Date 17th July 2020

Telecommunication Project Report

Submitted by:

Sachinkumar Pravinbhai Prajapati

Maharshi Patel

Indrajitsinh Solanki

Department of Informational and Electrical Engineering

Guided by

Dr.-Ing. Toralf Renkwitz

Contents

Abstract

- Radar is a detection system that uses radio waves to determine the range, angle, or velocity of objects. It can be used to decry ships, missiles, motor vehicles, weather formation, spacecraft, aircraft and so on. A radar system consists of a transmitter producing electromagnetic waves in the radio or microwaves domain, a transmitting and receiving antenna and a receiver and processor to determine properties of objects.

- A radar has a transmitter to relies the radar signals in all the direction. When these signals contact with the object they scattered back to the all the directions and absorbed by the radar. If the target is moving either towards or go for from the transmitter, and it will be a hold change in the frequency of the radio waves due to the Doppler effect.

<u>Acknowledgment</u>

I would like to express my deepest appreciation to all those who provided me the possibility to complete this report. A special gratitude I give to our project, **Dr.-Ing. Toralf Renkwitz**, whose contribution in stimulating suggestions and encouragement, helped me to coordinate my project especially in writing this report.

Furthermore, I would also like to acknowledge with much appreciation the role of special thanks goes to my team mate, who all helps me to give suggestion about this report. It is also helpful and appreciated to learn more in this subject deeply to **Dr.-Ing. Toralf Renkwitz.**

Tasks

0)Select the raw data file marked with your group number.

- We have taken our group's RAW data file from stud ip.

1) I-Q diagram of raw data radar signal for one receiver and one range gate with signal!

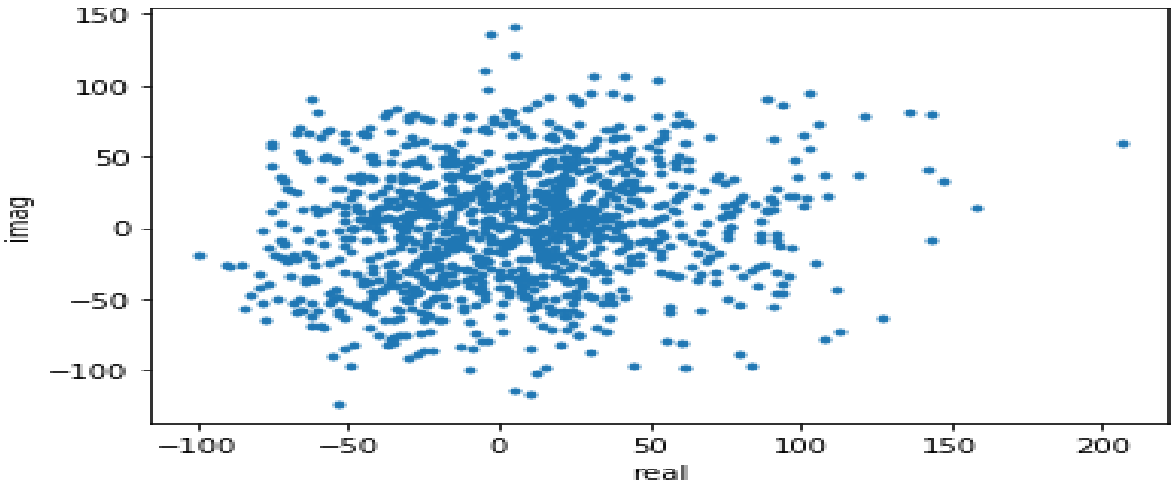

Figure 1: I-Q diagram

-I/Q diagram of raw data determines the signal for imaginary and real parts with uncontrolled ranges.

2a) Power profile - line plot - averaged time series power for each range and receiver.

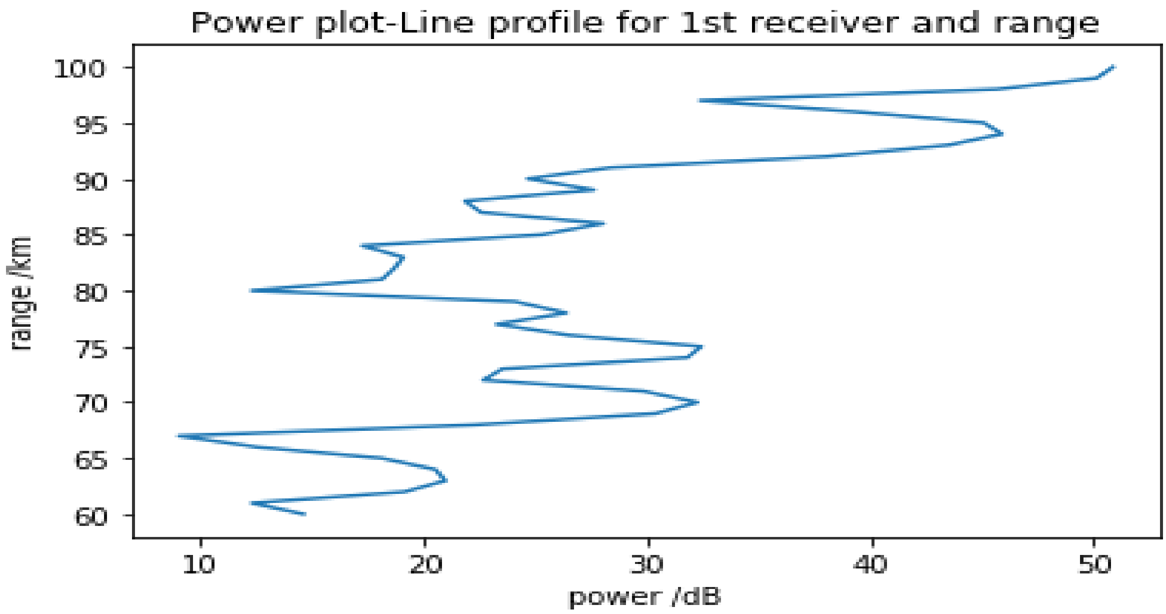

Figure 2: Power profile (Each range and Receiver)

- Above graph represent power profile for 1st receiver and range. Here, the low power detection and higher power detection are 68 km and 95 km respectively. The mean value of power detection at 75 km.

