Principles of Classical and Modern Radar

Monostatic Pulse Radar for Complex Targets

Dimitrios Kazantzis
March 2020

Contents

Radar Description

The aim of this project is to design a PC-based *Monostatic Pulse Radar* which employs the same Phase Array at both the radar's Tx and Rx for detecting and localising multiple complex targets.

The following figure shows the system modelling of the Monostatic Phased Array to be designed.

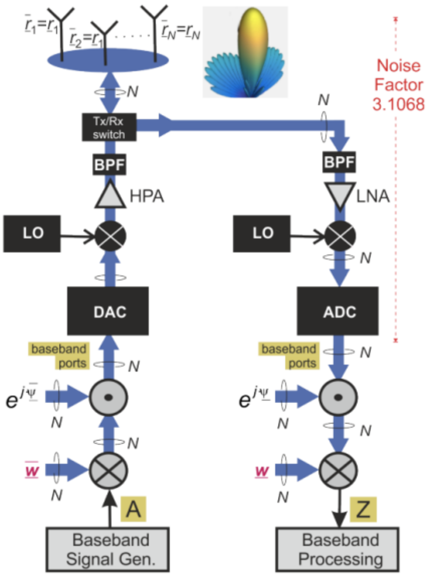


Figure 1: Monostatic Pulse Radar system design

Point A at Tx:

A digital monostatic radar generates a pulsed-signal according to the following parameters:

- amplitude A = 1kV
- clock frequency T_c = 28ns
- pulse duration $T_p = 7*T_c$
- pulse repetition interval RPI = $200*T_p$
- pulse compression = [-1,-1,-1,1,1,-1,1]
- 1 Dwell Time = 8*PRI

The code used to generate the initial signal is found in the pA_to_basebandports.m MATLAB function. This function is used to transmit the generated waveform to the baseband ports of the transmitter Tx.
The generated signal at point A is a 1xsnapshots vector (where in this case, snapshots are equal to 11,2000 since for one Dwell Time there are $200*7*8*T_p = 11{,}200\ T_p$). Hence, at point A: the signal is identified as a 1x11,200 vector. Then, the generated signal is multiplied by the weight vector $\underline{\overline{w}} = 1_{45}$ and thus the input signal becomes a 45x11,200 vector. As shown in figure 1, the vector then is phase shifted by $e^{j\overline{\psi}}$ (explained in Task1) and then reaches the baseband ports of the channel (discussed in Task 2).

The output of this function is defined as the Tx_baseband signal (45x11,200 matrix).

Task 1: Phase Shifters

<u>Vectors of Tx and Rx phase-shifter</u>

This model uses a phased-array radar that operates at a wavelength of λ and employs 45 antennas with an inter-antenna spacing d = $\frac{\lambda}{2}$.

In order to steer both the Tx and Rx main lobes towards a direction θ (azimuth angle) the estimated vectors of the phase-shifter are found by:

$$\underline{\psi}\ (\theta) = \underline{r}^T \mathrm{k}(\theta) \tag{1}$$

where it uses an array of 45 *isotropic* antennas with Cartesian coordinates defined as:

$$\underline{r} = [r_1, r_2, \ldots, r_k, \ldots, r_{45}] = \begin{bmatrix} -22d & -21d & \cdots & +22d \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \end{bmatrix} \epsilon R^{3x45} \quad , \quad \mathrm{d} = \frac{\lambda}{2} \tag{2}$$

Hence, the above equation is simplified for $\underline{r}_y = \underline{r}_z = 0$ and thus $\underline{r}_x$ is defined as:

$$\underline{r}_x = [-22d \quad -21d \quad \cdots \quad 0 \quad \cdots \quad +22d] \tag{3}$$

and the vectors of Tx and Rx phase-shifter are determined by the equation:

$$\underline{\psi}\ (\theta) = [\underline{r}_x \quad \underline{r}_y \quad \underline{r}_z] \frac{2\pi}{\lambda} \begin{bmatrix} cos\theta cos\varphi \\ sin\theta cos\varphi \\ sin\varphi \end{bmatrix} \tag{4}$$

Since it is assumed that *elevation* angle φ = 0° then the above equation is simplified to:

$$\underline{\psi}\ (\theta) = \frac{2\pi}{\lambda} \underline{r}_x cos\theta \tag{5}$$

Using the above information, the following MATLAB is used to calculate the vectors of Tx and Rx phase-shifter for the steering directions 40°,70°,120°:

```
%% Task 1: (a) Phase Shifters
% The following code estimates the vectors of Tx and Rx phase-shift
% for any steering direction theta (defined as theta_steer_angle)

N = 45;                                 % Number of array of antennas
Tc = 28e-9;                             % clock frequency
c = physconst('LightSpeed');
lambda = c*Tc;                          % Wavelength
d = lambda/2;

% Defining r
r = [-22*d,-21*d,-20*d,-19*d,-18*d,-17*d,-16*d,-15*d,-14*d,-13*d,-12*d,-11*d,-10*d,-9*d,-8
         0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
         0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0

theta = 0:1:360;                        % azimuth
phi = -90:1:90;                         % elevation
% Calculation of vector by setting the direction in degrees
theta_steer_angle = 40;                % azimuth angle
phi_steer_angle = 0;                    % elevation angle
k_theta = [cos(deg2rad(theta_steer_angle))*cos(deg2rad(phi_steer_angle));...
          sin(deg2rad(theta_steer_angle))*cos(deg2rad(phi_steer_angle));...
          sin(deg2rad(phi_steer_angle))];
w = exp(-1i*2*pi/lambda*r'*k_theta);
phase_shifters = angle(conj(w));
phase_shifters = rad2deg(phase_shifters);
% The phase shifter vector
phase_shifters_deg = mod(phase_shifters,360);
```

For θ = 40°

1	206.4640
2	344.3520
3	122.2400
4	260.1280
5	38.0160
6	175.9040
7	313.7920
8	91.6800
9	229.5680
10	7.4560
11	145.3440
12	283.2320
13	61.1200
14	199.0080
15	336.8960
16	114.7840
17	252.6720
18	30.5600
19	168.4480
20	306.3360
21	84.2240
22	222.1120
23	0
24	137.8880
25	275.7760
26	53.6640
27	191.5520
28	329.4400
29	107.3280
30	245.2160
31	23.1040
32	160.9920
33	298.8800
34	76.7680
35	214.6560
36	352.5440
37	130.4320
38	268.3200
39	46.2080
40	184.0960
41	321.9840
42	99.8720
43	237.7600
44	15.6480
45	153.5360

For θ = 70°

1	85.6002
2	147.1639
3	208.7275
4	270.2911
5	331.8547
6	33.4184
7	94.9820
8	156.5456
9	218.1092
10	279.6729
11	341.2365
12	42.8001
13	104.3637
14	165.9274
15	227.4910
16	289.0546
17	350.6182
18	52.1819
19	113.7455
20	175.3091
21	236.8727
22	298.4364
23	0
24	61.5636
25	123.1273
26	184.6909
27	246.2545
28	307.8181
29	9.3818
30	70.9454
31	132.5090
32	194.0726
33	255.6363
34	317.1999
35	18.7635
36	80.3271
37	141.8908
38	203.4544
39	265.0180
40	326.5816
41	28.1453
42	89.7089
43	151.2725
44	212.8361
45	274.3998

For θ = 120°

1	180.0000
2	90.0000
3	360.0000
4	270.0000
5	180.0000
6	90.0000
7	360.0000
8	270.0000
9	180.0000
10	90.0000
11	360.0000
12	270.0000
13	180.0000
14	90.0000
15	360.0000
16	270.0000
17	180.0000
18	90.0000
19	360.0000
20	270.0000
21	180.0000
22	90.0000
23	0
24	270.0000
25	180.0000
26	90.0000
27	1.6670e–...
28	270.0000
29	180.0000
30	90.0000
31	3.3340e–...
32	270.0000
33	180.0000
34	90.0000
35	2.4566e–...
36	270.0000
37	180.0000
38	90.0000
39	6.6680e–...
40	270.0000
41	180.0000
42	90.0000
43	8.8439e–...
44	270.0000
45	180.0000

Using the above phase-shifters, four types of **array patterns** are plotted below:

