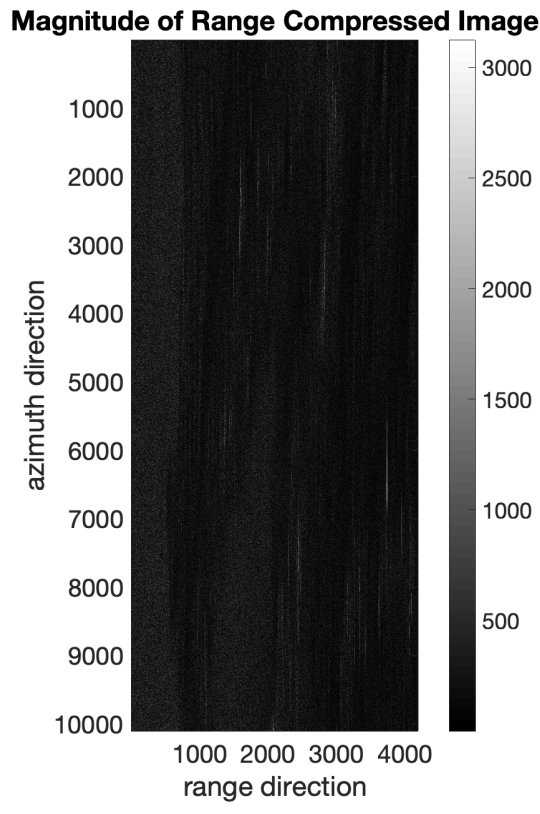
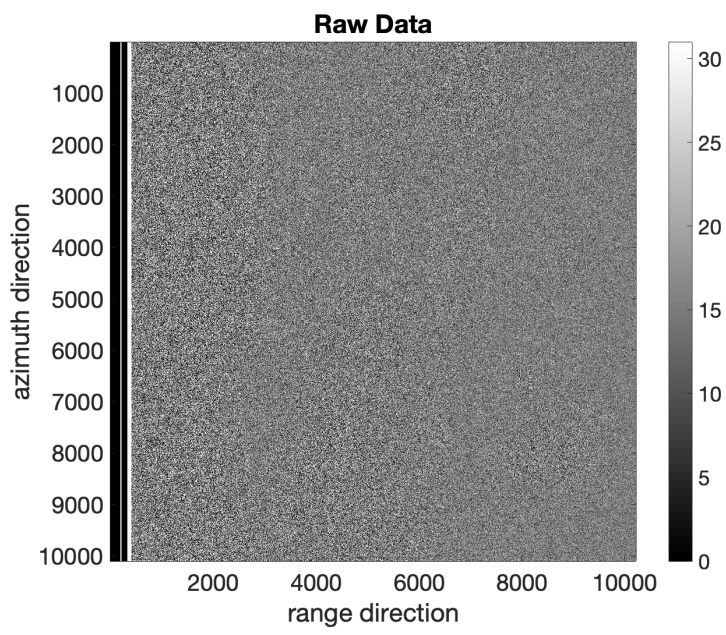


Geophysics 265: Imaging Radar  
Homework #3 Solutions

Before we do any azimuth processing, we need to read in the ERS data and do the range compression:



As in homework 2, we create the reference chirp and then transform each azimuth line into the frequency domain, and multiply it by the conjugate of the fft of the reference chirp (i.e. correlate each azimuth line with the reference chirp). Once we have the range compressed image, we keep only the valid range bins: The original ERS data in complex form has 4903 range bins, and the reference chirp has 703 samples, so we keep  $4903-703=4200$  range bins in the range compressed image. We will learn more about why this is the case in class.

```

%% Read in Raw Data and Range Compress
clc; clearvars; close all;

range = 10218;
lines = 10100;
nheader = 412;

% Open the file
fid = fopen('ersdata.hw3');
ERS = fread(fid,[range lines],'uint8');

% Display the raw data
figure
imagesc(ERS'); % show the data
set(gca,'fontsize',20)
colormap('gray');
caxis([0 31]); % scale the colorbar to 5-bit
colorbar;
title('Raw Data'); xlabel('range direction'); ylabel('azimuth direction');
axis image

% Create the reference chirp signal
num = (range-nheader)/2; % number of complex samples we want
s = 4.189166*10^11; % Slope in (Hz/s)
tau = 37.12*10^-6; % Pulse length (s)
fs = 18.96*10^6; % Sample rate (Hz)
prf = 1679.9;

chp = makechirp(s,tau,fs,0,1,num);
chp_fft = fft(chp); % transform to frequency domain

% Get the raw data in the right format
sig = ERS(nheader+1:end,:); % Extract the non-header columns

sig_even = sig(2:2:end,:)-15.5; % Get complex component
sig_odd = sig(1:2:end,:)-15.5; % Get real component
signal = sig_odd + 1i*sig_even; % Combine real and complex components
clear sig_even sig_odd

% Fourier Transform in range
signal_fft = fft(signal);

% Range compression with reference chirp
compress_fft = zeros(size(signal_fft));
for k=1:lines
    compress_fft(:,k) = signal_fft(:,k).*conj(chp_fft. ');
end

% Inverse Fourier Transform to get back to signal domain
sig_comp = ifft(compress_fft);

% Chop off the invalid pixels in range
nvalid = num-floor(tau*fs);
sig_comp_valid = sig_comp(1:nvalid,:);

```

```

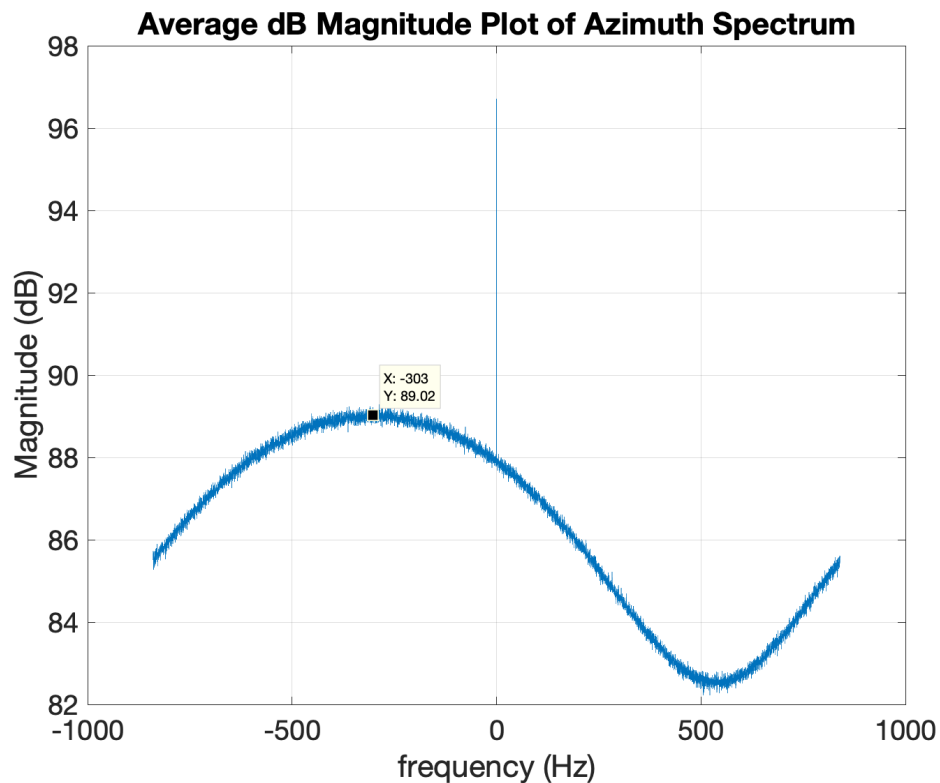
figure
imagesc(abs(sig_comp_valid'));
set(gca,'fontsize',20)
title('Magnitude of Range Compressed Image');
xlabel('range direction'); ylabel('azimuth direction');
colormap('gray');
colorbar
axis image

%%
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: location 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: Azimuth Spectrum

a) Plot the azimuth spectrum of the data averaged over all valid range bins.



**b) Determine the Doppler centroid using the spectrum in (a) and also the average phase change method.**

- From eye-balling the spectrum, the Doppler centroid appears to be at about -303 Hz, but it is difficult to know what the actual peak is in the noisy spectrum.
- Following the average phase change method, I determined that the Doppler centroid is at -300 Hz.

Compute the phase shift  $\Delta\phi(ir)$  at each range bin  $ir$  from line  $i_{az}$  to line  $i_{az} - 1$  in azimuth (where  $R$  is the range-compressed image):

$$P(ir) = \sum_{iaz=2}^{Naz} R(ir, iaz)R^*(ir, iaz - 1)$$

$$\Delta\phi(ir) = \tan^{-1} \frac{Im(P(ir))}{Re(P(ir))}$$

Then average  $\Delta\phi(ir)$  over all range bins and relate the phase to frequency, which will then be the Doppler centroid  $f_d$ .

$$\Delta\phi = \frac{\sum_{ir=1}^{Nr} \Delta\phi(ir)}{Nr}$$

$$\frac{\Delta\phi}{2\pi} = \frac{f_d}{prf}$$

$$f_d = prf \frac{\Delta\phi}{2\pi}$$