2b) Power profile - Line plot - add the profile for the combined receivers.

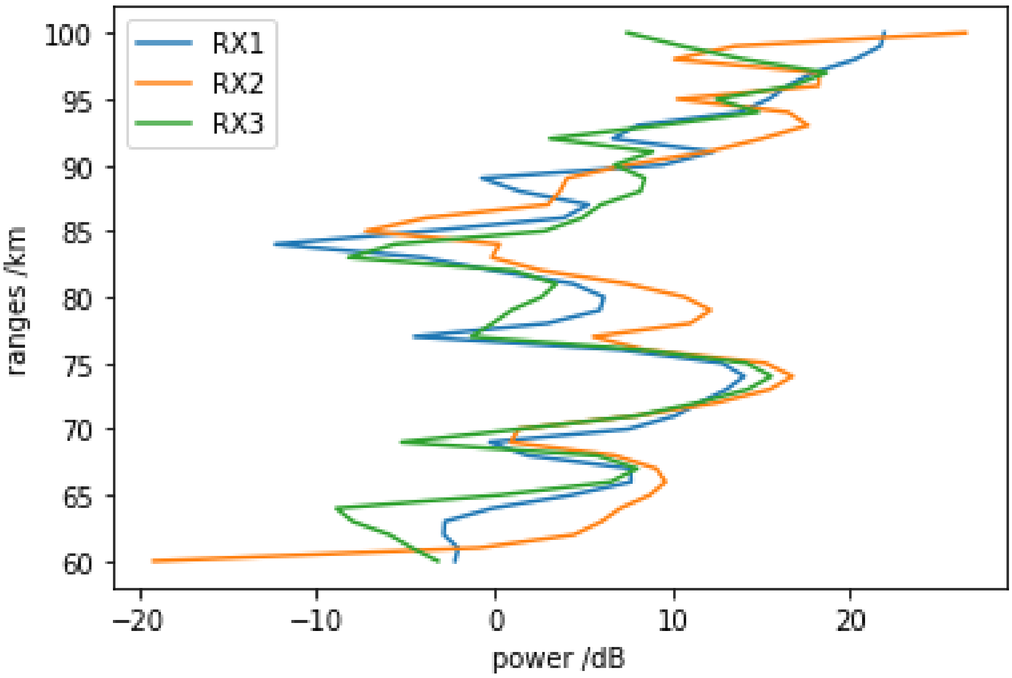

Figure 3: Power Profile (Combined Receivers)

- This graph shows that power profile for three receiver channel or combined receivers. Here, the high pick of power or Correlation plot between 70-75 km and it is near 20 dB. Mean of cross correlation for range case and receiver. Orange line represent **(power + noise)**.

3) Power - pcolor - for the combined receivers - time series all ranges.

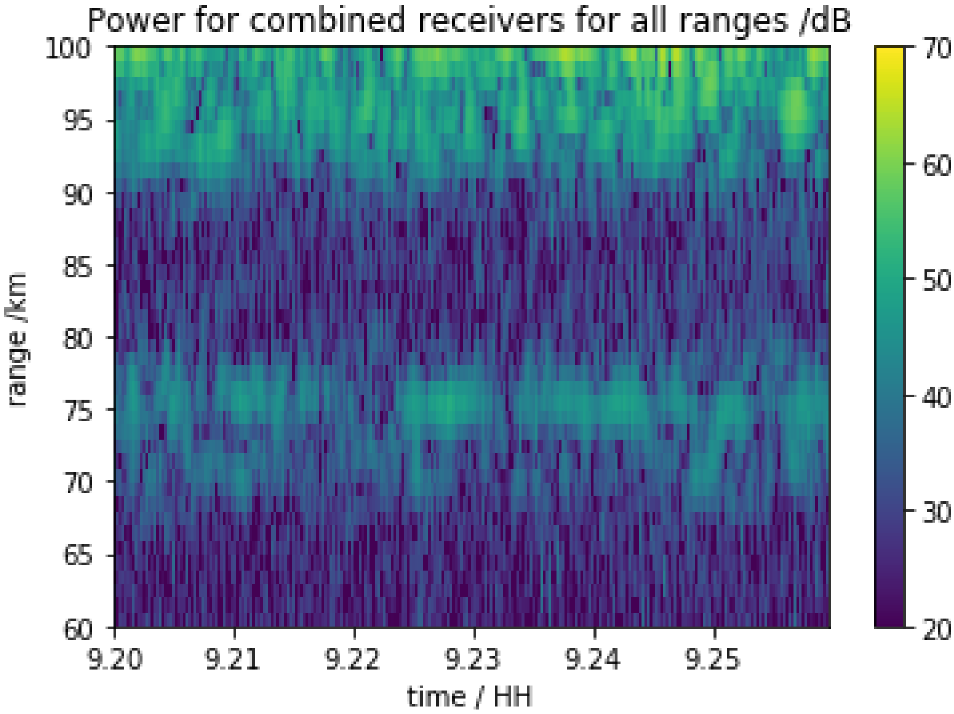

Figure 4: Power (Combined receivers - Time series all ranges)

- This graph shows that the power profile for combined receivers with ranges and time in y and x-axis respectively.
- Here, the blue lines are showing nothing but the yellow lines are defining some noise for ionospheres. The range shows from 75 to 85 km exactly.