1. Linear
2. Polar
3. 3D Cartesian
4. 3D Spherical

For: ***θ = 40°***

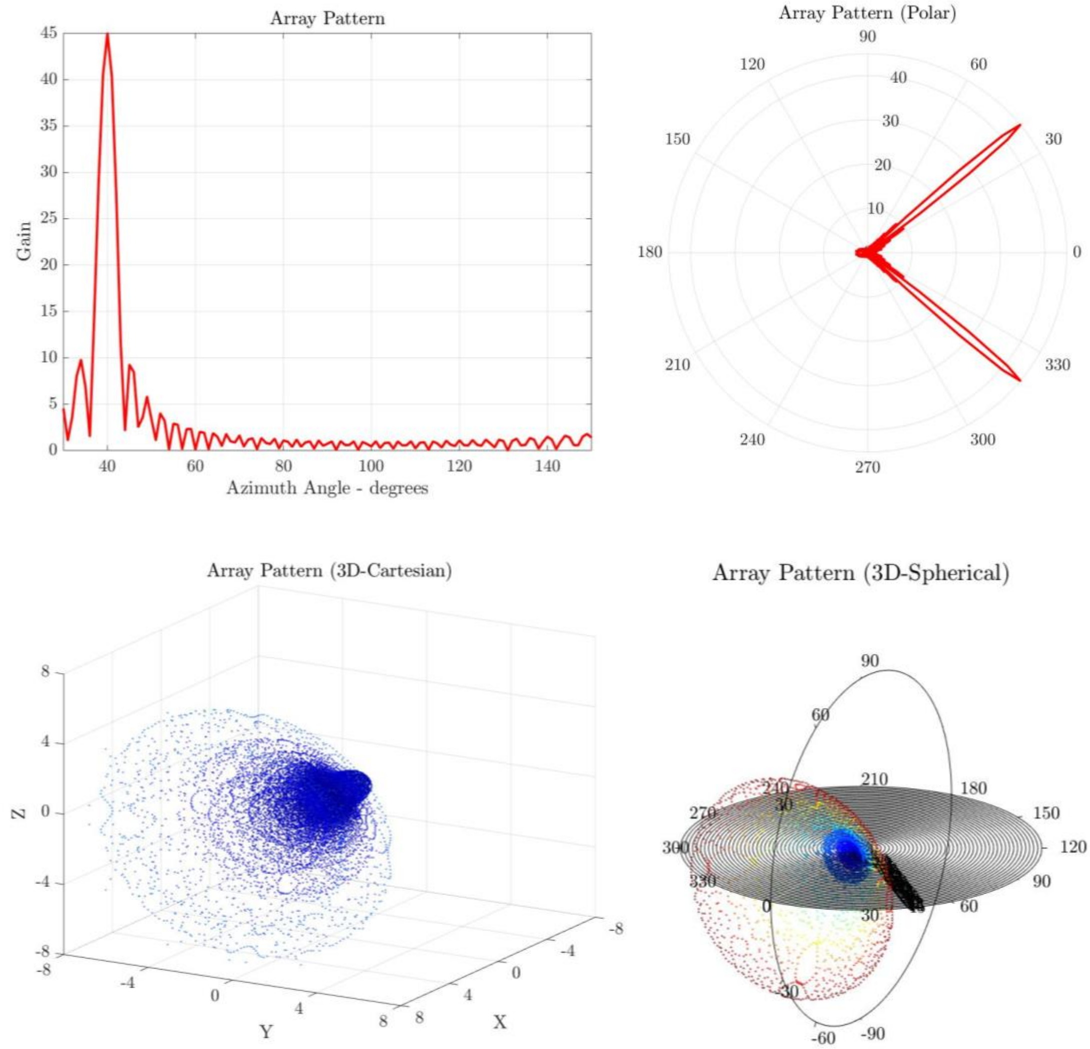
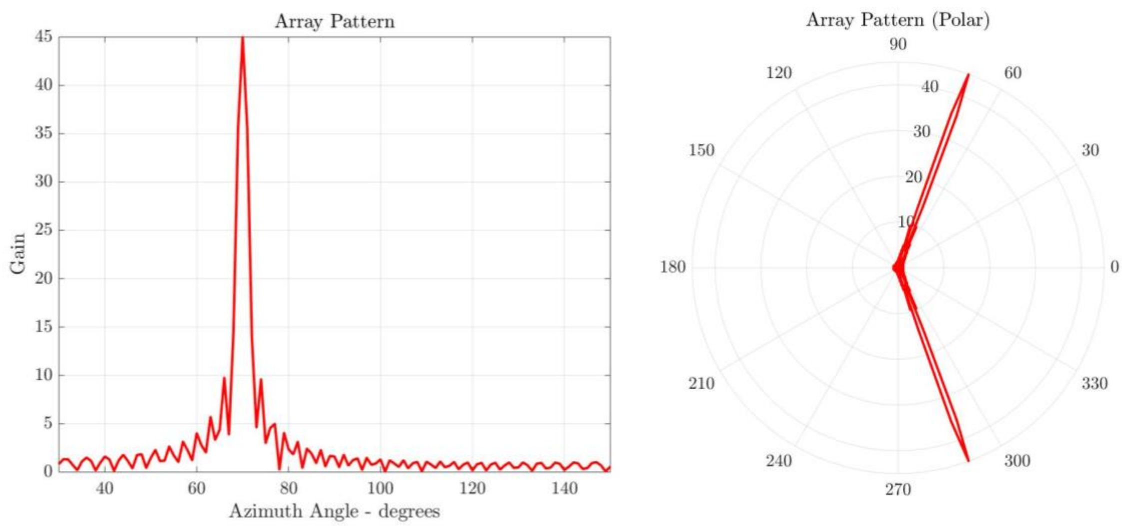


Figure 2: Array Patterns for azimuth angle = 40°

For: ***θ = 70°***



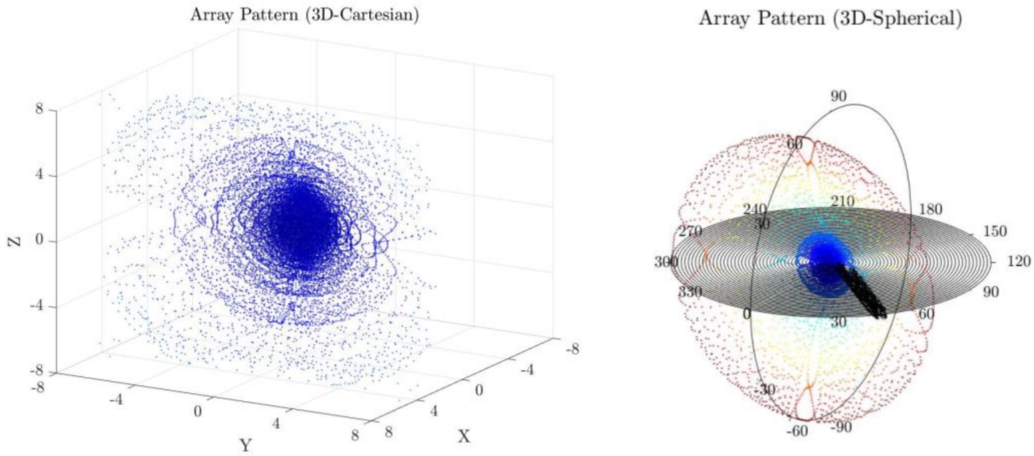


Figure 3: Array Patterns for azimuth angle = 70°

For: ***θ = 120°***

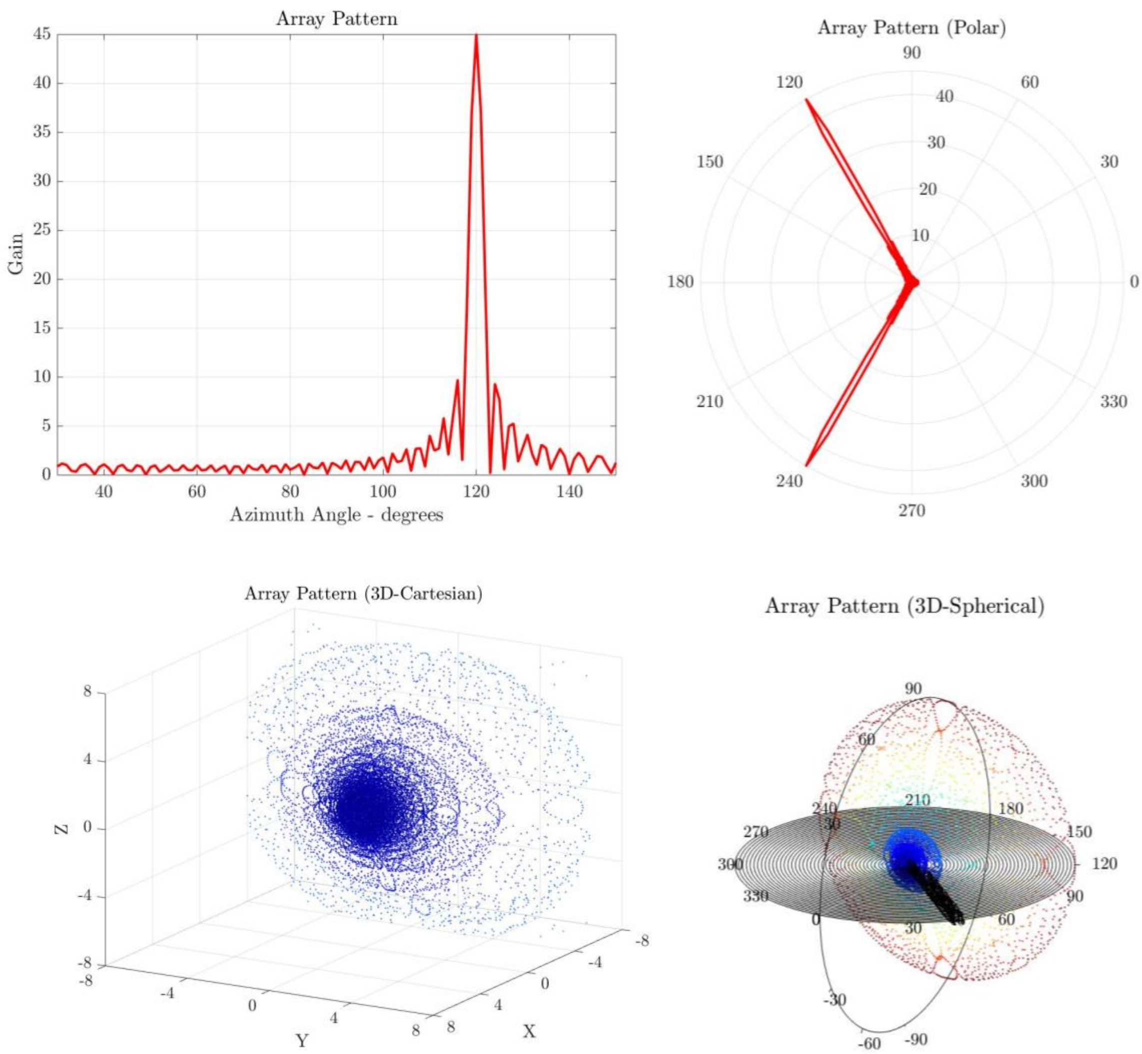


Figure 4: Array Patterns for azimuth angle = 120°

Task 2: MATLAB backscatter modelling function

The following figure illustrates the channel of the *Monostatic Pulse Radar* (figure 1).

