

% Open the ERS data
fid = fopen('ersdata');
ERS = fread(fid,[nsamp nlines],'uint8');
fclose(fid);
dat = ERS;
```

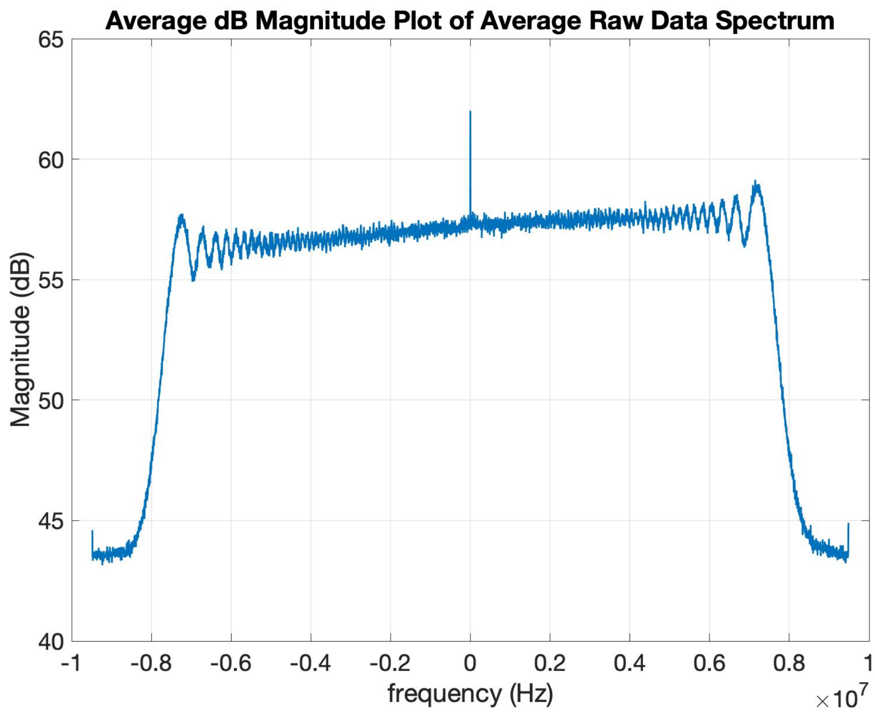
```
% adjust the data by multiplying by 8
dat(nhdr+1:end,:) = 8*ERS(nhdr+1:end,:);

figure
imagesc(dat'); % show the data
colormap('gray');
colorbar;
title('Raw Data'); xlabel('range (samples)'); ylabel('azimuth (lines)');
axis image
```

b) Plot the chirp spectrum with the given parameters.




```
%% Question 3 - Part B

% Parameters for reference chirp
num = (nsamp-nhdr)/2; % number of complex samples
s = 4.189166e11; % Slope in (Hz/s)
tau = 37.12e-6; % Pulse length (s)
fs = 18.96e6; % Sample rate (Hz)
fc = 0; % Center Frequency

% Make the chirp signal
chirp = makechirp(s,tau,fs,fc,1,num);

chp_fft = fft(chirp); % transform to frequency domain

% calculate the dB spectrum of the signal
magnitude = 20*log10(abs(chp_fft) + 1e-30);

