Homework 1.a : Radar Imaging

Prof. Andrea Virgilio Monti Guarnieri & Dr. Marco Manzoni

Mohammadjavad Embrahimpour 10696374
Group members :Thibault Fievez , Hiva Amiri,Kimia Kiyan ,Vahid Rajabi

October 2020

1 Introduction and virtual Array

In this paper we are depicting the steps we have done to determine the DoA (direction of arrival) of a signal with respect to an array of antennas. The idea is that there are transmitters antenna and receivers , and the receivers receive the signal that has been reflected on a target, and with that we are able to locate the direction of the target. To realise that we used two different methods proposed in the given Homework 1.a paper:

- FFT Described in section 3.
- BACKPROJECTION Described in section 4.

2 MIMO Radar Array

2.1 Design of the equivalent virtual array

2.1.1 Question and context

Let's suppose to have a 2D field of view (FOV) scanned by a MIMO radar. The operational wavelength is 9.6 GHz and the field of view (the region where the target could be) is delimited by the coordinates:

- XMIN = 35m
- XMAX = 35m
- YMIN = 100m
- YMAX = 150m

The desired minimum resolution in the FOV is 2m. Provide the specifications of the ULA needed to meet the requested resolution. What is the minimum number of antenna to place in order to properly sample the signal without aliasing?

2.1.2 Answer

To answer this question we shall use the equations provided from the homework 1a paper. We know that we want our target in the FOV, defined in the previous paragraph, a spatial definition ρ_x of 2m minimum. Using equation 1 [1], we can relate that parameters to others such as R, d_x (the distance between element of our virtual array) , N (the number of element in our virtual array), The goal is to inverse such equation and isolate N. That would be our minimal number of element required to have such ρ_x

[1] L=N.d_x , L being the total length of the array

$$\rho_x = \frac{\lambda}{2L}R = \frac{\lambda}{2Nd_x}R \tag{1}$$

First and foremost we can find the R_{max} that would give us the worst spatial definition.In another words, to satisfy minimum resolution equal to 2 m we have to put R_{max} in equation (1).As our FOV is determined by the context of the exercise we can simply use equation 2.

$$R_{max} = \sqrt{x_{max}^2 + y_{max}^2} = 154m \tag{2}$$

Since our configuration is limited by defined FOV, the angle of arrival is between $(-\theta_{max}, \theta_{max})$ which our θ_{max} is equal to $(19.29°)$