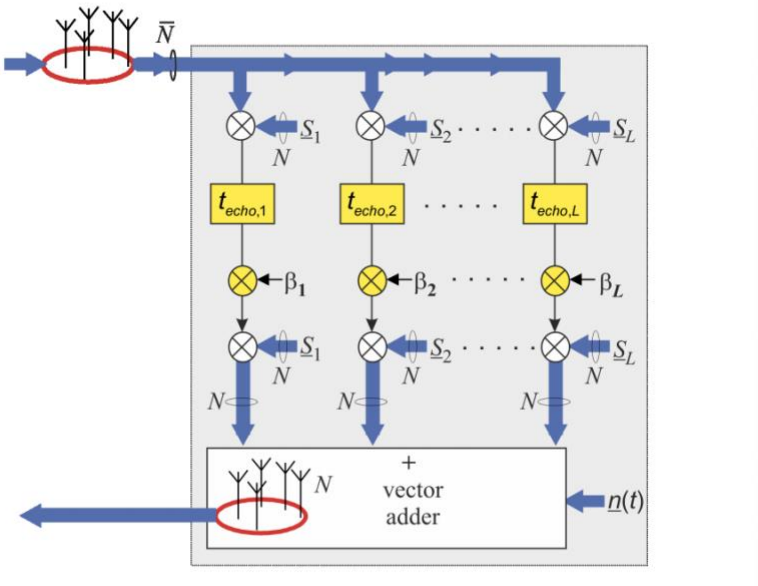


Figure 5: MIMO backscatter multi-target modelling

In this task, we create the MATLAB function that generates the backscatter data for a given steering direction. This function is found in the MATLAB file named as backscatterdata.m and is defined by two input parameters:

1. **Tx_baseband**: which is defined in the pA_to_basebandports function with input parameter: theta_steer_angle (explained in Radar Description section)
2. **Targets**: this parameter can take values from to 0 to 3

The input parameter Tx_baseband is a 45x11,200 vector:
- **Each row:** represents the signal of each Tx antenna array antenna
- **Each column:** represents the signal-vector sent for transmission at a particular time instant

The function is created according to figure 5. Testing of the function is found in Appendix A.

First step: Define the parameters of each target

The number of irritations of the first *for loop* is determined by how many targets are detected (depending on the Task).

	Target-1 (constant)	**Target-2 (complex)**	**Target-3 (complex)**
direction, θ	$\theta_1 = 40°$	$\theta_2 = 70°$	$\theta_3 = 120°$
range, R	R_1 = 2km	R_2 = 3km	R_3 = 2.5km
target, mean RCS	$RCS_1 = 1\text{m}^2$	$RCS_2 = 5\text{m}^2$	$RCS_3 = 4.5\text{m}^2$
complexity	constant RCS	scatters of similar amplitudes	scatters with one much larger than the other
model for RCS	constant	Sweling1&2	Swerling3&4

Table 1: Target parameters

A second *for loop* is used in order to assign an RCS vector for each PRI. Hence, for one dwell time the RCS vector would have dimensions 1x11,200.

<u>Second step:</u> Tx array manifold vectors

The input signal Tx_baseband is multiplied by the Tx-array manifold vector:

$$\underline{S}_1 = e^{j\psi(\theta)} \tag{6}$$

where $\psi(\theta)$ is defined in equation (5)

The Tx-array manifold vector is a complex vector of dimensions 45x1. Hence, the multiplication of the manifold vector's transpose with the input signal results to a 1x11,200 vector.

<u>Third step:</u> Define delay

According to figure 5, the resulted signal is delayed by *delay* = $\frac{t_{echo}}{T_c}$ units where:

$$t_{echo} = \frac{2R}{c} \quad , \quad c = \text{speed of light} \tag{7}$$

The signal is shifted by *delay* units to the right at the end of the loop.

<u>Fourth step:</u> Path attenuation

The signal is attenuated by the factor β:

$$\beta = \frac{P_{Tx} G_{Tx} G_{Rx}}{(4\pi)^3} \cdot \frac{\lambda}{R^2} \cdot \sqrt{RCS} \tag{8}$$

Since RCS is a 1x11,200 matrix, beta will also be a matrix with the same dimensions. Hence, the signal is multiplied *elementwise* with beta.

<u>Fifth step:</u> Rx array manifold vectors

The signal is multiplied with the conjugate of the S factor given by equation (6). The conjugate of S is a 45x1 vector while the signal has dimensions of 1x11,200 as explained above. Hence, the resulted signal at the Rx baseband ports is a 45x11,200 vector.

<u>Sixth step:</u> Add the effect of noise

The total effect of different noise sources is presented as a single noise source at the Rx antenna with power[1]:

$$\sigma^2 = k_B \cdot T_0 \cdot F_n \cdot B \tag{9}$$

where k_B = Boltzmann constant, T_0 = 290K temperature, F_n = Noise Figure of the Rx-subsystem (defined in Figure 1) and B = bandwidth.

[1] Notes

Task 3: 1st scan - no targets

(a) Generate the noise samples/snapshots at the baseband ports of the Rx (at the output of ADC in figure 1).

We assume the number of targets to be equal to 0 and set a random theta angle in order to generate the noise samples/snapshots.

The code for the generation of the noise snapshots is found in Task3.m file. It is evident that the noise samples/snapshots at the baseband ports of the Rx is a 45x11,200 vector:

Workspace

Name ▲	Value
noise_power	1.8754e-11
noise_sq	*1x11200 double*
Rx_baseband	*45x11200 compl...*
theta_steer_an...	30
Tx_baseband	*45x11200 compl...*
z	*1x11200 double*
z_out	*1x11200 comple...*

(b) Plot the magnitude (Volts) of noise snapshots for one Dwell-time.

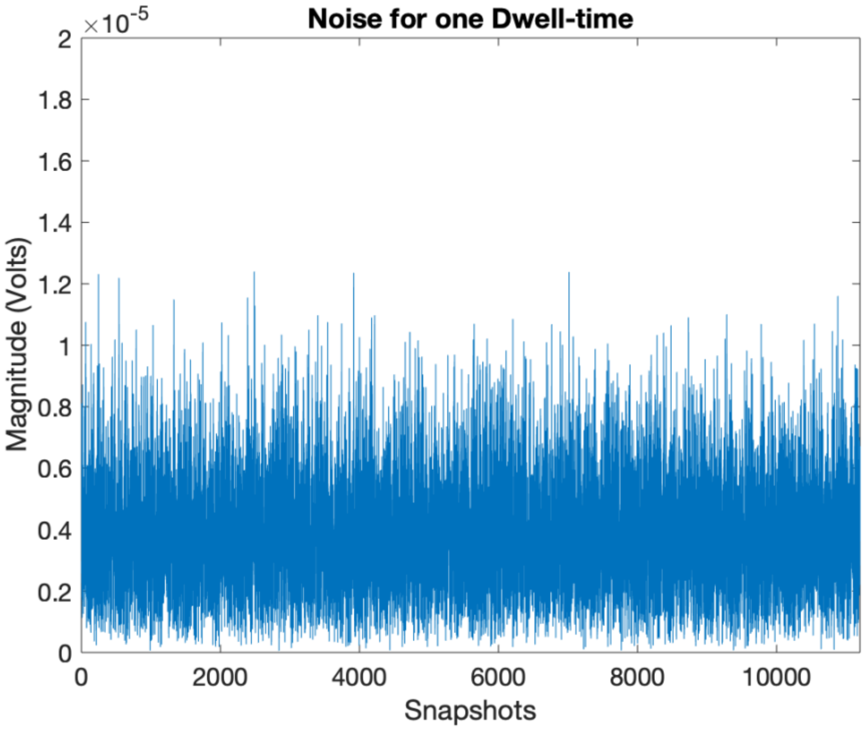


Figure 6: Noise for one dwell-time at point Z (11,200 snapshots)

(c) Estimate and plot the pdf of the noise data samples.

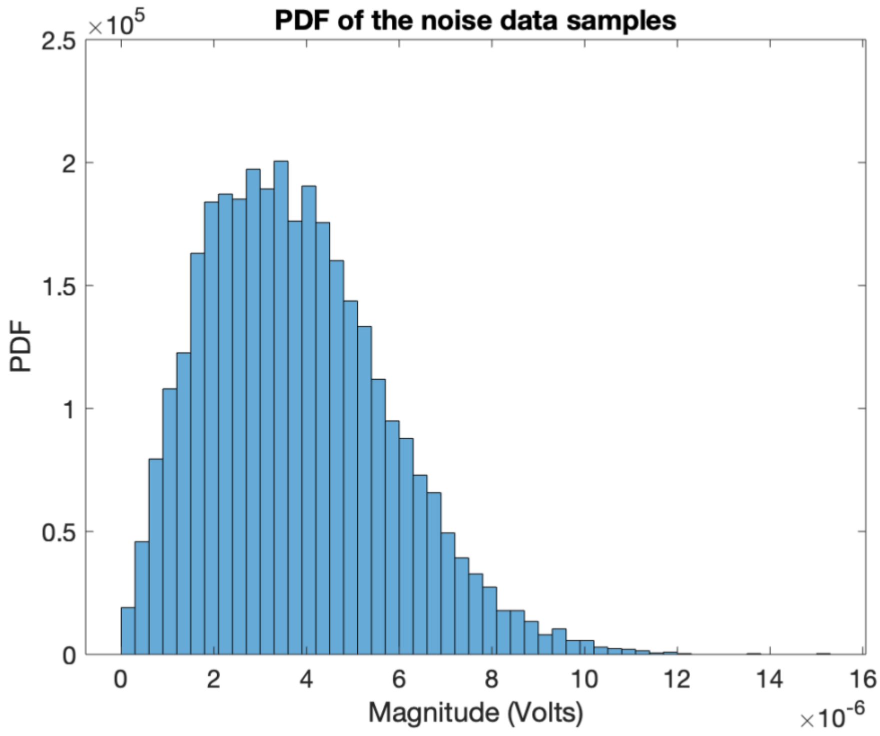


Figure 7: Probability density function of the noise data sample at point Z

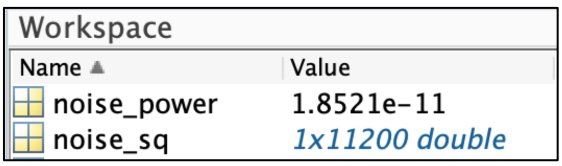
The above figure is plotted by using the MATLAB command histogram.