freq = linspace(-fs/2,fs/2,num); % Frequency axis

figure
plot(freq,fftshift(magnitude),'LineWidth',1.5);
title('dB Magnitude of Chirp Spectrum');
xlabel('frequency (Hz)'); ylabel('Magnitude (dB)');
grid on
```

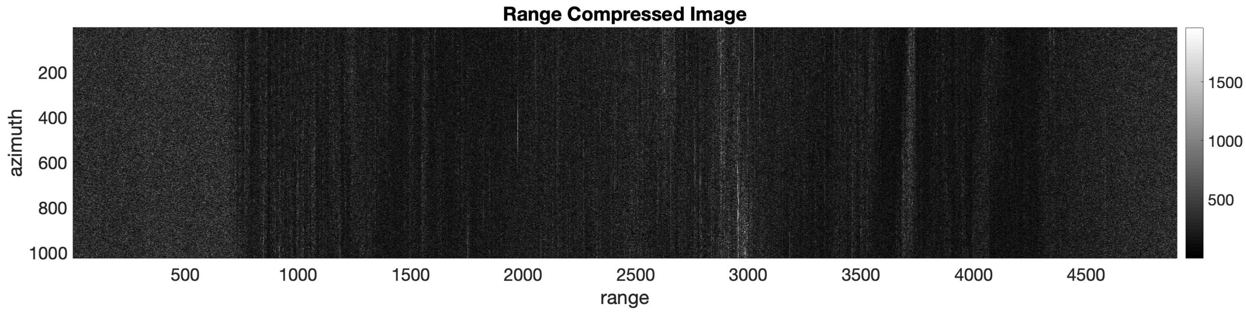
c) Average the 1024 spectra in the ersdata file, excluding the header data. Plot the result and compare to your plot from (b).



The average magnitude spectrum of the raw data is very similar to the reference chirp (same bandwidth). However, it is a slightly tilted version, due to the average value of the data not being exactly 15.5.

```
%% Question 3 - Part C
sig = ERS(nhdr+1:end,:); % Remove Header from Raw Data

sig_even = sig(2:2:end,:)-15.5; % Get the even component
sig_odd = sig(1:2:end,:)-15.5; % Get the odd component
signal = sig_odd + 1i*sig_even; % Make the complex number
clear sig_even sig_odd

% Get in Frequency Domain
signal_fft = fft(signal); % fft each azimuth line (column)

avg_signal = mean(abs(signal_fft),2); % Take the average of the spectra magnitudes
magnitude = 20*log10(avg_signal + 1e-30);

figure
plot(freq,fftshift(magnitude),'LineWidth',1.5);
title('Average dB Magnitude Plot of Average Raw Data Spectrum');
xlabel('frequency (Hz)'); ylabel('Magnitude (dB)');
grid on
```

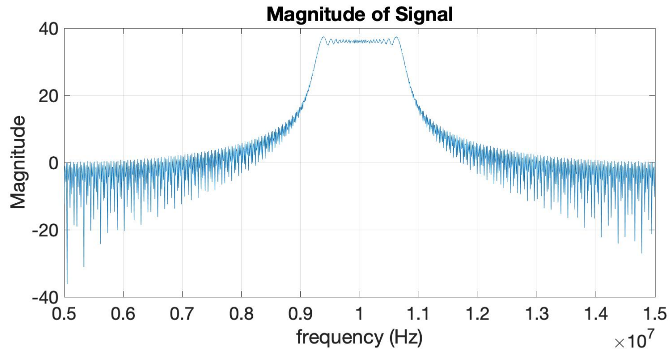
d) Compress each range line and create a range-compressed image.

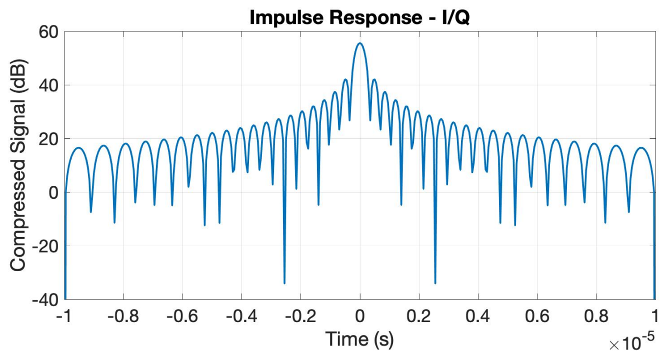



```
%% Question 3 - Part D

% Compress the signal with the reference chirp
compress_fft = zeros(size(signal_fft));
for k=1:nlines
   compress_fft(:,k) = signal_fft(:,k).*conj(chp_fft.');
end