```
%% Question 1 Part A & B
% Plot the azimuth spectrum of the data averaged over all valid range bins

azimuth_fft = zeros(nvalid,lines);
for k=1:nvalid
    azimuth_fft(k,:) = fft(sig_comp_valid(k,:));
end

% Calculate the dB magnitude
magnitude_azimuth = 20*log10(abs(azimuth_fft) + 1*10^-30);
mag_az = mean(magnitude_azimuth,1);
freq = linspace(-prf/2,prf/2,lines);

figure
plot(freq,fftshift(mag_az));
set(gca,'fontsize',20)
title('Average dB Magnitude Plot of Azimuth Spectrum');
xlabel('frequency (Hz)'); ylabel('Magnitude (dB)');
grid on

% Average Phase Change Calculation

for k=1:nvalid
    bin=sig_comp_valid(k,2:end).*conj(sig_comp_valid(k,1:end-1));
```

```

    result(k)=sum(bin);
    phase(k)=atan(imag(result(k))/real(result(k)));
end

avgphase=mean(phase);
doppler_freq = avgphase/2/pi*prf;

```

## Question 2: An Unfocused SAR processor

- a) Calculate the parameters of an unfocused processor for the radar in (1). Evaluate the azimuth resolution, pulse spacing at the spacecraft, and output pixel spacing. How long would each burst have to be? How often need we repeat if the antenna is 10m long?

Azimuth Resolution:

$$\delta_{az} = \sqrt{\lambda r} = \sqrt{(0.0566 \text{ m})(830000 \text{ m})} = 216.74 \text{ m}$$

The pulse spacing in azimuth:

$$\Delta p = \frac{v}{prf} = \frac{750 \text{ m/s}}{1679.9 \text{ Hz}} = 4.494 \text{ m}$$

The minimum number of pulses:

$$nump_{min} = \frac{\delta_{az}}{\Delta p} = \left\lceil \frac{216.74 \text{ m}}{4.494 \text{ m}} \right\rceil = 49 \text{ pulses}$$

Rounding up to the next highest power of 2, we use  $np = 64$  pulses in a single burst.

The frequency resolution is:

$$\delta f = \frac{prf}{np} = \frac{1679.9 \text{ Hz}}{64 \text{ pulses}} = 26.25 \text{ Hz}$$

The output pixel spacing is then:

$$\delta x = \frac{(\delta f)\lambda r_0}{2v} = \frac{(26.25 \text{ Hz})(0.0566 \text{ m})(830000 \text{ m})}{(2)(7550 \frac{\text{m}}{\text{s}})} = 81.66 \text{ m}$$