(d) Estimate the noise power at point-Z in figure 1.

The noise power at point-Z is estimated by the square of the noise vector. However, this results to a 1x11,200 matrix, and thus we estimate the mean of the vector. This is shown by the following code:

```
noise_sq = z.^2;
noise_power = mean(noise_sq);
```

Hence, the noise power estimate: P_{noise} =**1.8521** x 10^{-11} W



Task 4: 2nd scan – one target

(a) Assuming that the Target-1 parameters are known, generate synthetic backscatter-data for this scan at the baseband ports.

The parameters for each target are defined inside the backscatterdata.m function. Since we are interested in Target-1 we define the parameters targets = 1.

The backscatter-data at the Rx baseband ports are a matrix of dimensions 45x11,200 (N x snapshots). The generation of the backscatter-data is defined as backscatterData, a 121x1 cell where each cell contains a 45x11,200 matrix which is the signal at the baseband ports of the receiver Rx. Each cell represents an azimuth angle from 30 to 150. For example, cell 1 represents the received signal at the baseband ports for direction angle θ = 30°, cell 2: θ = 31°, ... , cell 121: θ = 150°.

(b) Plot the backscatter-data at point-Z for the dwell-time that corresponds to the direction of the Target-1.

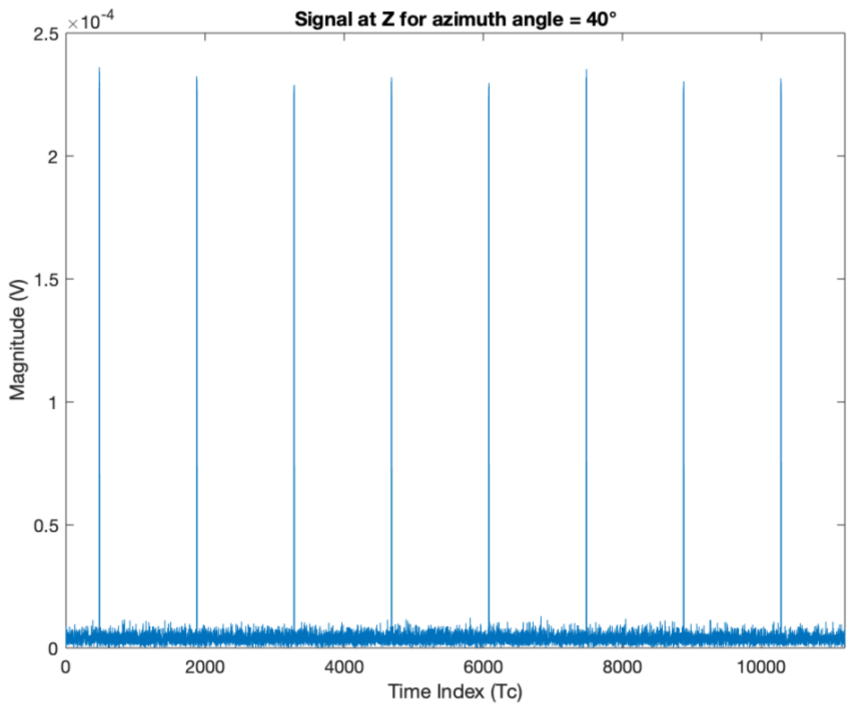


Figure 8: Backscatter-data at point Z for azimuth angle = 40°

In the above figure, it is evident that the magnitude of signal has a peak around 2.2 x 10^{-4} Volts. The noise appears to be very small compared to the magnitude of the signal at point Z. This observation can also be verified by comparing figures 6 (Task 3) and figure 8 since the noise signal has a much smaller amplitude. The following figure shows the same data but instead of the MATLAB command plot we use semilogy.

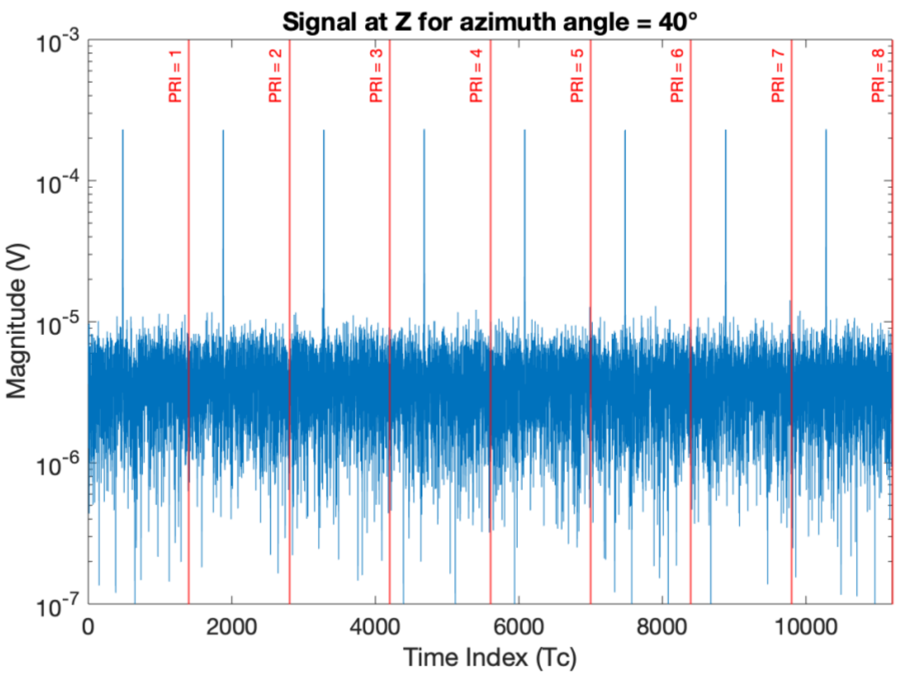


Figure 9: Backscatter-data at point Z for azimuth angle = 40° using MATLAB command *semilogy*

(c) Then forget that the parameters of the Target-1 are known. Using only your generated backscatter random numbers at point-Z, detect and estimate the parameters of this target.

Using the generated backscatter random numbers at point Z we detect the following signal at steering direction = 40° by setting the expected threshold.

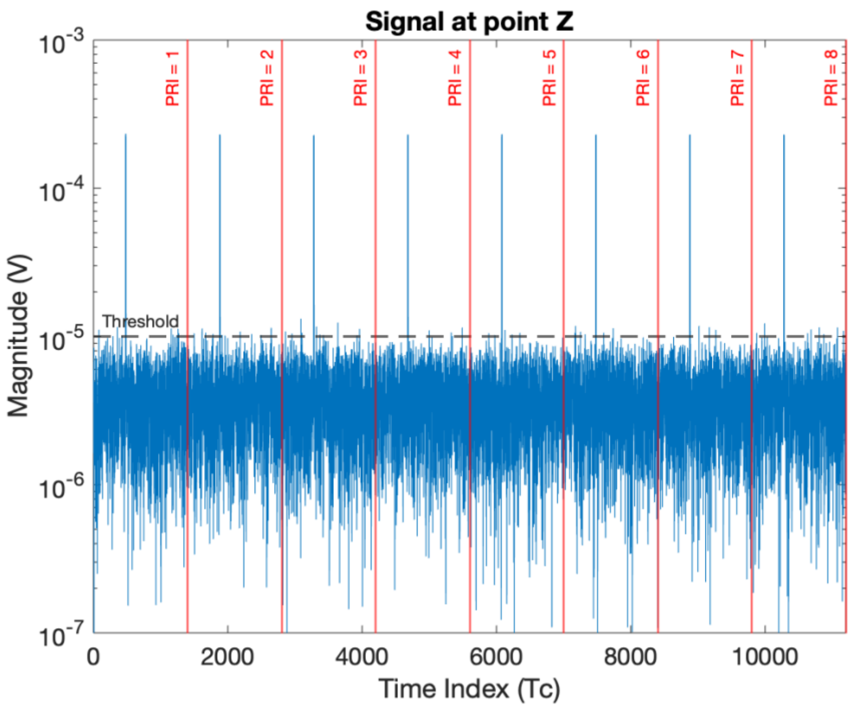


Figure 10: Random backscatter-data at point Z

The code for this Task is found in the MATLAB file named Task4.m. In order to ensure that the detection is accurate we set a threshold by taking into consideration the probability density function of noise and the probability of false alarm. The detection code uses pulse compression (explained in the MATLAB) and full correlation between the received signal and the PN code [-1 -1 -1 1 1 -1 1], as a matched filter. The output of the code is:

```
Command Window
Target direction: 40 degrees
Range: 1.999200e+03 m = 1999 m
RCS: 1.019720e+00 m^2  = 1 m^2
Range index: 69th
>>
```

Detection of range (after pulse compression and averaging) is shown below:

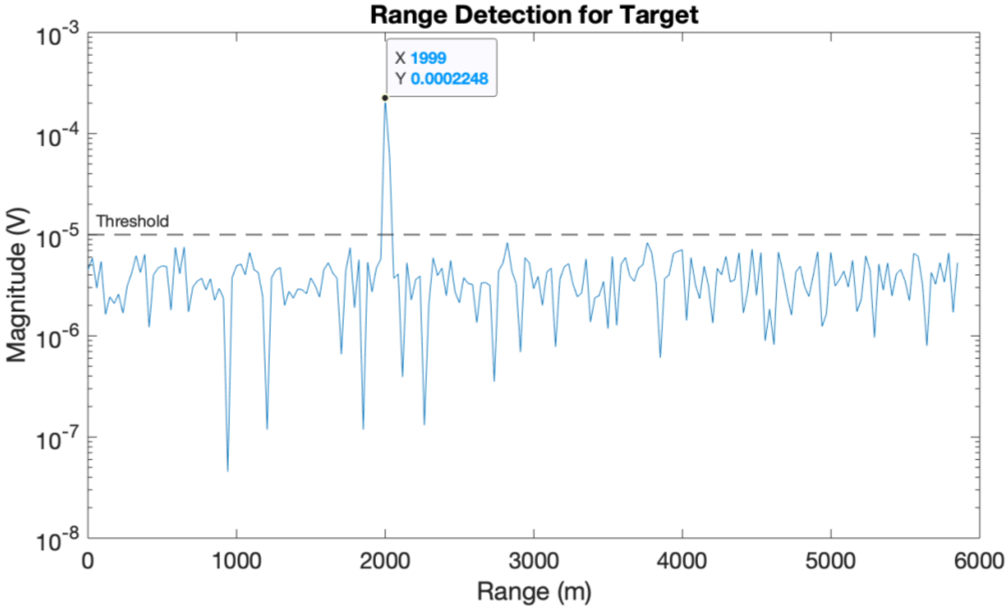


Figure 11: Range detected for the first target

Using equation (7), t_{echo} estimate is calculated as $t_{echo} = \frac{2R}{c} = \frac{2 \cdot 1999}{3 \cdot 10^8} = 1.33 \cdot 10^{-5}$ s.

