Luleå University of Technology

F70013R Radar for space and atmosphere research 7.5 ECTS

Assignment: ESRAD radar data analysis

By

Maitreya Ranade

Mini Gupta

Vanshika Kansal

Kiruna

Space Master Program

May 06, 2017

TASK – 1

a) Choose the date by signing your names in the data list. The data list is available on the reception. Make plots of signal-to noise ratio (SNR) as a function of universal time (UT) and altitude for each pulse length. Use Matlab commands “pcolor”, “shading flat”, “colormap jet”. Matlab codes should be enclosed.

Matlab code is given below. Click here!

Date of acquired data: 11 December 2006

SNR plots:

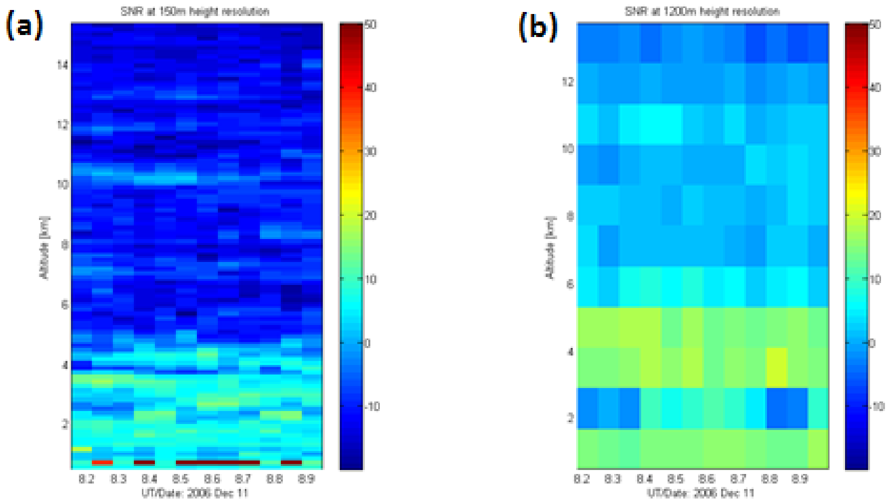

Figure 1: Plots of SNR as a function of UT and altitude for each pulse length (a) at 150m height resolution (b) 1200m height resolution.

b) Calculate pulse lengths, inter-pulse periods, pulse repetition frequency and maximum unambiguous range for these height resolutions.

Definitions:

$\Delta R = Resolution$,

R = Maximum Unambiguous range,

Ipp = Inter pulse period,

Tp = Pulse length,

D = Duty cycle (5%).

Formulae:

$\Delta R = \frac{C*Tp}{2}$ $\quad R = \frac{C*Ipp}{2}$ $\quad$ Ipp= $\frac{Tp}{D}$

Case 1. For $\Delta R = 150\ m$.

Tp = 1 microsec , Ipp = 20 microsec ,

$R = \frac{C*Ipp}{2}$ = 3 km.

$PRF = \frac{1}{Ipp}$ = 50 kHz

Case 2. For $\Delta R = 1200\ m$.

Tp = 8 microsec , Ipp = 160 microsec ,
$R = \frac{C*Ipp}{2}$ = 24 km.
$PRF = \frac{1}{Ipp}$ = 6.25 kHz

c) Derive the equation that shows relationship between the transmitted pulse length and the strength of the received signal. Support your theoretical results by the ESRAD experimental data.

Let the transmitted signal be represented by x(t). Therefore, power of the transmitted signal be

$$P_S = \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} (|x(t)|)^2 dt$$

If TP is the transmitted pulse length then x(t) will be zero outside the interval TP.

$$\Rightarrow P_S = \begin{cases} \lim_{T\to\infty} \frac{1}{T} \int_{-\frac{T}{2}}^{\frac{T}{2}} (|x(t)|)^2 dt\,, & t \geq T_P \\ 0, & t > T_P \end{cases}$$

Let Pr be the reflected power,
PS be the transmitted power,
R be the slant range,
and, σ be the radar cross section.