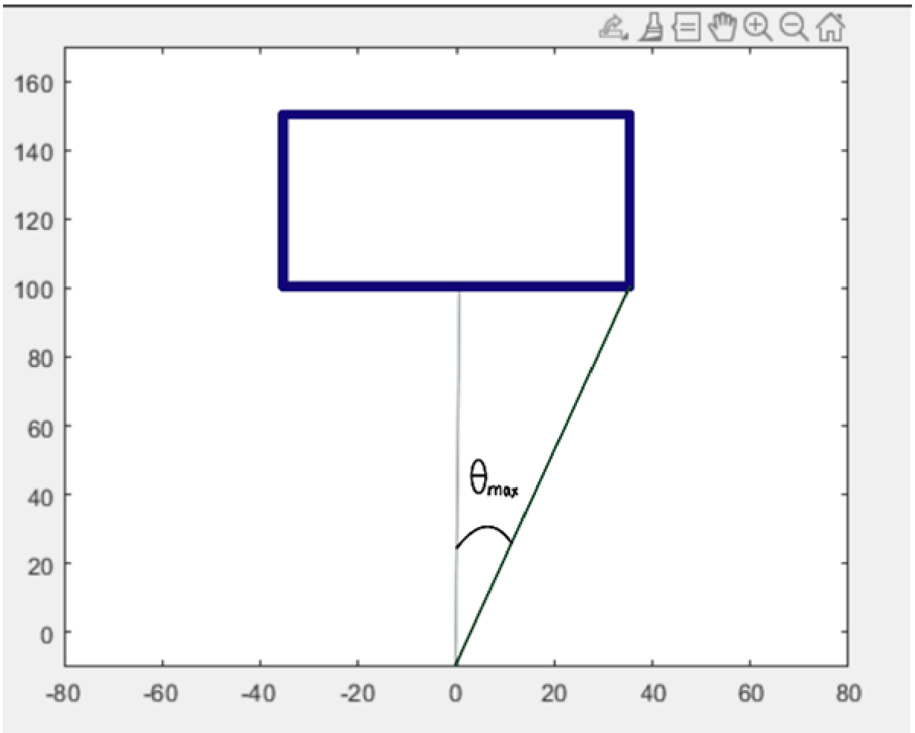

Figure 1: Maximum theta

Now, let's find d_x, we know that the spacial frequency is linked to the maximal angle of arrival as described by equation 3 [2]. In our case , we can imagine that our virtual antenna array is "sampling" the signal arriving into the array. Each specific element of the array "samples" the incoming wave, it take a "piece of the incoming wave that reaches it ", therefore we call it a spatial "sampling", with a spacial frequency. And just like in signal analyses, to avoid aliasing we need to make sure that $f_{sample} = 2f_{max}$ at least. But $f_{sample} = \frac{1}{d_x}$, so we can also find the appropriate $d_x = \frac{1}{2f_{max}} = 0.0236m$.

$$f_{max} = \frac{2sin(\theta_{max})}{\lambda} \tag{3}$$

With this , and by taking all those equation together we find that for imposing $\rho_x = 2m$ and R_{max} determined by the wanted FOV we need a minimum N=51 elements.

[2]This equation is true for any θ , not just the maximum values, $\theta = sin^{-1}(\frac{f_x\lambda}{2})$

```
%% configuration
c = 3e8; % light speed
f = 9.6e9;
lambda = c / f;
number_of_tx = 3;
number_of_rx = 17;
max_theta = atan(0.35);
dx = lambda/(2*sin(max_theta));
```

Figure 2: Code of this part

2.2 Design of the MIMO radar

Let's now design an actual MIMO radar that matches the requirements of the previous point. Now we have to calculate the distance between the transmitters d_t and for receivers d_R such that center of our virtual array should be placed on origin. By using $V = \frac{d_T + d_R}{2}$ we find that $d_t = N d_R$ & $d_R = 2d_x$. By doing this we assure that the distance between each virtual element is d_x.