Comments

A full correlation is used between the received signal and the PN code to avoid the effects of partial convolution when using the matched filter coding: conv(z,pn). For the method that was used, first we determine the sequence with the largest correlator and then use this sequence to find the Rx correlator. The construction of the Rx correlator uses elementwise multiplication and summation as explained in the corresponding MATLAB file.

In this task Target-1 was detected. According to Table 1, the parameters of this target are: azimuth angle = 40, Range = 2000m, RCS = 1 m^2. Hence, the program has effectively detected all the parameters of the target.

Task 5: 3rd scan – two targets

(a) Assuming that the Target-1 and Target-2 parameters are known, generate synthetic backscatter-data for this scan at the baseband ports.

The parameters for each target are defined inside the backscatterdata.m function. Since we are interested in Target-1 and Target-2 we define the parameters targets = 2. The generation of the backscatter-data is defined as backscatterData, where the data is stored as a 121x1 cell: each cell contains a 45x11,200 matrix which is the signal at the baseband ports of the receiver Rx for steering directions from 30 to 150.

(b) Plot the backscatter-data at point-Z for the dwell-time that corresponds to the directions of the two targets.

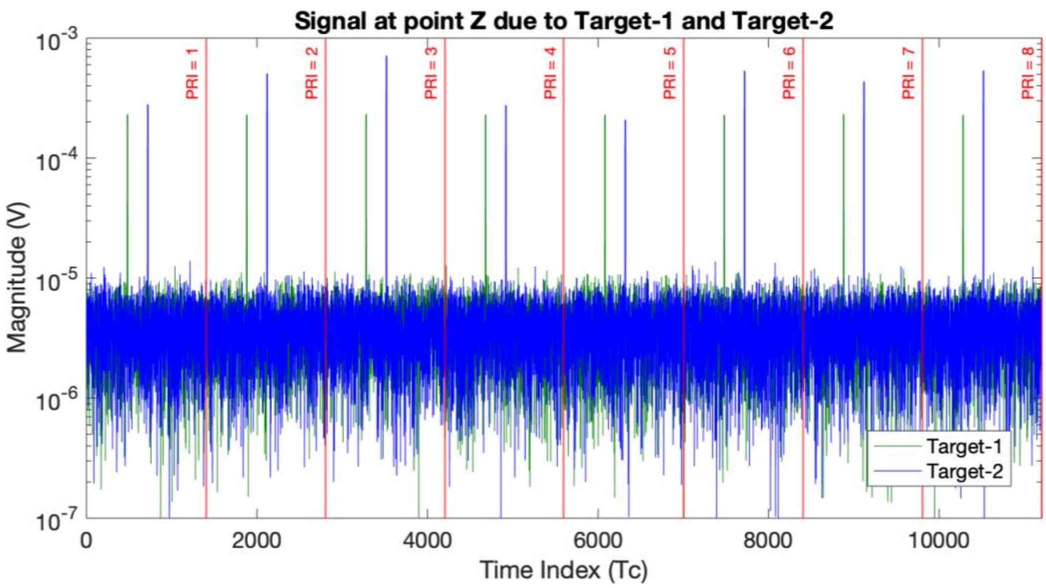


Figure 12: Signal at point Z due to both targets (at different steering directions)

In the below figure, it is evident that the magnitude of signal varies since the RCS value of Target-2 is random. Once again, the noise is considered small compared to the maximum amplitude of the signal at point Z. **Left figure**: steering direction = 40°,
Right figure: steering direction = 70°

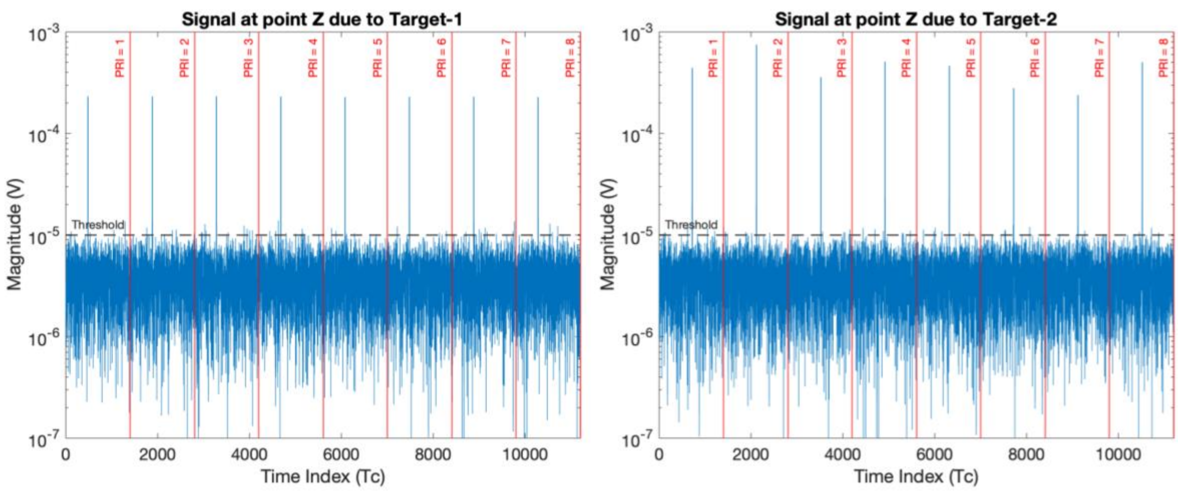


Figure 13: Signal at point Z at the two steering directions

(c) Then forget that the parameters of the two targets are known. Using only your generated backscatter random numbers at point-Z, detect and estimate the parameters of these two targets.

The code for this Task is found in the MATLAB file named Task5.m. In order to ensure that the detection is accurate we set a threshold by taking into consideration the probability density function of noise and the probability of false alarm. The detection code uses pulse compression (explained in the MATLAB file) and full correlation between the received signal and the PN code [-1 -1 -1 1 1 -1 1]. We know that for this scan two targets are present and thus a loop is used and is defined by 1:1:targets, where targets = 2.

The output of the code is:

```
Command Window
Target: 1
Target direction: 40 degrees
Range: 1.999200e+03 m = 1999 m
RCS average: 1.031534e+00 m^2  = 1 m^2
Range index: 69th

Target: 2
Target direction: 70 degrees
Range: 2.998800e+03 m = 2999 m
RCS average: 4.990443e+00 m^2  = 5 m^2
Range index: 103th

fx >>
```

Using the generated backscatter random data at point-Z, the following signal is received due to Target-1 and Target-2. The two signals are separated in order to effectively detect each target.

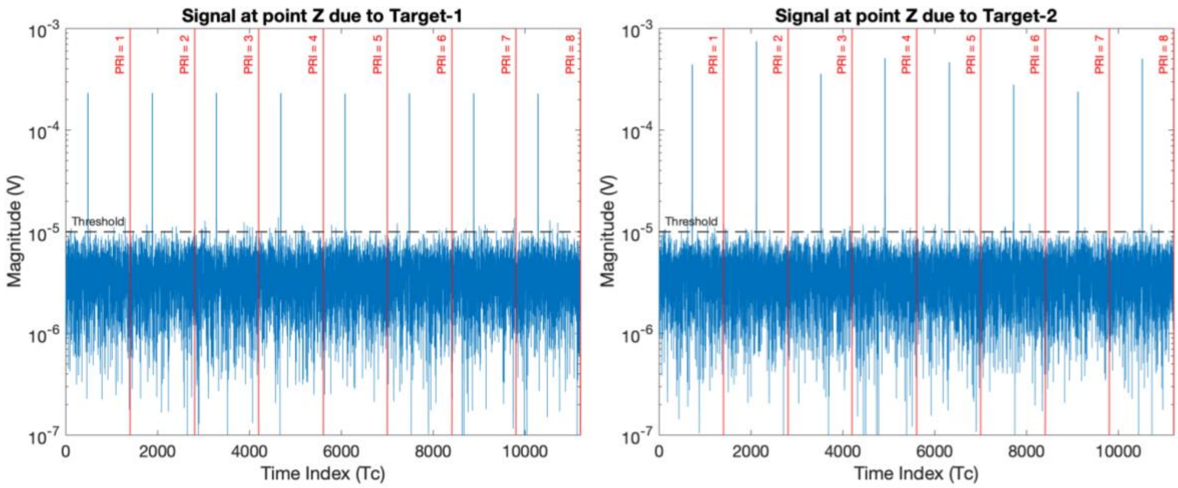


Figure 14: Random backscatter-data at point Z due to the two targets

Detection of range (after pulse compression and averaging) for Target-1 is shown below:

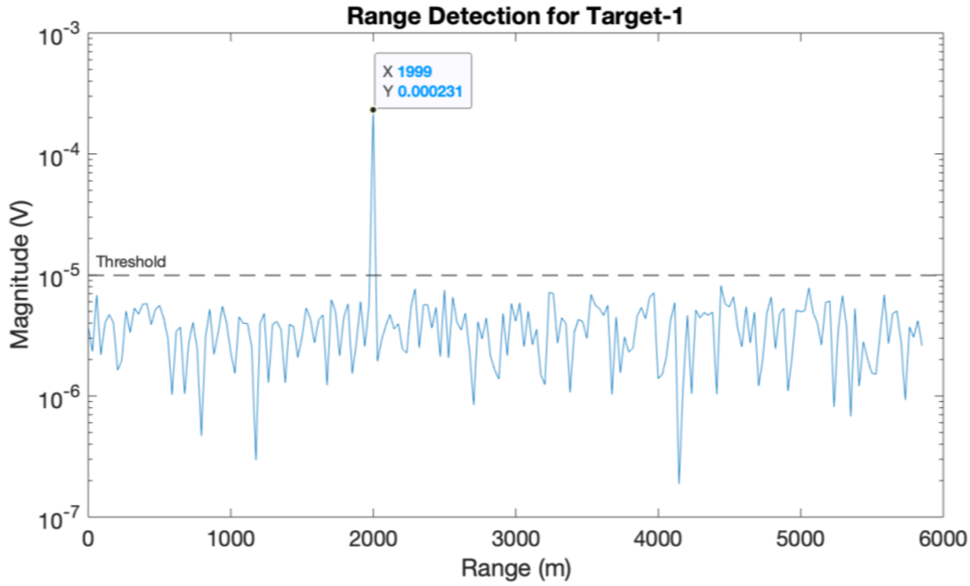


Figure 15: Range detected for the first target

Using equation (7), t_{echo} estimate is calculated as $t_{echo} = \frac{2R}{c} = \frac{2 \cdot 1999}{3 \cdot 10^8} = 1.33 \cdot 10^{-5}$ s.

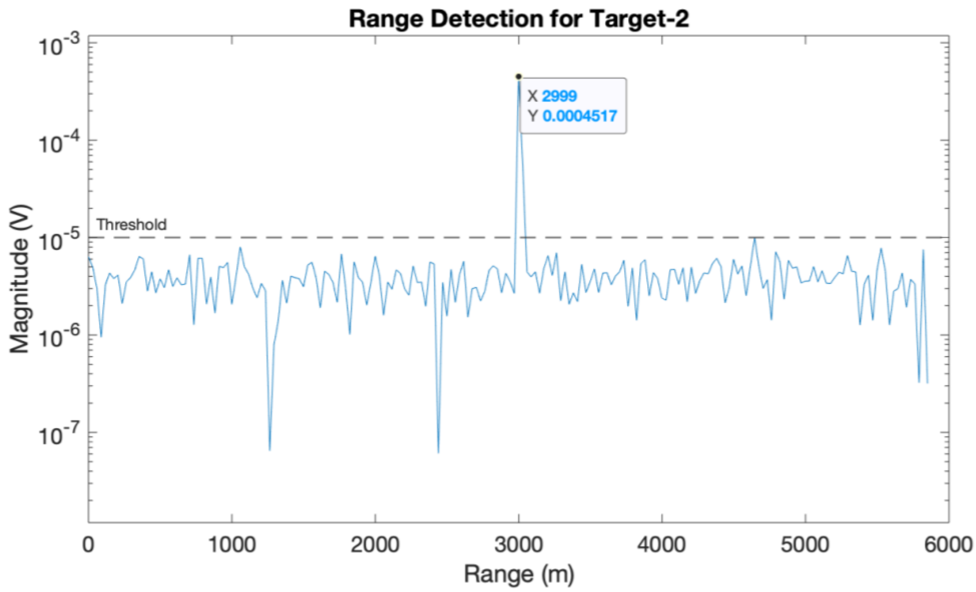


Figure 16: Range detected for the second target

Using equation (7), t_{echo} estimate is calculated as $t_{echo} = \frac{2R}{c} = \frac{2 \cdot 2999}{3 \cdot 10^8} = 1.99 \cdot 10^{-5}$ s.

Comments

The code effectively estimates the parameters of the two targets as shown in the above figures and at the output of the code (command window). However, it is important to note that the estimation of RCS for Target-2 is a bit unstable. More specifically, during testing of the code Target-2's RCS estimate takes values equal from 2 to 5 m^2. This is due to a small uncertainty added when using the Rx correlator.

Task 6: 4rth scan – three targets

(a) Assuming that the Target-1 and Target-2 and Target-3 parameters are known, generate synthetic backscatter-data for this scan at the baseband ports.

The parameters for each target are defined inside the backscatterdata function. Since we are interested in Target-1, Target-2 and Target-3 we define the parameter targets = 3. The signal at the Rx baseband ports are a matrix of dimensions 45x11,200 (N x snapshots). The generation of the back-scatter data is defined as backscatterData (a 121x1 cell) in Task6.m MATLAB file.

(b) Plot the backscatter-data at point-Z for the dwell-time that corresponds to the directions of the three targets.

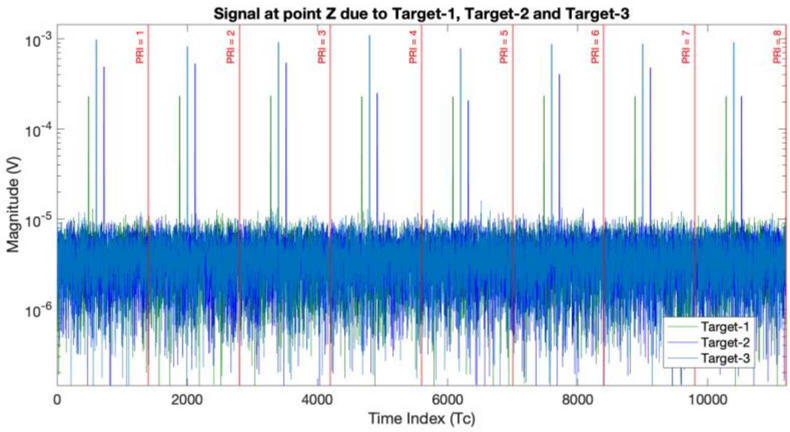


Figure 17: Signal at point Z due to the three targets

In the above figure, it is evident that the magnitude of signal varies since the RCS values of Target-2 and Target-3 are random in nature.

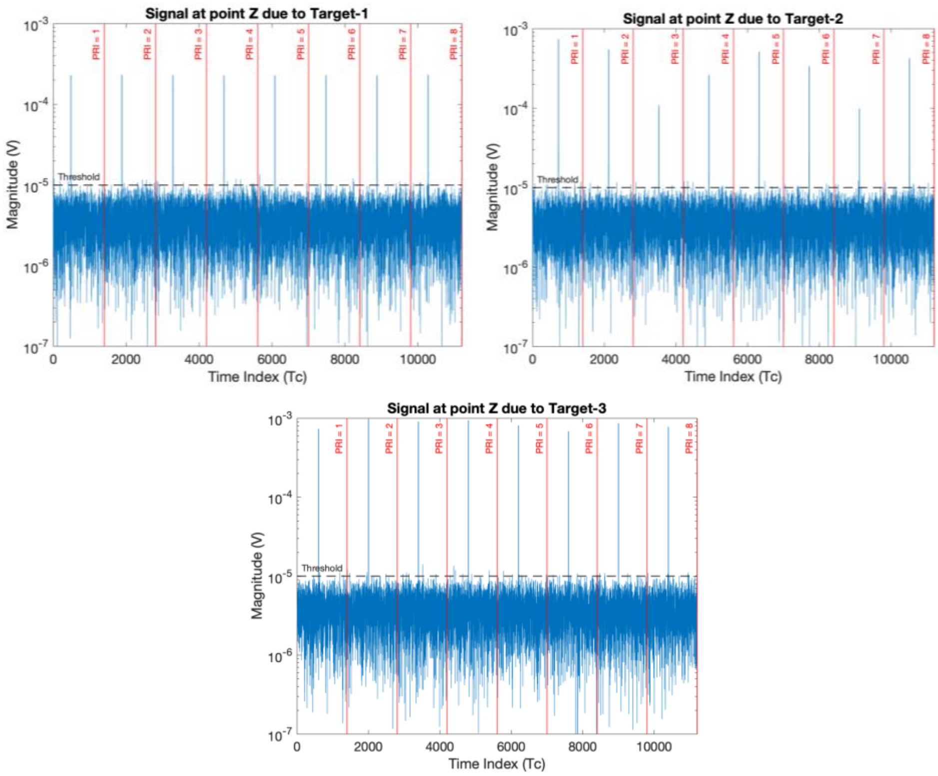


Figure 18: Signal at point Z for the three steering directions: θ = {40°, 70°, 120°}

(c) Then forget that the parameters of the two targets are known. Using only your generated backscatter random numbers at point-Z, detect and estimate the parameters of these three targets.

The code for this Task is found in the MATLAB file named Task6.m. In order to ensure that the detection is accurate we set a threshold by taking into consideration the probability density function of noise and the probability of false alarm. The detection code uses pulse compression (explained in the MATLAB) and full correlation between the received signal and the PN code [-1 -1 -1 1 1 -1 1]. We know that for this scan three targets are present and thus a loop is used and is defined by 1:1:targets, where targets = 3.

The output of the code is:

```
Command Window
Target: 1
Target direction: 40 degrees
Range: 1.999200e+03 m = 1999 m
RCS average: 9.800990e-01 m^2  = 1 m^2
Range index: 69th

Target: 2
Target direction: 70 degrees
Range: 2.998800e+03 m = 2999 m
RCS average: 5.175135e+00 m^2  = 5 m^2
Range index: 103th

Target: 3
Target direction: 120 degrees
Range: 2499 m = 2499 m
RCS average: 4.582478e+00 m^2
Range index: 86th

fx >>
```

Using the generated backscatter random data at point-Z, the following signal is received due to Target-1, Target-2 and Target-3. The signals occur at three different steering directions: θ = {40°, 70°, 120°}.