4) Spectra - pcolor - for the combined receiver - all ranges – Amplitude & Phase.

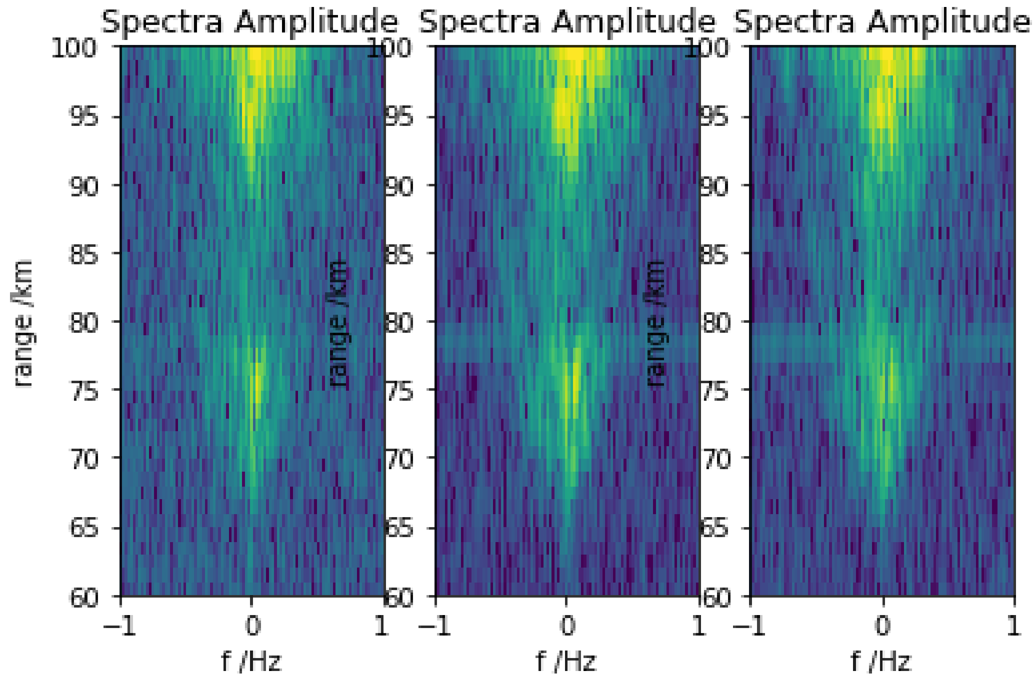

Figure 5: Spectra Amplitude

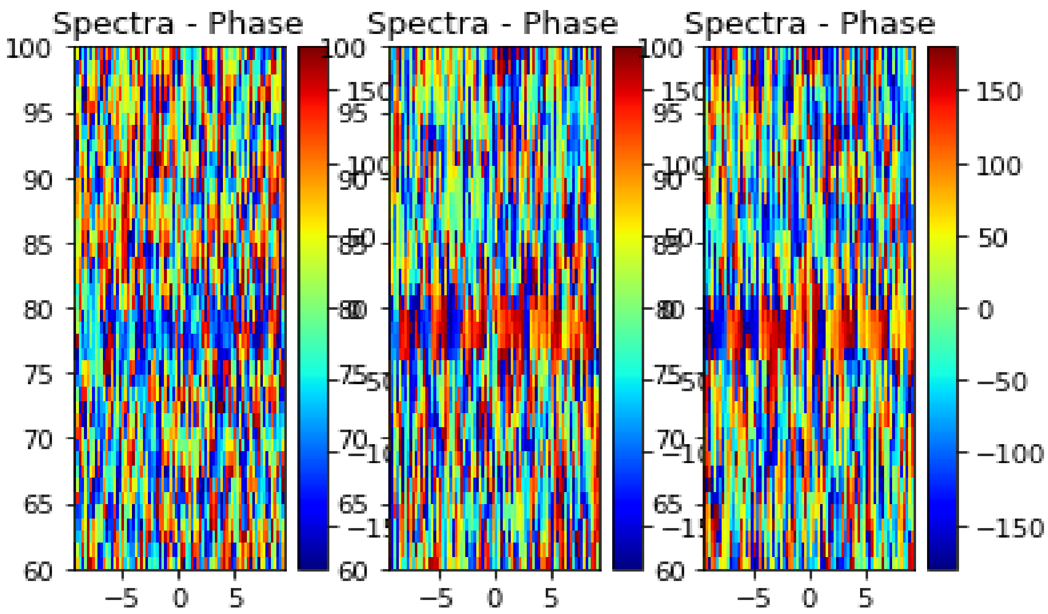

Figure 6: Spectra Phase

- As per the spectra of receiver 1 we can see that the parabolic curve ranges from -0.2 Hz to 0.3 Hz and spectrum looks brighter with the narrow beam at 0 Hz, where the signal power is high.

- As per the diagram in receiver 2 and receiver 3 the parabolic curve increases due to interference with noise ranging from -0.5 Hz to 0.2 Hz and the spectrum bean is become wider thus the signal power reduced.