The burst length is:

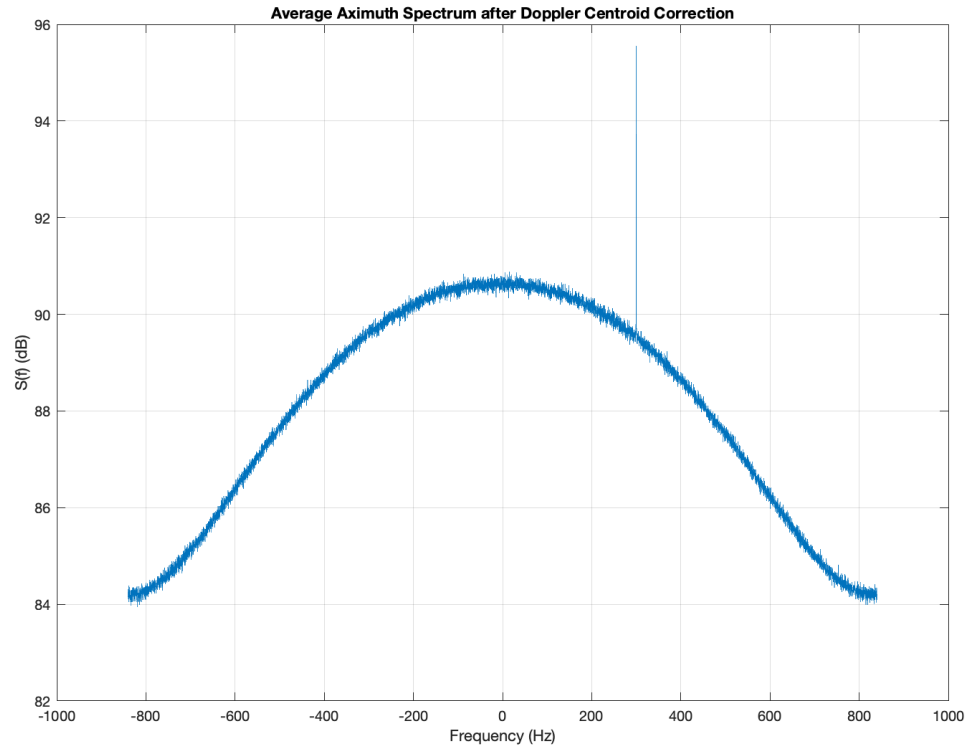
$$\delta t = \frac{np}{prf} = \frac{64 \text{ pulses}}{1679.9 \text{ Hz}} = 0.038 \text{ s}$$

The repeat cycle time for a 10 m long antenna is:

$$t_{cycle} = \frac{r_0 \lambda}{lv} = \frac{(830000 \text{ m})(0.0566 \text{ m})}{(10 \text{ m})(7550 \frac{\text{m}}{\text{s}})} = 0.622 \text{ s}$$

- b) Calculate the single-patch unfocused image from the first few lines of the file. Be sure and correct for the Doppler Centroid, and display the result with zeros displacement from the boresight at the center of the patch.

To process in azimuth, we need to make sure that we have corrected for the Doppler Centroid accurately. We can do the steering for the entire image and look at the average spectrum again. This time, our highest amplitude should be centered at 0 Hz.



```
%% Question 2
% Part A
lambda = 0.0566; %wavelength (m)
r0 = 830000; % range (m)
vel = 7550; %velocity (m/s)

w=1;
l = 10; % Antenna length
ground_extent = r0*lambda/l; %ground extent with 10m antenna
cycle_time = ground_extent/vel; %total cycle time (s)

% In order to know the number of looks to take, we need to know the bin
% spacing in range (m/pixel)
lk = 25.973;
dr=2.99792458e8/2/fs/sind(lk); % Based on the sample rate (m)

delta_az = sqrt(lambda*r0); % Azimuth resolution
dp = vel/prf; % pulse spacing (m/pulse)
burst = round(delta_az/dp); % minimum burst size (in number of pulses (pixels))
burst = pow2(ceil(log2(burst))); % rounded burst size to next power of 2 (transform
length)
df = prf/burst; % Frequency resolution (Hz)
dx = df*lambda*r0/2/vel; % output pixel spacing (m)
dt = burst/prf; % burst length (s)
dpatch = dp*burst/dx; % Offset between patches

nlk = floor(dx/dr); % The number of looks we need to take in range to get aquare
pixels at the end
```