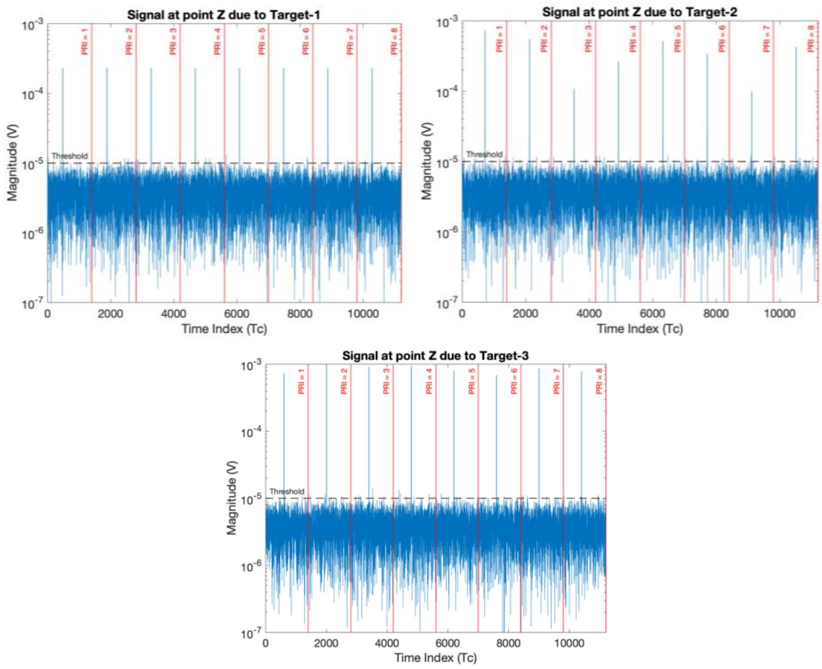


Figure 19: Random backscatter-data at point Z at the three steering directions

Detection of range (after pulse compression and averaging) for Target-1 is shown below:

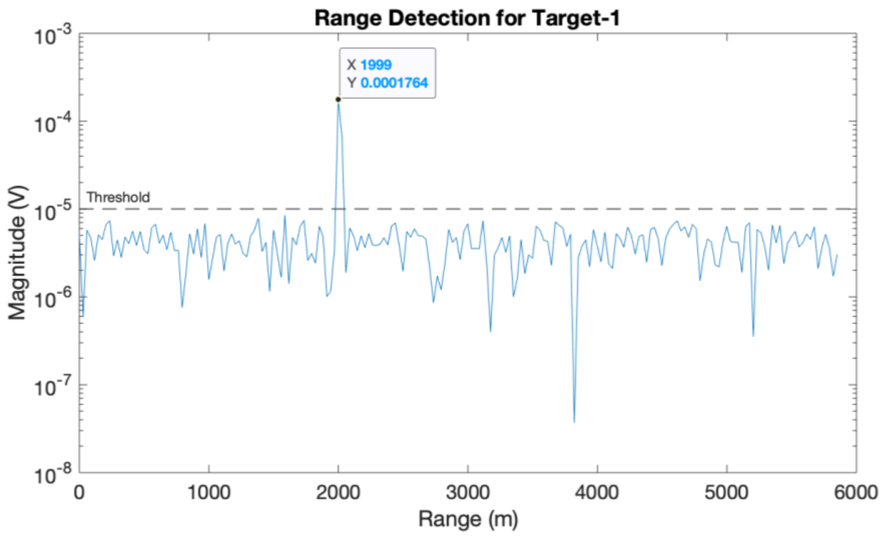


Figure 20: Range detected for the first target

Using equation (7), t_{echo} estimate is calculated as $t_{echo} = \frac{2R}{c} = \frac{2 \cdot 1999}{3 \cdot 10^8} = 1.33 \cdot 10^{-5}$ s.

Detection of range (after pulse compression and averaging) for Target-2 is shown below:

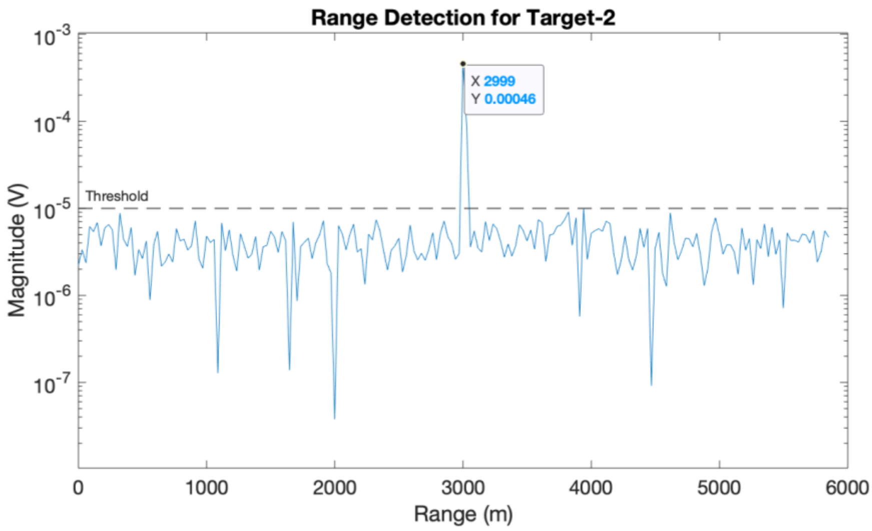


Figure 21: Range detected for the second target

Using equation (7), t_{echo} estimate is calculated as $t_{echo} = \frac{2R}{c} = \frac{2 \cdot 2999}{3 \cdot 10^8} = 1.99 \cdot 10^{-5}$ s.

Detection of range (after pulse compression and averaging) for Target-3 is shown below:

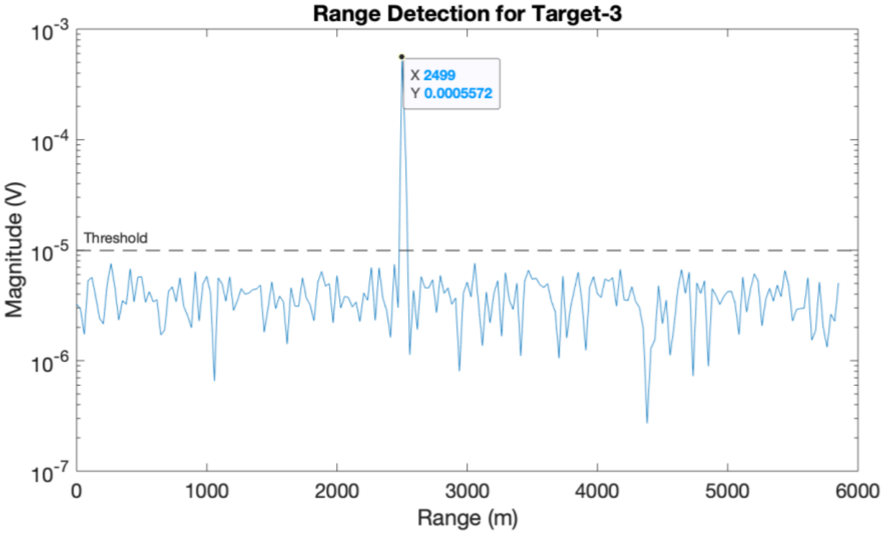


Figure 22: Range detected for the third target

Using equation (7), t_{echo} estimate is calculated as $t_{echo} = \frac{2R}{c} = \frac{2 \cdot 2499}{3 \cdot 10^8} = 1.66 \cdot 10^{-5}$ s.

Comments

The code effectively estimates the parameters of the three targets as shown in the above figures and the command window. However, that the estimation of RCS for Target-2 and Target-3 is a bit unstable. More specifically, during testing of the code Target-2's RCS estimate takes values equal from 2 to 5 m^2 and Target-3's RCS estimate from 3 to 7. This is due to a small uncertainty added when using the Rx correlator.

Task 7: Radar data – multi-target detection / parameter estimation

The data file that is used is called BackscatterData.mat. The data is a 121x1 cell where each cell contains a 45x11,200 matrix which is the signal at the baseband ports of the receiver Rx. Each cell represents an azimuth angle from 30 to 150. For example, cell 1 represent the received signal at the baseband ports for direction angle $\theta = 30°$, cell 2: $\theta = 31°$, ... , cell 121: $\theta = 150°$.

A number of targets are detected at the angle $\theta = 142°$. The signal at point Z for steering direction equal to 142° is shown below.

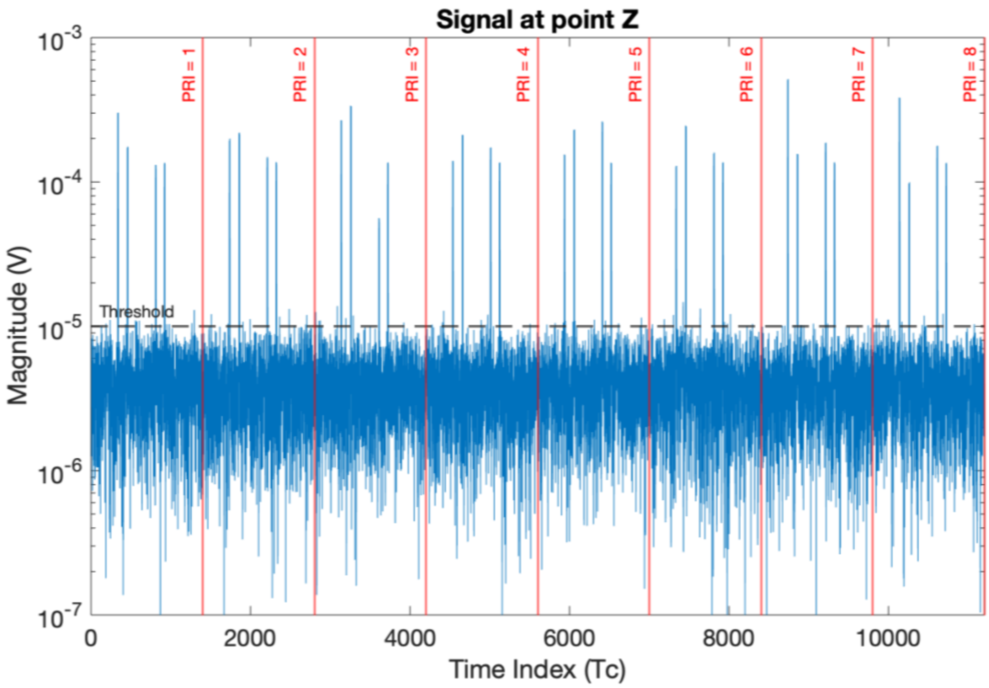


Figure 23: Signal detected at point Z

It is evident in Figure 23 that 4 peaks occur at each PRI. This means that 4 targets are present at this azimuth angle but at a different distance from the radar.