- As per the Combination of receiver 2 and receiver 3 the curve becomes become thinner ranging from -0.50 to 0.30 thus adding noise to the signal in the range from 80 to 85 and power is least detected compared to other two spectra. Averaging removes the noise and we get only signal which increases the spectral density of the signal.

❖ As per the reduction of noise we will apply coherent integrations.

❖ Why did you choose this integration method and certain number of integrations, if so...?
    - coherent averaging or time-domain averaging. The time-domain integration of measurements in a coherent radar over a sequence of pulses or over an observation interval, prior to estimating the signal properties, to improve the signal-to-noise ratio while minimizing signal processing.
    - The effect of the coherent integration process is to reduce the effective data sampling rate and the Nyquist Frequency.

5) Power - pcolor - combined receivers - time series, all ranges.

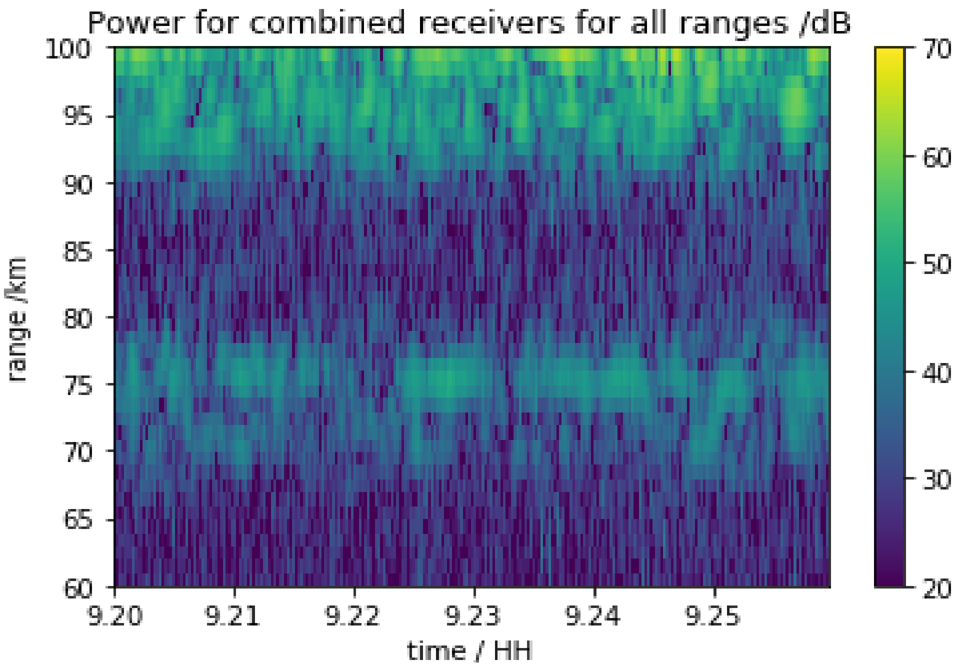

Figure 7: Power (Combined Receivers - Time series, all ranges)

- This graph shows that the Power for combined receivers with time and all ranges.

6) Cross-spectra - pcolor - for all receiver combinations – Amplitude & Phase.