```

npatch = floor(lines/burst); % number of patches we will process
Naz = floor(burst + (npatch - 1)*dpatch);

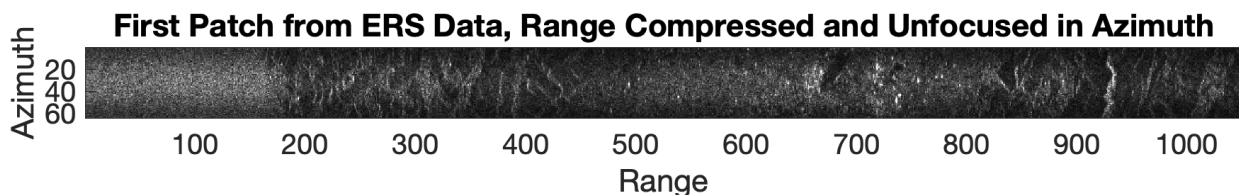
% Doppler Centroid Correction
rcompfd=zeros(nvalid,lines);
for k=1:lines
    az=sig_comp_valid(:,k)*exp(-1i*2*pi*doppler_freq*k/prf);
    rcompfd(:,k)=az;
end

% Check if we have done the doppler shift correctly ( The image should have
% our peak at zero frequency
spect_az = fftshift(fft(rcompfd,[],2));
spect_averaz = mean(abs(spect_az));
spectdb_az = 20*log10(abs(spect_averaz));

figure
set(gca,'fontsize',20)
plot(freq,spectdb_az)
xlabel('Frequency (Hz)')
ylabel('S(f) (dB)')
title('Average Azimuth Spectrum after Doppler Centroid Correction')
grid on

```

Now, we can process the first 64 pulses (first patch) and see how the single patch looks. To get square pixels, we can also “multilook” in range by 4 (i.e. average the amplitudes of 4 adjacent range pixels, effectively reducing the number of range bins by 4).



We can clearly see the antenna pattern in this first patch, where the azimuth edges are darker than the center of the beam.

```

%%
% Do the test image first
patch = rcompfd(:,1:burst);
data = abs(fftshift(fft(patch,[],2),2));

imagetest = zeros(floor(nvalid/nlk),burst);

for k=1:floor(nvalid/4)
    imagetest(k,:) = sum(data(1 + (k-1)*nlk:k*nlk,:));
end