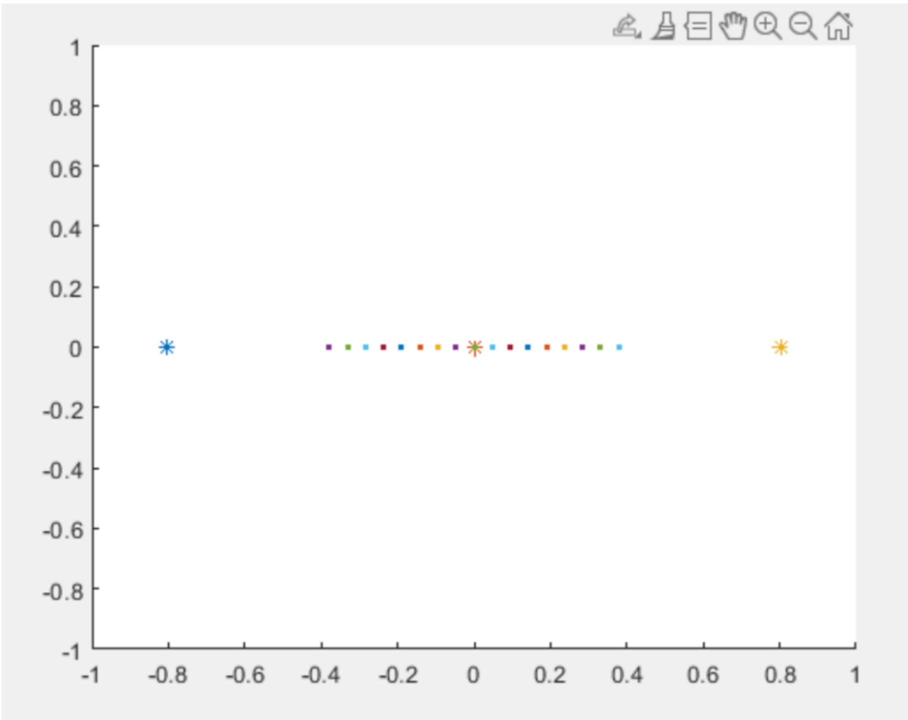

Figure 3: Layout of the T_x (the stars) and R_x (the dots).

Our corresponding virtual antenna is symmetrically placed with respect to the $(0, 0)$ with the distance d_x which is shown in the figure bellow

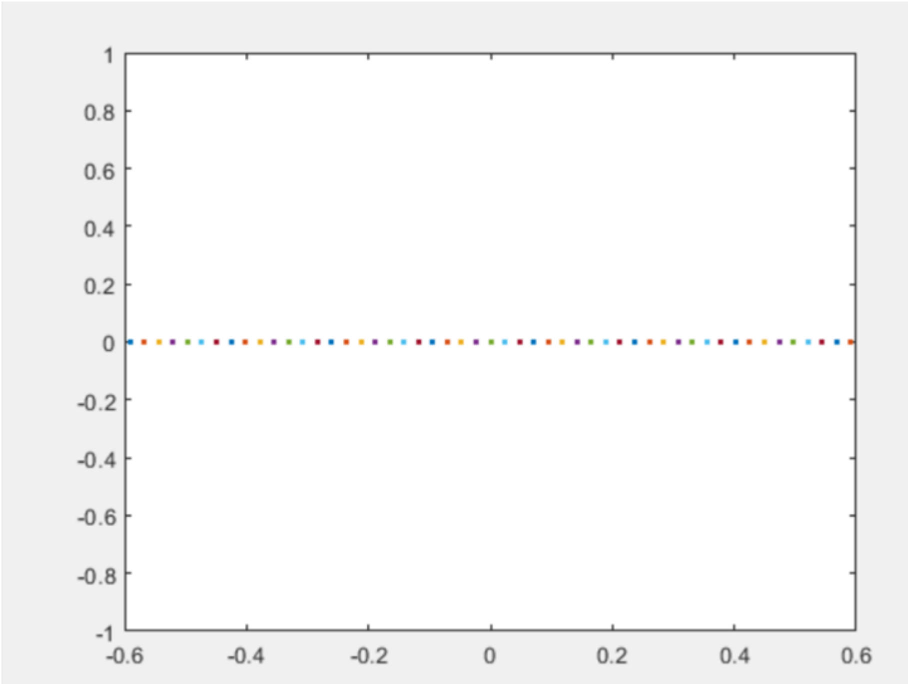

Figure 4: Layout of the virtual antenna positions

3 DoA estimation using FFT

3.1 Single target without noise

3.1.1

Now we generate a random point inside our FoV.

```
%% part 3.1.1 : place the target in a random position
xBox= [-35,-35,35,35,-35];
yBox= [100,150,150,100,100];
figure(1)
plot(xBox,yBox)
hold on
a=rand;
target_x1 = -35+(70)*a;
target_y1 = 100+(50)*a;
plot (target_x1, target_y1, '.' )
xlim ([-80 80])
ylim ([-10 150])
```