If we assume that the electromagnetic waves propagate without undergoing any dispersion and high frequency energy is emitted by an isotropic radiator, then the Non-directional Power Density SU is given by

$$S_u = \frac{P_s}{4.\pi.R_1^2} \left[\frac{W}{m^2}\right] \quad -(1)$$

However, if the radiated power is redistributed to provide more radiation in one direction, then the directional power density is given by

$$S_g = S_u.G \quad -(2)$$

where G is the antenna gain.

In reality, radar antennas have a beam width and an antenna gain. The target detection is also dependent on radar cross section since it determines how much power is reflected back in the direction of the radar and on the power density at the target position.

Therefore, the reflected power is given by

$$P_r = \frac{P_s}{4.\pi.R_1^2}.G.\sigma \quad [W] \qquad -(3)$$

When a target is considered as a radiator of the reflected power, the reflected power becomes equal to the transmitted power.

Since Pr depends on P_s and P_s depends on TP, so this becomes quite clear that the received power is dependent on TP. Higher TP means transmitted signal has a higher energy.

From the above two parts, it can be observed that as we increase the length of our transmitted pulse from 1 µsec to 8 µsec, SNR increases approximately by 20 dBs.

d) Identify the atmospheric parameters that might determine SNR variations on your plots.

Dispersion, humidity, temperature, pressure and electron density are some of the atmospheric parameters that might determine SNR variations on the plot. Also, since the atmospheric parameters vary significantly within a few 10s and the data is taken every 60 seconds, so it might result in variations of SNR due to changes in atmospheric properties.

TASK – 2

a) Make plots of signal-to noise ratio (SNR) as a function of universal time (UT) and altitude for each dataset. Use Matlab commands "pcolor", "shading flat", "colormap jet". Matlab codes should be enclosed.

Matlab code is given below: Click here!

Date of acquired data: 11 December 2006

Plots:

A.SNR plot of (i) Baker coding, (ii) Complementary coding, (iii) Uncoded data.

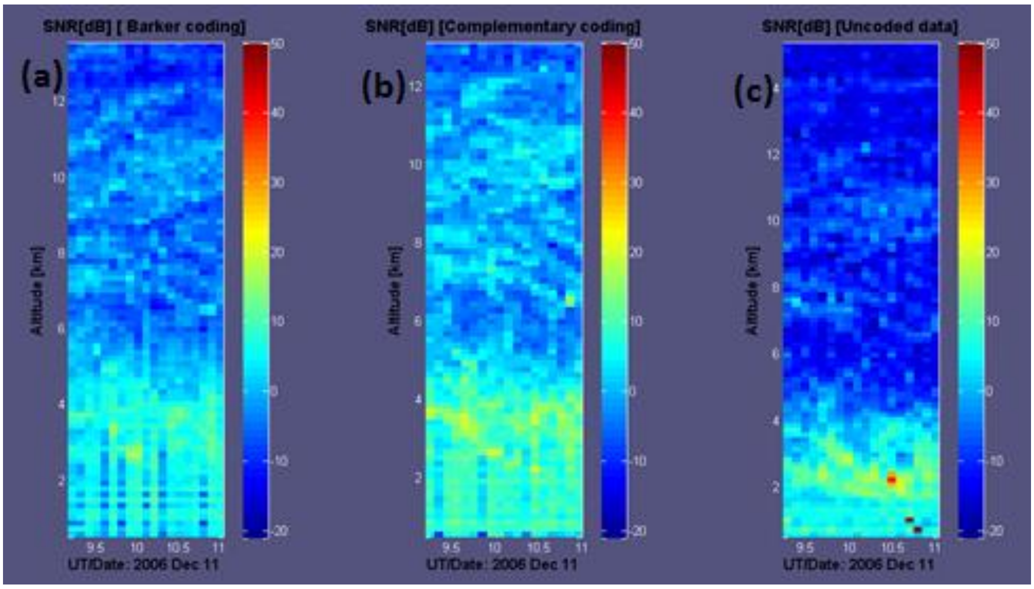

Figure 2.1: Plots of SNR as a function of UT and altitude for each pulse length (a) Barker coding (b) Complementary coding, & (c) Uncoded data.

b) Analyse and discuss shortly the quality of the obtained datasets in terms of their usefulness for scientific observations.