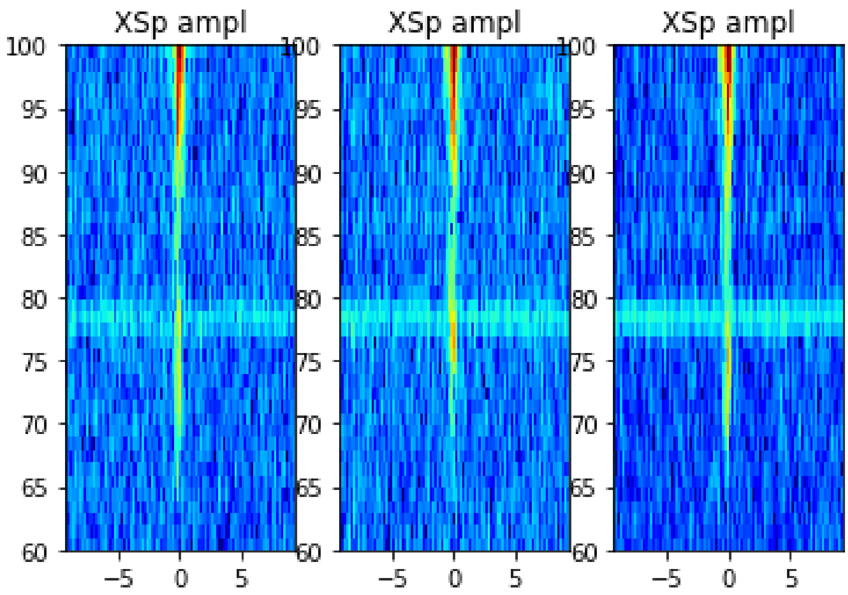

Figure 8: Cross spectra Amplitude

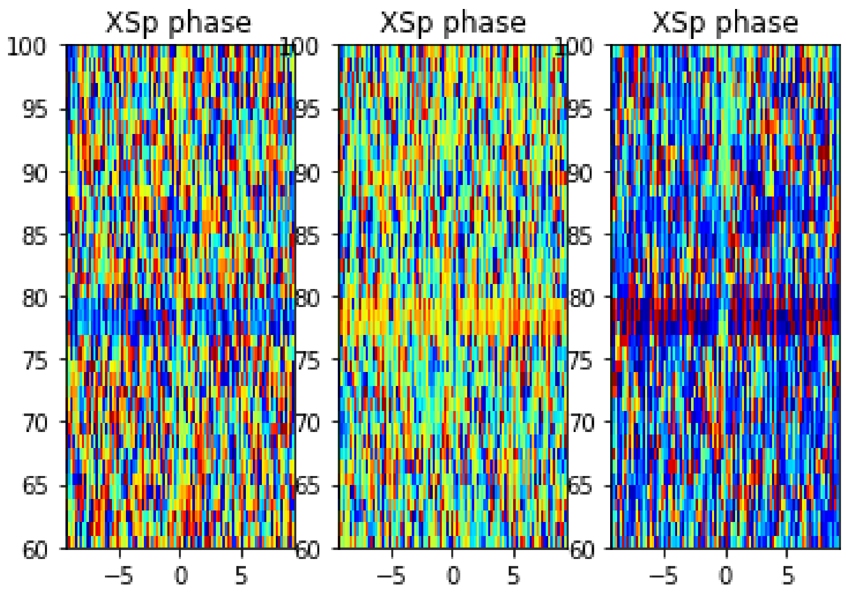

Figure 9: Cross spectra phase

- Cross spectra is the process of multiplying Spectra from one channel conjugate to another channel.

- As per the graph of the cross spectra amplitude blue colour represents noise, red colour represents signal and the curve represents the phase which is below threshold. Cross spectra amplitude is defined as the magnitude of cross spectra divided by 2.
- Noise covers the cloud of Signal in the frequency range -0.5 to 0.1 in range 93 to 100 km and signal is small part of the noise.
- In cross spectra amplitude graph, for 3rd receiver we can see that after applying SNR Signal crosses the threshold values at 90 km thus cancelling noise. We can see that Doppler and Phase shift between individual Patterns. In figure 1 and figure 2 there is negative Doppler Phase shift and in 3rd diagram there is positive Doppler Phase shift, hence we can see that Signal overlaps threshold.

7) Cross-correlation – pcolor – for all receiver combinations and ranges – Amplitude & Phase.

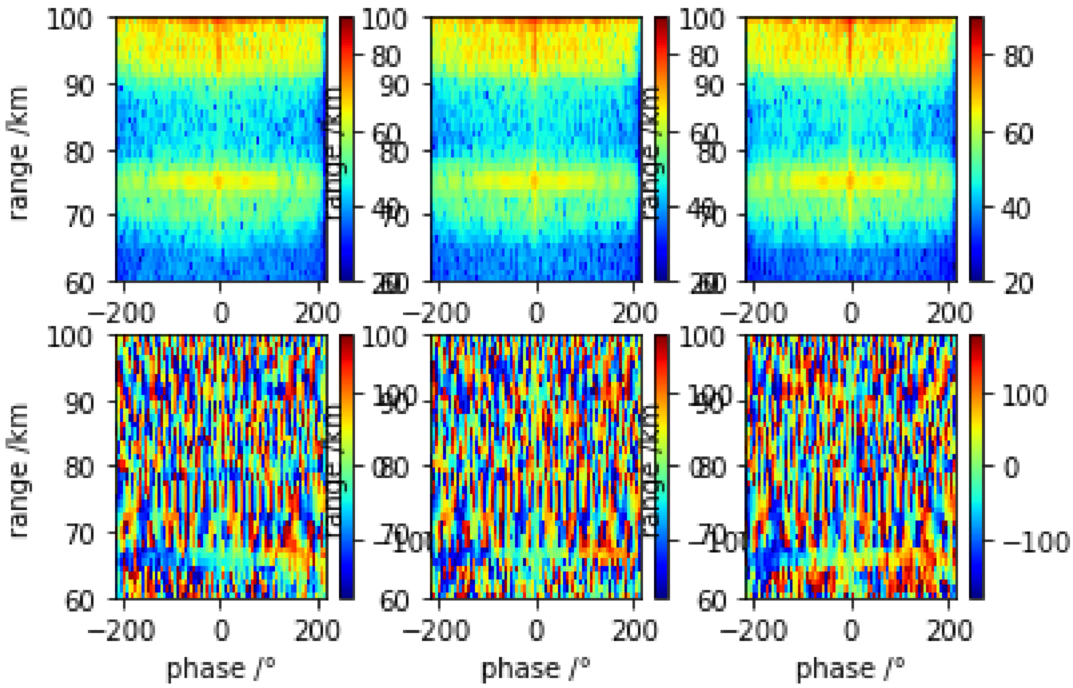

Figure 10: Cross-correlation Amplitude & Phase

Receiver 1: In the first receiver plot the noise intensity is high where the signal in the ranges of 70 to 80 km. So, the receiver represents the signal with higher correlation and the central beam of the signal is the spectrum and the surrounding signal is the phase.

Receiver 2: In the second receiver diagram the signal power is detected at 80 range/km this due to noise alters the signal and for correlation the noise is detected at high bandwidth and naked detected power is at lower amplitude signal in the range of 60 to 65 km.

Receiver 3: By analysing three diagrams, we can see that the stable phase relation seen in third diagram and the plot is scattered from red to blue in the range of 70 to 85 km. The high correlation and at 80 range/km the signal represent echo with different noise apart from the signal the top portion and bottom portion represent phase.

8) Coherence - pcolor - for all receiver combinations and ranges.