% Convert back to time domain
sig_comp = ifft(compress_fft);

im = abs(sig_comp');

figure
imagesc(abs(sig_comp')); title('Range Compressed Image');
xlabel('range'); ylabel('azimuth');
colormap('gray');
colorbar
axis image
```

Question 4: I/Q and offset video processing

a) Assume I/Q system and that the chirp is centered at the carrier fc. Compress the chirp and plot the impulse response.

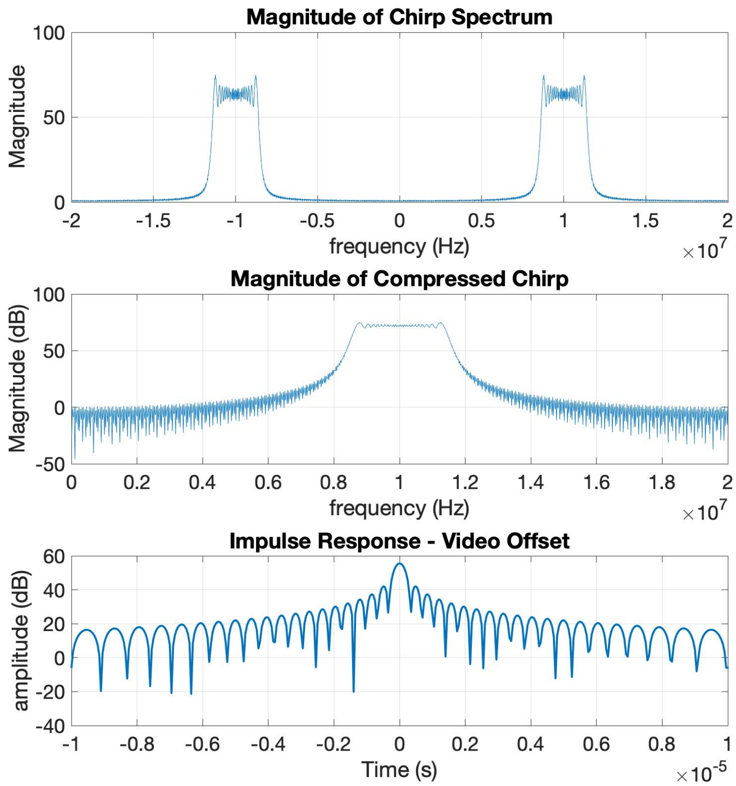





```
%% Question 4 - Part A
clc; clearvars; close all;

s = 1e11; % Slope in (Hz/s)
tau = 30e-6; % Pulse length (s)
fs = 20e6; % Sample rate (Hz)
fc = 10e6; % center frequency (Hz)
num = 1024;
dt = 1/fs;
t = (-num/2:num/2-1)*dt;

ref = makechirp(s,tau,fs,fc,1,num);
sig = makechirp(s,tau,fs,fc,1,num);

sig_fft = fft(sig); % transform to frequency domain
ref_fft = fft(ref);

compress_fft = sig_fft.*conj(ref_fft);

signal_comp = fftshift(ifft(compress_fft));
mag_sig = 20*log10(abs(signal_comp)+1e-30);

freq = linspace(fc-fc/2, fc+fc/2, num); % Frequency axis

figure
subplot(2,1,1)
plot(freq,20*log10(abs(sig_fft) + 1e-30)); title('Magnitude of Signal')
xlabel('frequency (Hz)'); ylabel('Magnitude');
grid on
subplot(2,1,2)
plot(t,mag_sig,'LineWidth',2); title('Impulse Response - I/Q');
xlabel('Time (s)'); ylabel('Compressed Signal (dB)');
```