As it can be clearly seen from the results, the SNR observed is highest in Complementary coding and lowest in Uncoded data. Whereas, Barker coding stays moderate but has a significant high SNR than uncoded data. Hence, it can be inferred that coding data leads to increase in noise immunity. As strength of the signal is constant in all the data, it can be said that only decrease in noise has resulted in the change of SNR.

The data obtained from (i) Complementary coding, (ii) Barker coding and (iii) Uncoded data will have decreasing quality due to the noise impact respectively. Hence, the coding methods (Preferably complementary) will be used when we need highest noise immunity. For short distance object detection or applications where there will be no need of focus on detail, uncoded data or baker code can be used.

c) Estimate values and directions of the horizontal winds.

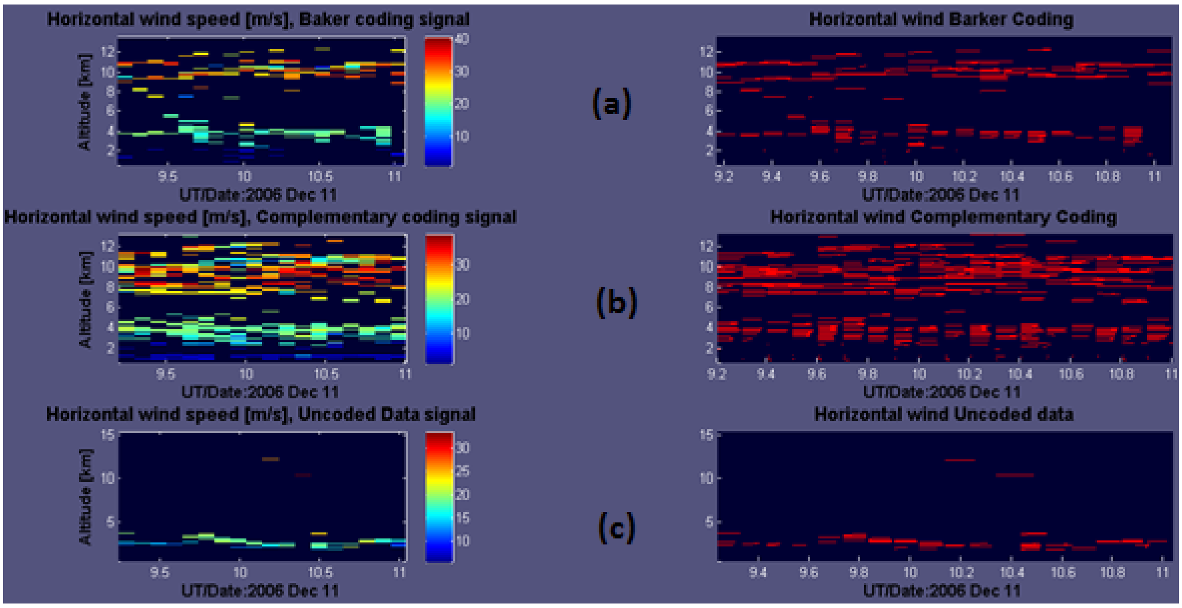

Figure 2.2: Plots of horizontal wind speed and directions for (a) Barker coding (b) Complementary coding, & (c) Uncoded data.

The direction of the horizontal wind is from west to east as can be seen from the MATLAB figures and at an altitude of 10 km, its value is around 30km/s.