Figure 5: Generation of the target randomly in the FOV

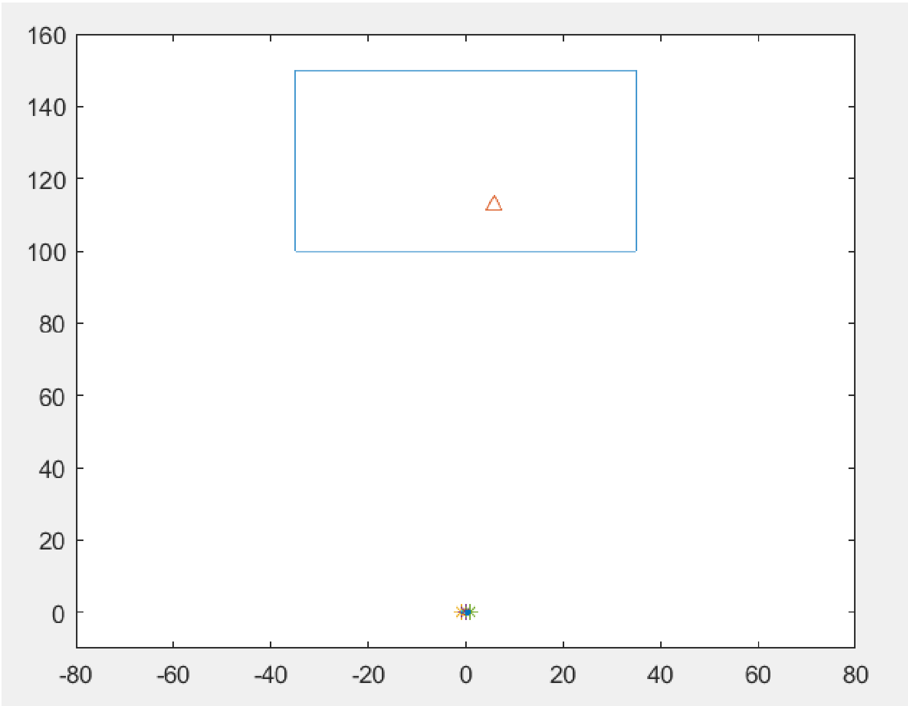

Figure 6: The whole configuration , with the blue rectangle being the field of view (FOV), the red triangle being a target.

3.1.2

Now we place our receiver and transmitter based on calculations.

```
%% part 3.1.2 placing TX and RX elements in 2D space
mode_tx = mod(number_of_tx,2);
mode_rx = mod(number_of_rx,2);
if mode_rx == 0
    xn = dx_rx*((number_of_rx+1)/2);
    rx_position = (-xn+dx_rx : dx_rx : xn-dx_rx );
else
    xn = ((number_of_rx-1)/2)*dx_rx;
    rx_position = (-xn : dx_rx : xn);
end
if mode_tx == 0
    xn = dx_tx*((number_of_tx+1)/2);
    tx_position = (-xn+dx_tx : dx_tx : xn-dx_tx );
else
   xn = ((number_of_tx-1)/2)*dx_tx;
    tx_position = (-xn : dx_tx : xn);
end
plot(tx_position,0,'*')
plot(rx_position,0,'.')
hold off
```

Figure 7: Generation of the $T_X$and R_x.This code was also used to generate the figures at point 2.2

3.1.3

```
%% part 3.1.3 : theta1 calculation

theta1 =  rad2deg(atan(target_x1 / target_y1)) % computing true angle of arival using geometry
```

Figure 8: Simply calculating the actual angle of arrival according to coordinates of the target

3.1.4

```
%% 3.1.4 : propagating the signal

R0_1 = zeros(1,number_of_tx);
R1_1= zeros(1,number_of_rx);
for m = 1:1:number_of_tx
   R0_1(1,m) = sqrt((target_x1 -(tx_position(1,m)))^2 + target_y1^2); % calculating R1 and store it in an array
end

for n = 1:1:number_of_rx
     R1_1(1,n) = sqrt((target_x1 -rx_position(1,n))^2 + target_y1^2);% calculating R2 and store it in an array
end
w=rand;
phi=rand*2*pi;
row_p = w*exp(-1i*phi) ;
Smn1 = zeros(number_of_tx, number_of_rx);
R1 = zeros(1,number_of_tx*number_of_rx);
temp = 1;
for m = 1:1:number_of_tx
   for n = 1:1: number_of_rx
       R1(1,temp) =  R0_1(1,m)+R1_1(1,n);
       temp = temp + 1;
   end
end
received_vector1 = (row_p./R1).*exp(-(1i*2*pi.*R1)/lambda); %% received signal stored in an array
```

Figure 9: We implement the equation (1) provided in the homework paper, to achieve received signals vector.