```
xlim([-1 1]*1e-5); ylim([-40 60]);
grid on
```

b) Now, assume you are using an offset video system. What are the maximum and minimum chirp frequencies? Plot your impulse response. How similar is it to your answer in (a)?



The bandwidth is:
BW = $s\tau$ = (10^{11} Hz/s)(30 * 10^{-6} s) = 3 * 10^6 Hz = 3MHz

The minimum chirp frequency is:
fmin = fc - $\frac{BW}{2}$ = 10 MHz - $\frac{3\ MHz}{2}$ = 8.5 MHz

and the maximum chirp frequency is:
$f_{max} = f_c + \frac{BW}{2}$ = 10 MHz + $\frac{3\ MHz}{2}$ = 11.5 MHz

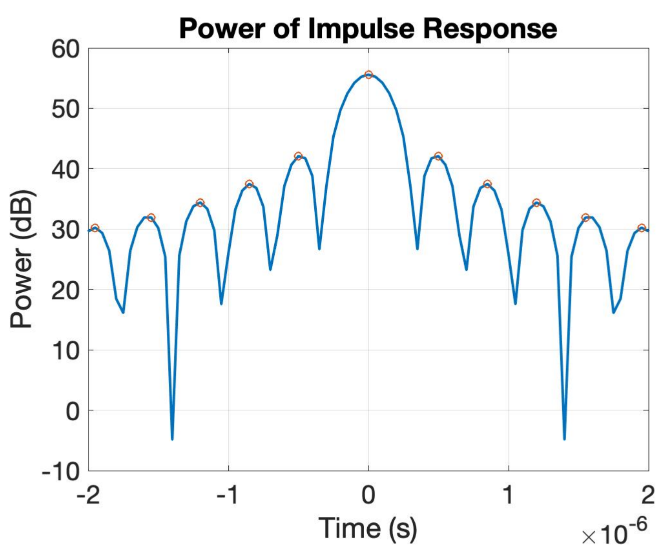
The impulse response of the offset video system should be identical to the impulse response for the I/Q system. Since we sampled at 2*fs, we have conjugate symmetry, and can simply take the positive frequencies of the spectrum for the signal compression. The result has slight numerical differences to the I/Q system at very small numbers.

```
%% Question 4 - Part B

fs2 = fs*2; %40e6; % New sample rate (Hz)

chirp2 = makechirp(s,tau,fs2,fc,1,num*2);

chp_fft_2 = fft(real(chirp2)); % transform to frequency domain
chp_fft_side = chp_fft_2(1:num); % Take only the positive frequencies (one sideband)
```

```
compress_fft = chp_fft_side.*conj(ref_fft);
mag_comp = 20*log10(abs(compress_fft) + 1e-30);

signal_comp = fftshift(ifft(compress_fft));
mag_sig2 = 20*log10(abs(signal_comp) + 1e-30);

freq = linspace(-fc-(fc), fc + fc, num*2); % Frequency axis
freq2 = freq(num + 1:end);

figure(11)
subplot(3,1,1)
plot(freq,abs(chp_fft_2)); title('Magnitude of Chirp Spectrum');
xlabel('frequency (Hz)'); ylabel('Magnitude');
grid on
subplot(3,1,2)
plot(freq2,mag_comp); title('Magnitude of Compressed Chirp');
xlabel('frequency (Hz)'); ylabel('Magnitude (dB)');
grid on
subplot(3,1,3)
plot(t,mag_sig2,'LineWidth',2); title('Impulse Response - Video Offset');
xlabel('Time (s)'); ylabel('amplitude (dB)');
ylim([-40 60]); xlim([-1 1]*1e-5);
grid on
```

Question 5: Sidelobe filtering

a) What are the peak and integrate sidelobe levels for the chirp in problem 4?

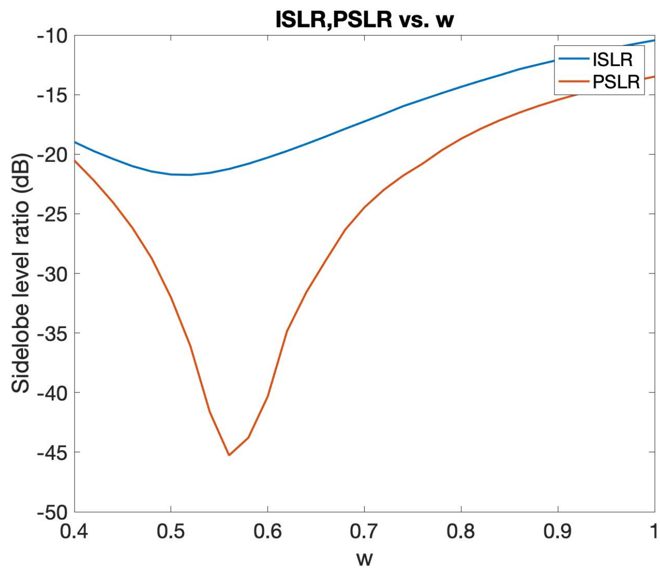


The Peak sidelobe ratio (PSLR) is -13.49, and the integrated sidelobe level ratio (ISLR) is -10.44 dB.