Task 1 - MATLAB code

```
load TXT_20061211_test1.fca;
load TXT_20061211_test2.fca;
test_1 = TXT_20061211_test1;
test_2 = TXT_20061211_test2;

date = ['2006 Dec 11'];

% SNR at 150m height resolution

Time_1 = unique(test_1(:,1));
Altitude_1 = unique(test_1(:,2));
Size_Time_1 = size(Time_1,1);
Size_Altitude_1 = size(Altitude_1,1);


%SNR Ratio in dB

for i= 1:Size_Altitude_1
   for j= 1: Size_Time_1
      SNR(i,j) = test_1(i+((j-1)*Size_Altitude_1),4);
   end
end

% SNR at 1200m height resolution

Time_2 = unique(test_2(:,1));
Altitude_2 = unique(test_2(:,2));
Size_Time_2 = size(Time_2,1);
Size_Altitude_2 = size(Altitude_2,1);


%SNR Ratio in dB

for i= 1:Size_Altitude_2
   for j= 1: Size_Time_2
      SNR2(i,j) = test_2(i+((j-1)*Size_Altitude_2),4);
   end
end

%Setting limits of colorbar
bottom = min(min(min(SNR)),min(min(SNR2)));
top  = max(max(max(SNR)),max(max(SNR2)));

%Plotting
FigHandle = figure();
```

```
hold on
subplot ( 1 , 2 , 1 ) ;
colormap(jet);
pcolor(Time_1,Altitude_1,SNR);
shading flat;
% This sets the limits of the colorbar to manual for the first plot
caxis manual
caxis([bottom top]);
ylabel('Altitude [km]');
xlabel(['UT/Date: ',date ]);
title('SNR at 150m height resolution');
set(FigHandle, 'Position', [100, 0, 600, 800]);

c = colorbar ;
c.Label.String = 'SNR[dB]';

%Plotting
hold on
subplot ( 1 , 2 , 2 ) ;
colormap(jet);
pcolor(Time_2,Altitude_2,SNR2);
shading flat;
% This sets the limits of the colorbar to manual for the first plot
caxis manual
caxis([bottom top]);
ylabel('Altitude [km]');
xlabel(['UT/Date: ',date ]);
title( 'SNR at 1200m height resolution');
c = colorbar ;
c.Label.String = 'SNR[dB]';
```