3.1.5

```
%% part 3.1.5 fft calculation
max_theta =(atan(0.35));
fs = 4 * sin(max_theta) / lambda ; % sampling frequency
dx = 1/fs ;
sample_number = 2000*length(received_vector1);% zero padding
ff1 = fftshift(fft(received_vector1,sample_number)); % uaing fft shift to shift properly between -fs/2 and fs/2.

power1 = abs(ff1).^2 / sample_number; % calculating the Power spectrum
x = (-fs/2 : fs/sample_number : fs/2-fs/sample_number);
figure(2)
plot(x , power1) % plot the Power spectrum
```

Figure 10: We apply the FFT to the received vector computed on the previous point

3.1.6

For a target that was positioned in $x = 5.8m$ and $y = 112.59$, the following spectrum is received.

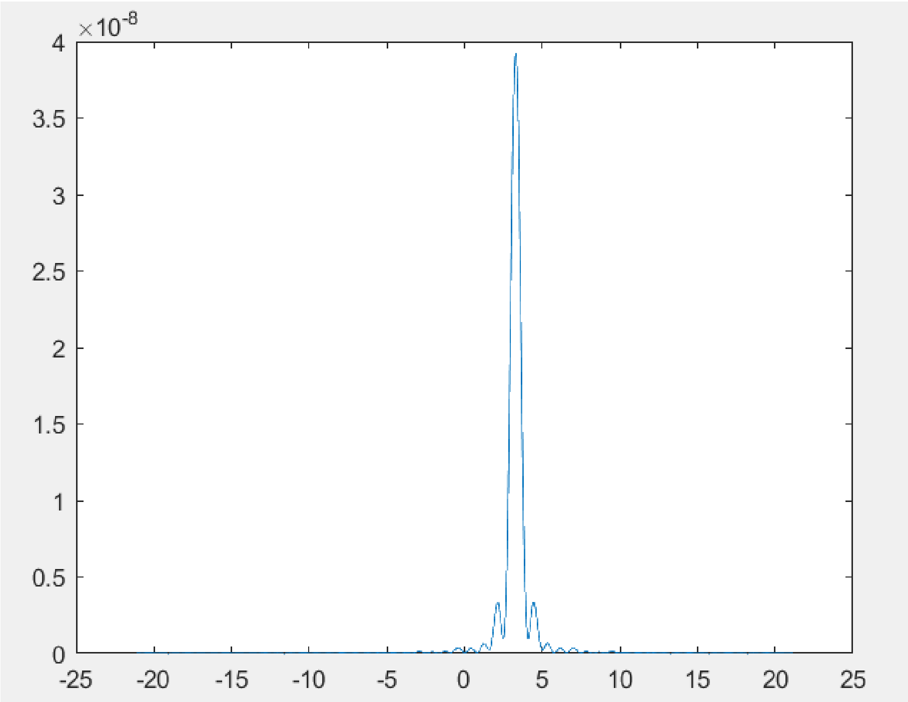

Figure 11: Spacial sprectrum

We therefore just need to find the max/ peak of that spectrum and determine the angle from the equation 3.This equation is true for any θ so we get $\theta = sin^{-1}(\frac{f_x \lambda}{2})$. As seen on figure 12.

```
%% part 3.1.6 angle estimation
[M1,I1] = max(power1);
fx1 = -fs/2 + I1*fs/sample_number;% calculating the fx of the maxmumim
sine_theta1 = fx1 * lambda / (2) ;
estimated_theta1 = rad2deg(asin(sine_theta1))% estimate the theta
```

Figure 12: Code that calculate the angle of incidence from the maximum of the spectrum.

```
theta1 =

    2.9535


estimated_theta1 =

    2.9537
```

Figure 13: Comparing the result to the real angle of incidence

3.2 Validation of the nominal resolution - Multiple targets

In this section we use to target to "stress" test our implementation of the code for 2 targets. We can see on figure 16 the spectrum the 2 targets and can estimate 2 different angle of arrival corresponding the the situation decpicted by 14.