```
%% Question 5
clear;close all;clc
set(0,'defaultAxesFontSize',20)

s=1e11; % Chirp slope, Hz/s
tau=30e-6; % Pulse length, s
fs=20e6; % Complex sample rate, Hz
fc=10e6; % Center frequency (Carrier frequency)

Nsample=ceil(fs*tau); % Number of samples
fftlen=2^ceil(log(Nsample)/log(2));
dt=1/fs; % Sample spacing in time domain
df=fs/fftlen; % Sample spacing in frequency domain
f=(0:fftlen-1)*df;
t=(-fftlen/2:fftlen/2-1)*dt;

% construct ref chirp
ref_chirp=makechirp(s,tau,fs,fc,1,fftlen);
ref_fft=fft(ref_chirp);

% construct impulse response
impres=fftshift(ifft(ref_fft.*conj(ref_fft)));


% calculate PSLR and ISLR
impres_pw = abs(impres).^2;
[val,ind] = findpeaks(impres_pw);

[islr,pslr]=property_imp(impres_pw);

figure
plot(t,10*log10(abs(impres_pw)),'LineWidth',2)
hold on
plot(t(ind),10*log10(abs(val)),'o')
xlim([-20,20]*1e-7)
xlabel('Time (s)')
ylabel('Power (dB)')
grid on
title('Power of Impulse Response')

function [ISLR,PSLR]=property_imp(imprespw)
% this function computes ISLR and PSLR of a given impulse response (power)
% it also requires that the imprespw is symmetric about the peak
n = length(imprespw);
% first, find the position of the peak
[pkv,pkidx]=max(imprespw);
% then, find the first null point location
for m=pkidx+1:n-1
    if(imprespw(m)<imprespw(m-1) && ...
        imprespw(m)<imprespw(m+1))
        null_indx = m;
        break;
    end
end
% lastly, find the first sidelobe location
for m=null_indx+1:n-1
    if(imprespw(m)>imprespw(m-1) && ...
        imprespw(m)>imprespw(m+1))
        sidelobe_indx = m;
        break;
    end
end

% PSLR
PSLR = imprespw(sidelobe_indx)/imprespw(pkidx);
```

```
PSLR = 10*log10(PSLR);
% ISLR
sidelobepw = sum(imprespw(null_indx:n));
pkpw = sum(imprespw(pkidx:null_indx));
ISLR = sidelobepw/pkpw;
ISLR = 10*log10(ISLR);

return

end
```

b) Weight the chirp. Let w vary from 0.4 to 1. Plot the ISLR and PSLR as a function of w



The PSLR is minimized at w = 0.56, which is consistent with the equation for the Hamming window (w = 0.54) that best surpresses the amplitude of the first sidelobe. The ISLR is minimized at w = 0.52, which is consistent with the equation for the Hann window that minimizes total sidelobe energy.

```
%% impulse response weighting
ww=0.4:0.02:1; % weighting
bw=s*tau; % bandwidth

figure(3)
for k=1:length(ww)
    w=ww(k);
    weight=w+(1-w)*cos(2*pi/bw.*(f-fc));

    spect_newref=ref_fft.*weight;

    newimpres=fftshift(ifft(ref_fft.*conj(spect_newref)));
    newimprespw=abs(newimpres).^2;

    [islr(k),pslr(k)]=property_imp(newimprespw);


    hold on
    plot(t,10*log10(abs(newimprespw)),'LineWidth',1)
    hold off
end
ylim([0 60])
xlim([-0.2 0.2]*1e-5)
```

```
figure
plot(ww,islr,'linewidth',2)
hold on
plot(ww,pslr,'linewidth',2)
xlabel('w')
ylabel('Sidelobe level ratio (dB)')
title('ISLR,PSLR vs. w')
legend('ISLR','PSLR')
saveas(gcf,'pb5_PSLR_ISLR','tiff')

[~,idx1]=min(islr);
ww(idx1)
[~,idx2]=min(pslr);
ww(idx2)
```