Task 2 - MATLAB code

```
load TXT_20061211_test3.fca;
load TXT_20061211_test4.fca;
load TXT_20061211_test5.fca;
test_3 = TXT_20061211_test3;
test_4 = TXT_20061211_test4;
test_5 = TXT_20061211_test5;

date = ['2006 Dec 11'];

% Barker Coding

Time_3 = unique(test_3(:,1));
Altitude_3 = unique(test_3(:,2));
Size_Time_3 = size(Time_3,1);
Size_Altitude_3 = size(Altitude_3,1);


%SNR Ratio in dB

for i= 1:Size_Altitude_3
   for j= 1: Size_Time_3
      SNR3(i,j) = test_3(i+((j-1)*Size_Altitude_3),4);
   end
end

% Complementary coding

Time_4 = unique(test_4(:,1));
Altitude_4 = unique(test_4(:,2));
Size_Time_4 = size(Time_4,1);
Size_Altitude_4 = size(Altitude_4,1);

%SNR Ratio in dB

for i= 1:Size_Altitude_4
   for j= 1: Size_Time_4
      SNR4(i,j) = test_4(i+((j-1)*Size_Altitude_4),4);
   end
end

% Uncoded data

Time_5 = unique(test_5(:,1));
Altitude_5 = unique(test_5(:,2));
Size_Time_5 = size(Time_5,1);
```

```
Size_Altitude_5 = size(Altitude_5,1);


%SNR Ratio in dB

for i= 1:Size_Altitude_5
   for j= 1: Size_Time_5
      SNR5(i,j) = test_5(i+((j-1)*Size_Altitude_5),4);
   end
end

%Setting limits of colorbar
bottom = min(min(min(SNR3)),min(min(SNR4)));
bottom=min(min(min(SNR5)),bottom);
top  = max(max(max(SNR4)),max(max(SNR5)));
top=max(max(max(SNR3)),top);

%Plotting
FigHandle = figure();
hold on
subplot ( 1 , 3 , 1 ) ;
colormap(jet);
pcolor(Time_3,Altitude_3,SNR3);
shading flat;
% This sets the limits of the colorbar to manual for the first plot
caxis manual
caxis([bottom top]);
ylabel('Altitude [km]');
xlabel(['UT/Date: ',date ]);
title('SNR[dB] [ Barker coding]');
set(FigHandle, 'Position', [100, 0, 600, 800]);

c = colorbar ;
c.Label.String = 'SNR[dB]';

%Plotting
hold on
subplot ( 1 , 3 , 2 ) ;
colormap(jet);
pcolor(Time_4,Altitude_4,SNR4);
shading flat;
% This sets the limits of the colorbar to manual for the first plot
caxis manual
caxis([bottom top]);
ylabel('Altitude [km]');
xlabel(['UT/Date: ',date ]);
title( 'SNR[dB] [Complementary coding]');
```

```
c = colorbar ;
c.Label.String = 'SNR[dB]';

%Plotting
hold on
subplot ( 1 , 3 , 3 ) ;
colormap(jet);
pcolor(Time_5,Altitude_5,SNR5);
shading flat;
% This sets the limits of the colorbar to manual for the first plot
caxis manual
caxis([bottom top]);
ylabel('Altitude [km]');
xlabel(['UT/Date: ',date ]);
title( 'SNR[dB] [Uncoded data]');
c = colorbar ;
c.Label.String = 'SNR[dB]';
hold off

% Values and directions of the horizontal wind
% Barker Coding
for i= 1:Size_Altitude_3
   for j= 1: Size_Time_3
      Zonal_Wind(i,j)= test_3(i+((j-1)*Size_Altitude_3),5);
      Meridional_Wind(i,j)= test_3(i+((j-1)*Size_Altitude_3),6);
   end
end
Horizontal_Wind = sqrt((Zonal_Wind .^ 2) + (Meridional_Wind .^ 2));
Direction_angle = atan2(Zonal_Wind, Meridional_Wind);

figure()
subplot(1,2,1);
pcolor(Time_3,Altitude_3,Horizontal_Wind);
shading flat;
colorbar;
title('Horizontal wind speed [m/s], Baker coding signal');
xlabel(['UT/Date:',date]);
ylabel('Altitude [km]');


subplot(1,2,2);
hold on;
whitebg([0.0 .0 .2]);
quiver(Time_3,Altitude_3,Zonal_Wind,Meridional_Wind,'r');
xlabel(['UT/Date:',date]);
title('Horizontal wind Barker Coding');
axis([min(Time_3),max(Time_3),min(Altitude_3),max(Altitude_3)]);
```

```
% Complementary Coding
for i= 1:Size_Altitude_4
   for j= 1: Size_Time_4
      Zonal_Wind_2(i,j)= test_4(i+((j-1)*Size_Altitude_4),5);
      Meridional_Wind_2(i,j)= test_4(i+((j-1)*Size_Altitude_4),6);
   end
end
Horizontal_Wind_2 = sqrt((Zonal_Wind_2 .^ 2) + (Meridional_Wind_2 .^ 2));
Direction_angle_2 = atan2(Zonal_Wind_2, Meridional_Wind_2);

figure()
subplot(1,2,1);
pcolor(Time_4,Altitude_4,Horizontal_Wind_2);
shading flat;
colorbar;
title('Horizontal wind speed [m/s], Complementary coding signal');
xlabel(['UT/Date:',date]);
ylabel('Altitude [km]');


subplot(1,2,2);
hold on;
whitebg([0.0 .0 .2]);
quiver(Time_4,Altitude_4,Zonal_Wind_2,Meridional_Wind_2,'r');
xlabel(['UT/Date:',date]);
title('Horizontal wind Complementary Coding');
axis([min(Time_4),max(Time_4),min(Altitude_4),max(Altitude_4)]);

% Uncoded Data
for i= 1:Size_Altitude_5
   for j= 1: Size_Time_5
      Zonal_Wind_3(i,j)= test_5(i+((j-1)*Size_Altitude_5),5);
      Meridional_Wind_3(i,j)= test_5(i+((j-1)*Size_Altitude_5),6);
   end
end
Horizontal_Wind_3 = sqrt((Zonal_Wind_3 .^ 2) + (Meridional_Wind_3 .^ 2));
Direction_angle_3 = atan2(Zonal_Wind_3, Meridional_Wind_3);

figure()
subplot(1,2,1);
pcolor(Time_5,Altitude_5,Horizontal_Wind_3);
shading flat;
colorbar;
title('Horizontal wind speed [m/s], Uncoded Data signal');
xlabel(['UT/Date:',date]);
ylabel('Altitude [km]');
```

```
subplot(1,2,2);
hold on;
whitebg([0.0 .0 .2]);
quiver(Time_5,Altitude_5,Zonal_Wind_3,Meridional_Wind_3,'r');
xlabel(['UT/Date:',date]);
title('Horizontal wind Uncoded data');
axis([min(Time_5),max(Time_5),min(Altitude_5),max(Altitude_5)]);
```