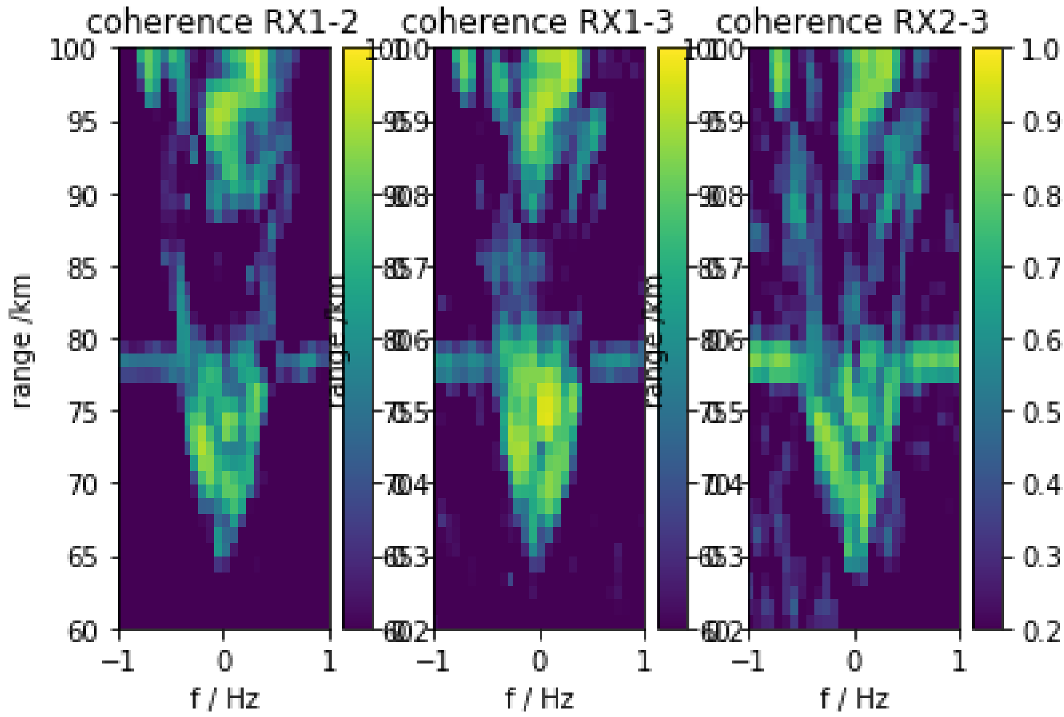

Figure 11: Coherence (All Receiver Combinations and Ranges)

- The structure of coherence is same as normalised cross spectra. It has a parabolic shape.
- In the 1-3 receiver, the coherence is high which have constant phase leading to parabolic curve between -0.5 & 0.3, A beam of spectrum this means that the signal has high spectral density with less amount of noise.
- In 1-2 receiver, the parabolic structure is shifted slightly towards the negative Doppler shifting between -0.9 & 0.4 and the beam of spectrum is reduced from the earlier spectrum due to the addition of noise.
- In 2-3 receiver, the parabolic curve is slightly shifted to positive Doppler and the beam of spectrum is totally discarded due to interference of noise compared to the other two spectrums the frequency not in range.