figure
imagesc(imagetest')
set(gca,'fontsize',20)
colormap('gray')
set(gca,'fontsize',20)
title('First Patch from ERS Data, Range Compressed and Unfocused in Azimuth')
xlabel('Range')
ylabel('Azimuth')
axis image
cax = caxis;
caxis(0.4*cax)

```

c) **Determine the patch to patch spacing if we want to process all of the data (in pixels)**

$$\Delta_{patch} = \frac{(\Delta p)(np)}{\delta x} = \frac{(4.494 \text{ m})(64 \text{ pulses})}{(81.66 \text{ m})} = 3.522 \text{ pixels}$$

The patch spacing is not a whole number, so we alternate between taking 3 and 4 pixels for the patch-to-patch spacing when computing the final image in part (d).

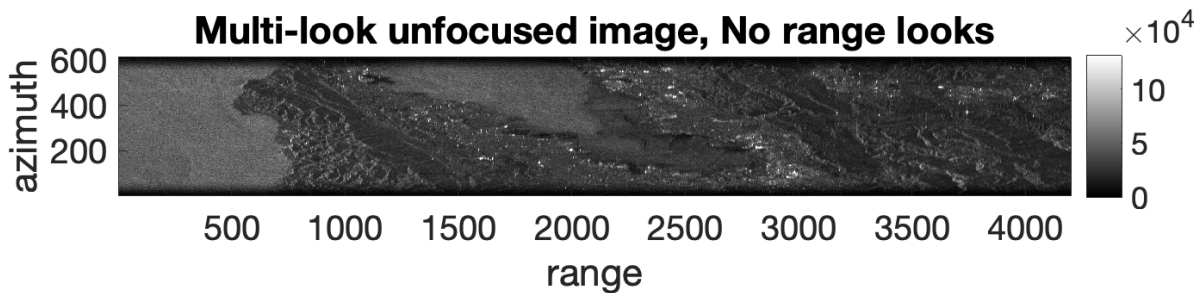
d) **Write a multi-looked unfocused processor and create an image from the full ersdata.hw3 data set. Can you begin to identify parts of the image?**

First compute the number of patches given the amount of data:

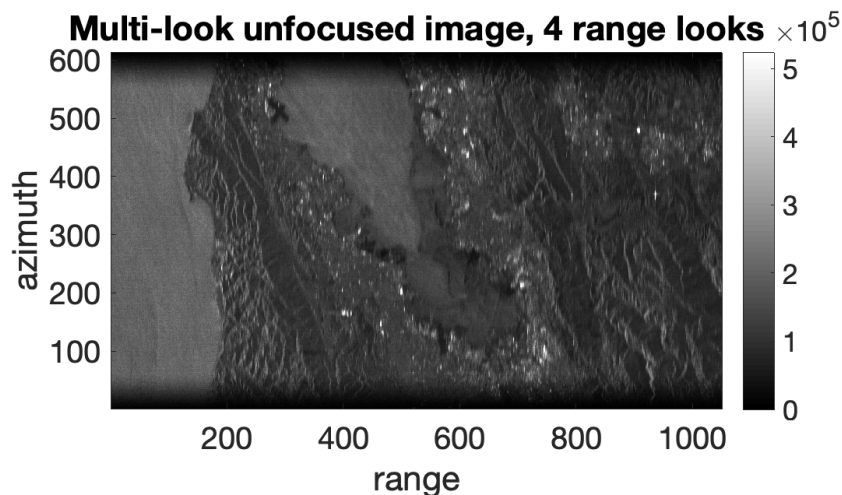
$$n_{patches} = \left\lfloor \frac{n_{az}}{np} \right\rfloor = \frac{10100 \text{ az lines}}{64 \text{ pulses}} = 157 \text{ patches}$$

Then compute the number of azimuth pixels in the multi-looked image, to preallocate space:

$$N_{az,looks} = np + (n_{patches} - 1)\Delta_{patch} = 613 \text{ samples}$$



If we take 4 looks in range, we get approximately square pixels and the image is much more recognizable. It's the Bay Area!





```

%% Now, do the unfocused processing for the entire image

unfcs = zeros(nvalid,Naz); % preallocate the final image
unfcs(:,1:burst) = data;
shift = zeros(npatch,1);
for k = 2:npatch
    disp(['Processing patch ' num2str(k)])
    patch = rcompfd(:,1 + (k-1)*burst:k*burst);
    data = abs(fftshift(fft(patch,[],2),2));
    shift = round((k-1)*dpatch);
    unfcs(:,shift:shift+burst - 1) = unfcs(:,shift:shift+burst - 1) ...
        + data;
end

figure
imagesc(unfcs')
set(gca,'fontsize',20)
colormap(gray)
colorbar
title('Multi-look unfocused image, No range looks');
xlabel('range')
ylabel('azimuth')
set(gca,'YDir','normal')
cax=caxis;
caxis(0.2*cax)
axis image
set(gca,'YDir','normal')
%% Multilook in range
imagefinal = zeros(floor(nvalid/nlk),Naz);
for k = 1:floor(nvalid/nlk)
    imagefinal(k,:) = sum(unfcs(1+(k-1)*nlk:k*nlk,:));
end

figure
imagesc(imagefinal')
set(gca,'fontsize',20)
colormap(gray)
colorbar
title('Multi-look unfocused image, 4 range looks');
xlabel('range')
ylabel('azimuth')
set(gca,'YDir','normal')
cax=caxis;
caxis(0.3*cax)
axis image
set(gca,'YDir','normal')

```