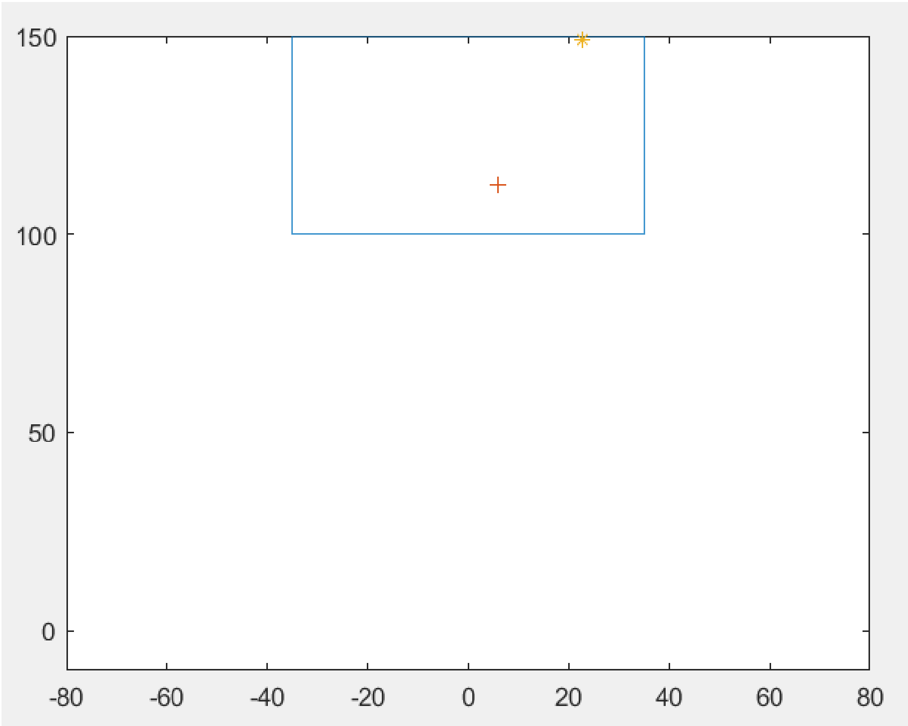

Figure 14: The positioning of the 2 targets

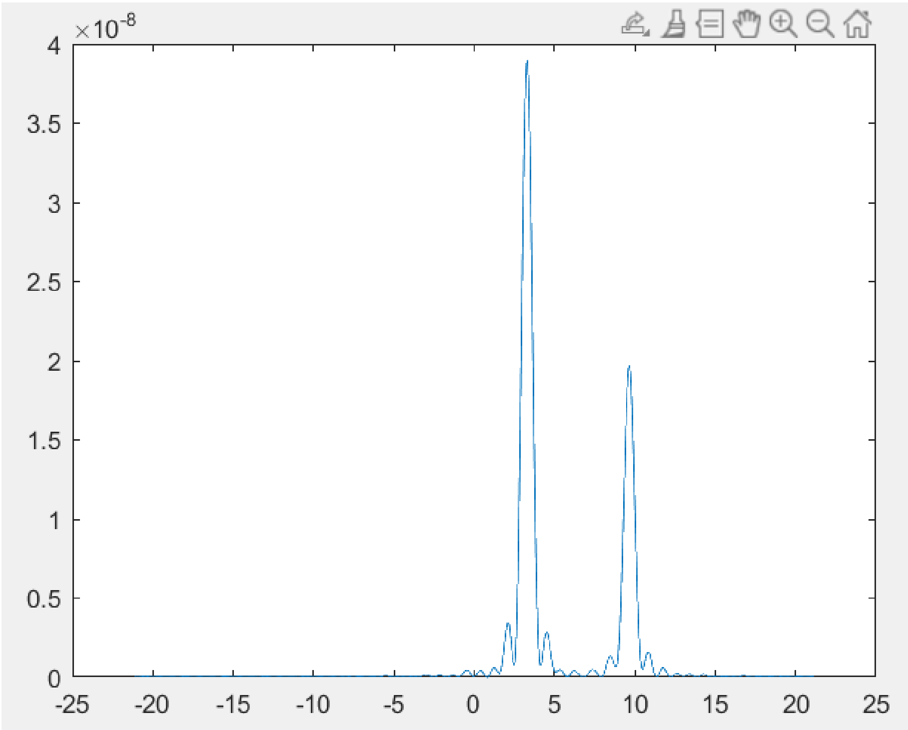

Figure 15: The spectrum for 2 targets

```
two_obj_theta =

    2.9535    8.6571


two_object_estimated_theta =

    2.9496    8.6779
```

Figure 16: The estimated thetas vs real thetas

4 DoA with backprojection

The intuition behind backprojection is we assume assume propagating the signal from our virtual antennas proportionnaly to the received signal in all of them. By doing so the signals are added together in the space domain and create "constructive" or "destructive" interference.