- **9) Final Python code for all tasks.**

```
# -*- coding: utf-8 -*-
"""
Created on Fri Jul 17 14:28:27 2020
@author: MAHARSHI PATEL
"""
# %reset -f
import scipy.io as spio
import os
import matplotlib.pyplot as plt
import numpy as np
# import datetime
# DataPath='D:/Python/Uni_HWI/scripts20/RAW/'
DataPath='D:/Hochschule Wismar/stud.ip/Sem
2/Lecture_Nachrichtentechnisches_Projekt_-_2020 (3)/RAW data/'
Files=os.listdir(DataPath)
currentfile=str(DataPath)+str(Files[12])
# importing MATLAB mat file   (containing radar raw data)
mat = spio.loadmat(currentfile, squeeze_me=True)
datenums=mat['datenums']
ranges=mat['ranges']
data=mat['data']
# datenums ~ days since year 0
# here only the time is important for us -> hours, minutes, seconds
# => fraction / remainder of the integer
t=(datenums-np.floor(np.min(datenums)))*24
# number of range gates, data points, receivers
noRG=np.size(data,0)
```

```
noDP=np.size(data,1)
noRx=np.size(data,2)
RXsel=0;
# time series
y=data [1,:,RXsel]

# I/Q diagram
plt.figure()
plt.plot(np.real(y),np.imag(y),'.')
plt.xlabel('real')
plt.ylabel('imag')
plt.title('I/Q diagram for one receiver and one range with signal')

# power of the complex valued voltages (measured by the receiver)
PWR=20*np.log10(np.abs(data[:,:,RXsel]))
#PWR=20*np.log10(np.abs(y))
PWR.shape
Type (PWR)
plt.figure()
plt.plot(PWR[:,0],ranges)
plt.ylabel('range /km')
plt.xlabel('power /dB')
plt.title('Power plot-Line profile for 1st receiver and range')
plt.figure()
plt.pcolor(t,ranges,PWR)
plt.xlabel('time / HH')
plt.ylabel('range /km')
```

```
plt.title('power /dB')
plt.clim(20,70)
plt.colorbar()
# combine the data of all three receivers - complex sum!
PWR=np.zeros([noRG,noRx])
for rx in range(noRx):
    PWR[:,rx]=20*np.log10(np.abs(np.mean(data[:,:,rx],1)))
plt.figure()
plt.plot(PWR,ranges)
plt.legend(['RX1','RX2','RX3'])
plt.xlabel('power /dB')
plt.ylabel('ranges /km')
datacomb=np.sum(data,2)

# power for the combined receivers
PWRcomb=20*np.log10(np.abs(datacomb))
plt.figure()
plt.pcolor(t,ranges,PWRcomb)
plt.xlabel('time / HH')
plt.ylabel('range /km')
plt.title('Power for combined receivers for all ranges /dB')
plt.clim(20,70)
plt.colorbar()
def make_fft(t,y):
    dt = t[1]-t[0] # dt -> temporal resolution ~ sample rate
    f = np.fft.fftfreq(t.size, dt) # frequency axis
    Y = np.fft.fft(y) # FFT
```

```
    f=np.fft.fftshift(f)
    Y= np.fft.fftshift(Y)/(len(y))
    return f,Y
tsec=t*60*60
f,spec=make_fft(tsec,data[-25,:,RXsel])
plt.figure()
plt.plot(f,10*np.log10(abs(spec)))
plt.grid ('on')
plt.xlabel('f / Hz')
plt.ylabel('amplitude /dB')

# Spectra for all ranges and all receivers
Spectr=np.zeros([noRG,noDP,noRx])+1j*np.zeros([noRG,noDP,noRx])
for rx in range(noRx):
    for rg in range(noRG):
        f,Spectr[rg,:,rx]=make_fft(tsec,data[rg,:,rx])
plt.figure()
for rx in range(noRx):
    plt.subplot(1,3,rx+1)
    plt.pcolor(f,ranges,10*np.log10(abs(Spectr[:,:,rx])))
    plt.clim([-15, 15])
    plt.xlabel('f /Hz')
    plt.ylabel('range /km')
    plt.xlim([-1,1])
    plt.title('Spectra Amplitude')
    # plt.colorbar()
phases=[]
```

```
plt.figure()
for rx in range(noRx):    # rx=0
    plt.subplot(1,3,rx+1)
    phases=np.angle(Spectr[:,:,rx])/np.pi*180
    plt.pcolor(f,ranges,phases,cmap='jet')
    plt.clim([-180, 180])
    plt.colorbar()
    plt.title('Spectra Phase')

 # perform coherent integrations
def make_ci(t,y, ci):
    nptsn=int(np.floor(len(y)/ci))
    yn=np.empty(nptsn)+1j*np.empty(nptsn)
    tn=np.empty(nptsn)
    for i in range(0,nptsn):
        yn[i]=np.mean(y[i*ci:i*ci+ci-1])
        tn[i]=np.mean(t[i*ci:(i+1)*ci])
    return tn,yn
# make FFT spectrum, frequency axis
def make_fft(t,y):
    dt = t[1]-t[0] # dt -> temporal resolution ~ sample rate
    f = np.fft.fftfreq(t.size, dt) # frequency axis
    Y = np.fft.fft(y)   # FFT
    f=np.fft.fftshift(f)
    Y= np.fft.fftshift(Y)/(len(y))
    return f,Y
# number of coherent integrations of I/Q raw data (time series)
```

```
ci=2 #low noise as comapred to below Ci values
#ci=12    Moderate Noise
#ci=30   After performing CI, we found that Noise is very large
y=data[-25,:,0]
tn,yn=make_ci(t,y,ci)
plt.figure()
plt.subplot(1,2,1)
plt.plot(np.real(y),np.imag(y),'-*')
plt.xlim([-100,100])
plt.ylim([-100,100])
plt.subplot(1,2,2)
plt.plot(np.real(yn),np.imag(yn),'-*')
plt.xlim([-100,100])
plt.ylim([-100,100])
# lentgh of the "new" integrated time series
noDPn=int(np.floor(noDP/ci))
# predefine matrix for integrated raw data
datan=np.zeros([noRG,noDPn,noRx])+1j*np.zeros([noRG,noDPn,noRx])
for rx in range(noRx):
    for rg in range(noRG):
        tn,datan[rg,:,rx]=make_ci(t,data[rg,:,rx],ci)
#Powwer-combined recievers , time series and all ranges
datacomb=np.sum(datan,2)
PWRcomb=20*np.log10(np.abs(datacomb))
plt.figure()
plt.pcolor(tn,ranges,PWRcomb)