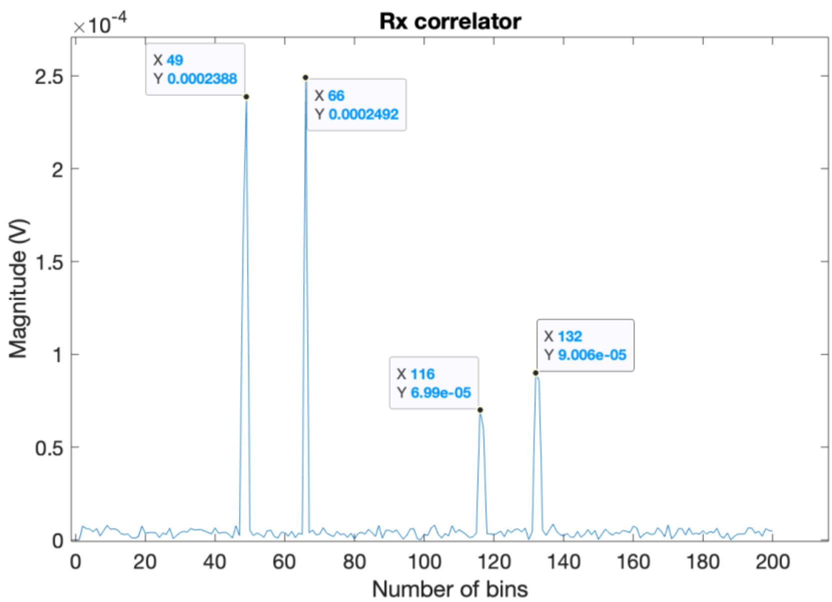


Figure 24: Signal Z after the Rx correlator

In order to ensure accurate detectability, we use an Rx correlator (defined by the full correlation between the PN code and the received signal Z). The output of the correlator is shown in figure 24 above.

Detection of range (after pulse compression and averaging) for the targets is shown below:

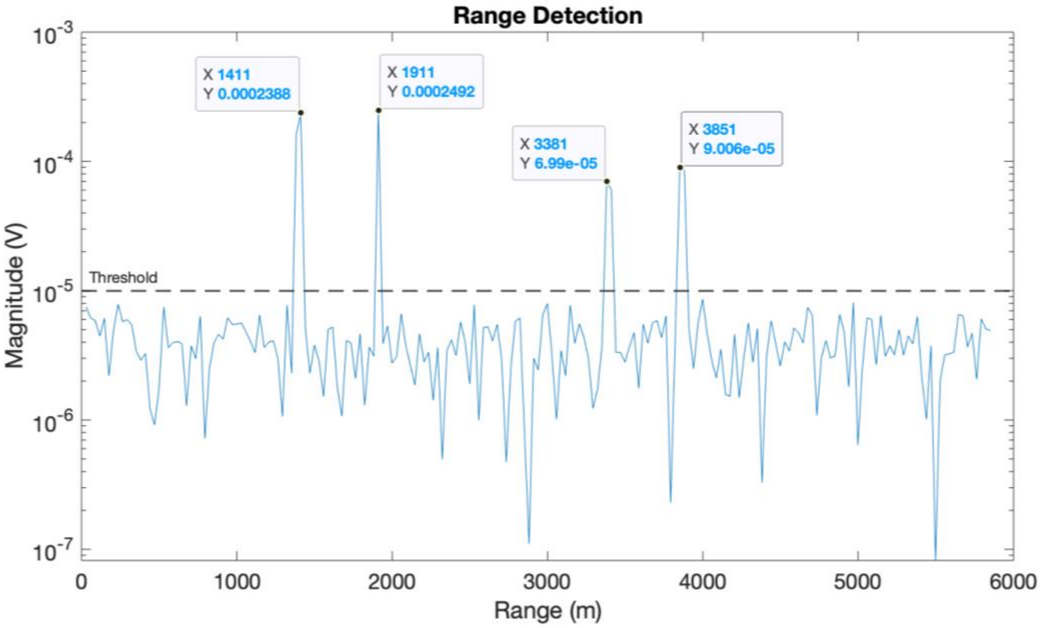
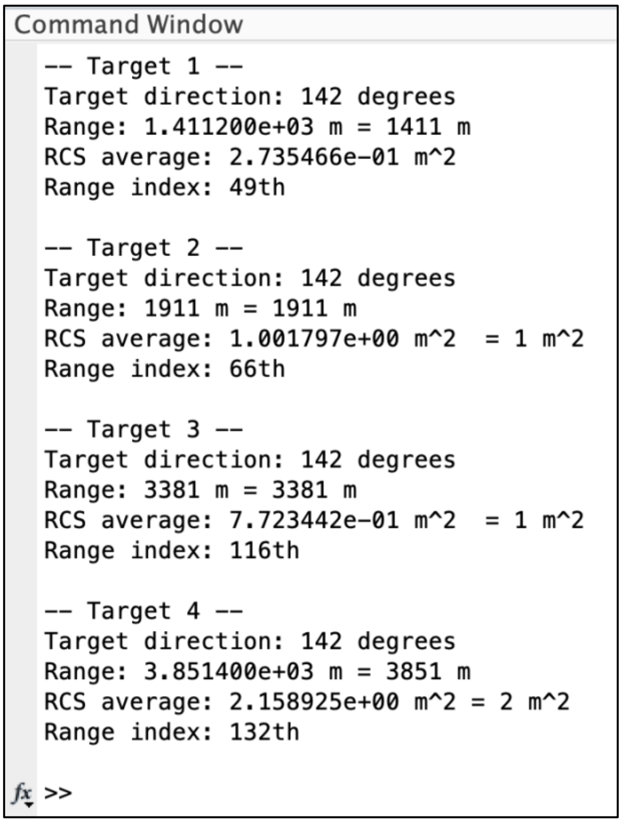


Figure 25: Range detection for the four targets

The output of the code is:



```
Command Window
-- Target 1 --
Target direction: 142 degrees
Range: 1.411200e+03 m = 1411 m
RCS average: 2.735466e-01 m^2
Range index: 49th

-- Target 2 --
Target direction: 142 degrees
Range: 1911 m = 1911 m
RCS average: 1.001797e+00 m^2  = 1 m^2
Range index: 66th

-- Target 3 --
Target direction: 142 degrees
Range: 3381 m = 3381 m
RCS average: 7.723442e-01 m^2  = 1 m^2
Range index: 116th

-- Target 4 --
Target direction: 142 degrees
Range: 3.851400e+03 m = 3851 m
RCS average: 2.158925e+00 m^2 = 2 m^2
Range index: 132th

fx >>
```

Appendix A

An initial testing was also conducted on the backscatterdata function (for Task 2) by defining the following parameters:

- delay = 0
- betas = 1
- targets = 1

By setting the above parameters to these values it is expected to receive a signal with magnitude equal to 1000*45*45 ≈ 2 x 10^6.

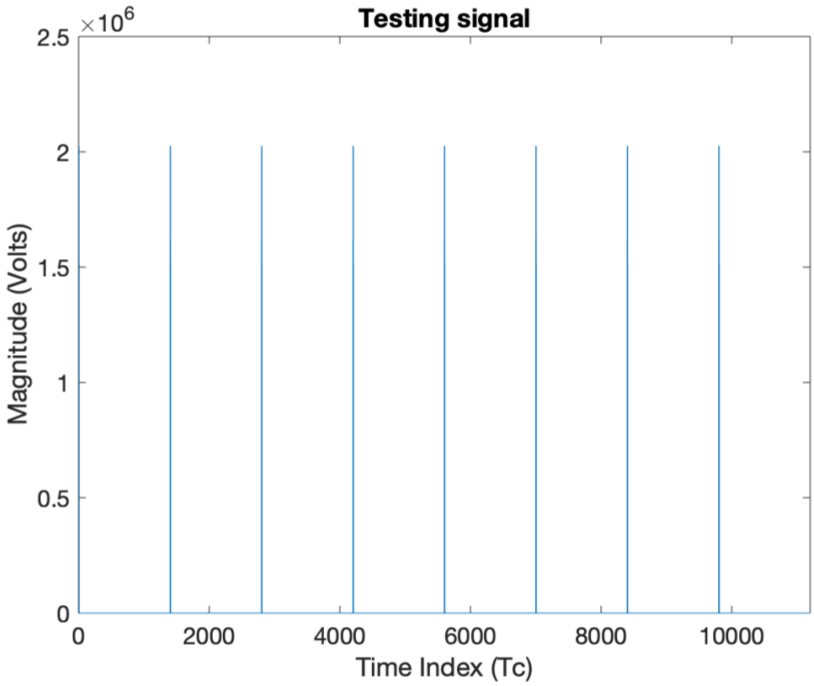


Figure 26: Signal at point Z for delay = 0 and betas = 1

Hence, the testing confirms that the baskscatterdata function works as expected.