We can therefore find the maximum , the constructive interference that should be in the DOA of our target.

For this section, we are using a meshgrid function to creates "pixels" of our FOV. Those pixels are of dimension $d_x.d_x$.Using the formula given in the homework and plotting it with the function imagesec we get the following results:

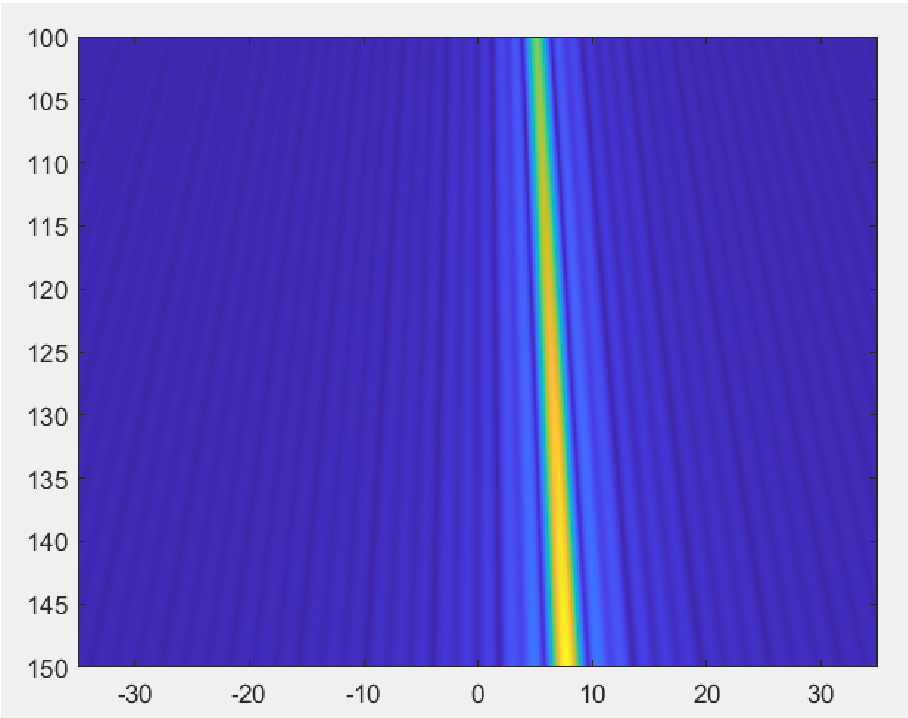

Figure 17: DoA with backprojection

And that gives us an angle of incidence as seen on figure 18

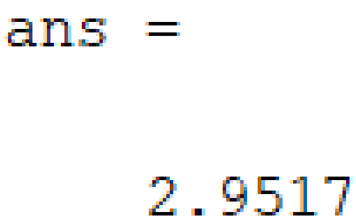

Figure 18: Result of backprojection