plt.xlabel('time / HH')
```

```
plt.ylabel('range /km')
plt.title('combined power /dB')
plt.clim(20,70)
plt.colorbar()

# Cross-Spectra for all ranges and all receivers
XSpectr=np.zeros([noRG,noDP,noRx])+1j*np.zeros([noRG,noDP,noRx])
XSpectr[:,:,0]=Spectr[:,:,0]*np.conj(Spectr[:,:,1])
XSpectr[:,:,1]=Spectr[:,:,0]*np.conj(Spectr[:,:,2])
XSpectr[:,:,2]=Spectr[:,:,1]*np.conj(Spectr[:,:,2])
plt.figure()
plt.pcolor(f,ranges,10*np.log10(abs(XSpectr[:,:,1]))/2)
plt.pcolor(f,ranges,np.angle(XSpectr[:,:,1]))
plt.colorbar()
plt.clim([-15, 15])
plt.figure()
for rx in range(noRx):
    plt.subplot(1,3,rx+1)
    plt.pcolor(f,ranges,10*np.log10(abs(XSpectr[:,:,rx])/2),cmap='jet')
    plt.clim([-15, 15])
    plt.title('XSp ampl')
phases=[]
plt.figure()
for rx in range(noRx):    # rx=0
    plt.subplot(1,3,rx+1)
    phases=np.angle(XSpectr[:,:,rx])/np.pi*180
    plt.pcolor(f,ranges,phases,cmap='jet')
```

```
    plt.clim([-180, 180])
    # plt.colorbar()
    plt.title('XSp phase')
    plt.figure()
for rx in range(noRx):
    plt.subplot(1,3,rx+1)
    ampl=10*np.log10(abs(XSpectr[:,:,rx])/2)
    SNRsel=ampl<-11
    ampl[SNRsel]="nan"
    plt.pcolor(f,ranges,ampl,cmap='jet')
    plt.clim([-10, 20])
    plt.title('XSp ampl')
plt.figure()
for rx in range(noRx):    # rx=0
    plt.subplot(1,3,rx+1)
    phases=np.angle(XSpectr[:,:,rx])/np.pi*180
    SNRsel=10*np.log10(abs(XSpectr[:,:,rx])/2)<-11
    # phases[SNRsel]=float("nan")
    phases[SNRsel]="nan"
    plt.pcolor(f,ranges,phases,cmap='jet')
    plt.clim([-180, 180])
    # plt.colorbar()
    plt.title('XSp phase')
    plt.xlim([-.75,.75])

# cross-correlation for one range and two receivers -> testing reasons
RG=17
```

```
x1=data[RG,:,0]

x2=data[RG,:,1]

# t=np.linspace(0,15,16)

# x1=[1,0,0,1,1,1,0,0,0,1,1,0,0,1,0,0,]

# x2=np.round(np.random.rand(16))

plt.figure()

plt.plot(t,np.abs(x1),t,np.abs(x2))

# xcor=signal.correlate(x1s,x2s,method='direct',mode='full')

xcor=signal.correlate(x1,x2,method='direct',mode='full')

# xcor2=np.correlate(a=x1,v=x2)

# xcor3=np.cov(x1,x2)

#
xcor4=signal.correlate(np.real(x1),np.real(x2))+1j*signal.correlate(np.imag(x1),
np.imag(x2))

# xcor=xcor/max(xcor)

tc=(t-min(t))*60*60

t2=-1*tc[::-1]

t2=np.append(t2, tc[1:])

plt.figure()

plt.subplot(2,1,1)

plt.plot(t2,abs(xcor))

plt.xlabel('tau')

plt.ylabel('abs(xcor)')

plt.subplot(2,1,2)

plt.plot(t2,np.angle(xcor)/np.pi*180)

plt.xlabel('tau')

plt.ylabel('phase /°')

# cross-correlation for all ranges and all receivers
```

```
Xcor=np.zeros([noRG,noDP*2-1,noRx])+1j*np.zeros([noRG,noDP*2-1,noRx])
for rg in range(noRG):
    Xcor[rg,:,0]=signal.correlate(data[rg,:,0],data[rg,:,1])
    Xcor[rg,:,1]=signal.correlate(data[rg,:,0],data[rg,:,2])
    Xcor[rg,:,2]=signal.correlate(data[rg,:,1],data[rg,:,2])
plt.figure()
for rx in range(noRx):
    plt.subplot(2,3,rx+1)
    plt.pcolor(t2,ranges,10*np.log10(np.abs(Xcor[:,:,rx])),cmap='jet')
    plt.xlabel('abs(xcor) /dB')
    plt.ylabel('range /km')
    plt.clim([20,90])
    plt.colorbar()
for rx in range(noRx):
    plt.subplot(2,3,rx+4)
    plt.pcolor(t2,ranges,np.angle(Xcor[:,:,rx])/np.pi*180,cmap='jet')
    plt.clim([-180, 180])
    plt.colorbar()
    plt.ylabel('range /km')
    plt.xlabel('phase /°')
# line plots for the mean of xcor (along the data points - time )
plt.figure()
plt.subplot(1,3,1)
plt.plot(PWR,ranges)
plt.legend(['Rx1','Rx2','Rx3'])
plt.xlabel('power /dB')
plt.grid('on')
```

```
plt.subplot(1,3,2)
plt.plot(10*np.log10(Xcor[:,noDP,:]),ranges)
plt.legend(['1-2','1-3','2-3'])
plt.xlabel('xcor ampl. /dB')
plt.grid('on')
plt.subplot(1,3,3)
plt.plot(np.angle(Xcor[:,noDP,:])/np.pi*180,ranges)
plt.legend(['1-2','1-3','2-3'])
plt.xlabel('phase / °')
plt.grid('on')

# coherence for all ranges and all receivers
npts=256
cohtemp=np.zeros(256)
Coh=np.zeros([noRG,npts,noRx])
for rg in range(noRG):
    f,cohtemp=signal.coherence(data[rg,:,0],data[rg,:,1],Fs)
    Coh[rg,:,0]=np.fft.fftshift(cohtemp)
    f,cohtemp=signal.coherence(data[rg,:,0],data[rg,:,2],Fs)
    Coh[rg,:,1]=np.fft.fftshift(cohtemp)
    f,cohtemp=signal.coherence(data[rg,:,1],data[rg,:,2],Fs)
    Coh[rg,:,2]=np.fft.fftshift(cohtemp)
f=np.fft.fftshift(f)
plt.figure()
for rx in range(noRx):
    plt.subplot(1,3,rx+1)
    plt.pcolor(f,ranges,(np.abs(Coh[:,:,rx])))
```

```
    plt.xlabel('f / Hz')
    plt.ylabel('range /km')
    plt.clim([0.2,1])
    plt.xlim([-1,1])
    plt.colorbar()
plt.subplot(1,3,1)
plt.title('coherence RX1-2')
plt.subplot(1,3,2)
plt.title('coherence RX1-3')
plt.subplot(1,3,3)
plt.title('coherence RX2-3')
```

Summary

- To summarise all over report, signals from the raw data of the radar system and signal processing the following data operation such as there is a power which derived in spectra, X-spectra, coherence and correlation of ranges and time series.