Marouen Azzouz

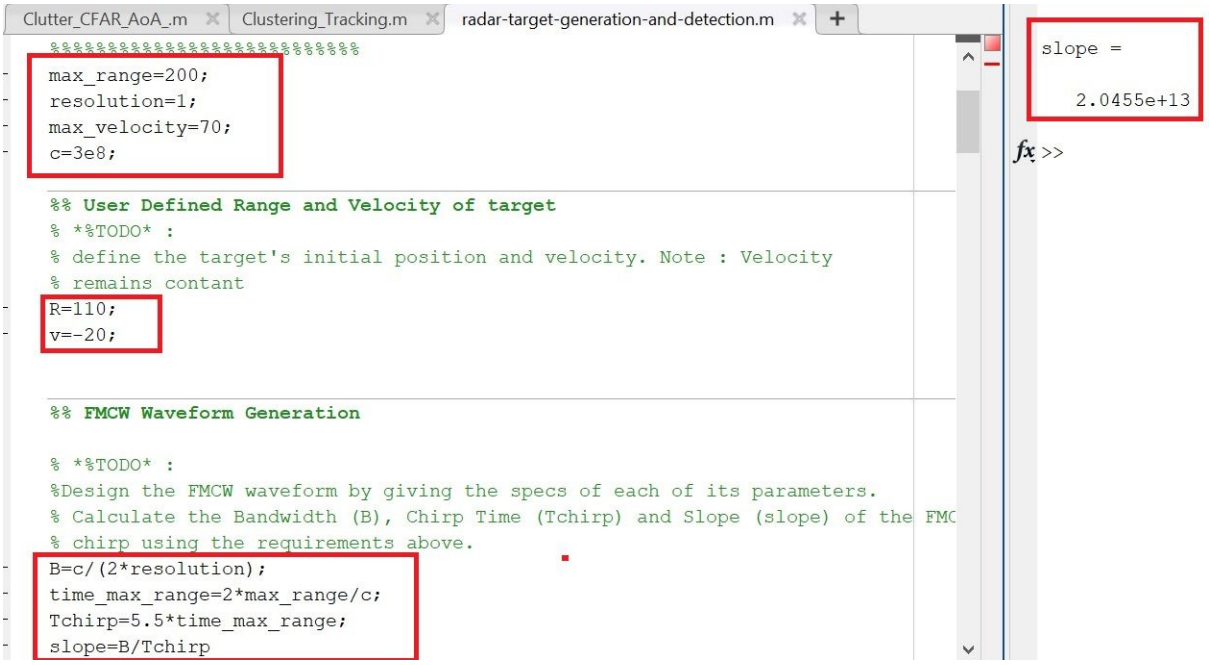
SFND4- Radar Target Generation and Detection

The submission

The submission contains the following file + the matlab script.

FMCW Waveform Design

In below snapshot, I provide initialization parameters for the radar. Slope value= 2.045e13



Simulation Loop

To generate the mix signal, we take the following steps:

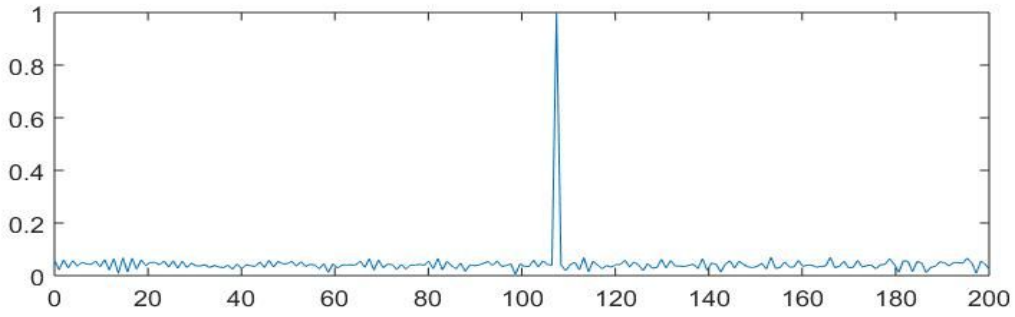
- Initialize timestep + compute position according to constant speed model.
- Compute round trip time tau
- Compute the emitted and received signals Tx & Rx
- Mix the signals

The steps are highlighted in red color in below screenshot.

```
ditor - C:\Users\azzouzm.EMEA\work\projects\SFND_Radar_Target_Generation_Tracking\radar-target-generation-and-detect
Clutter_CFAR_AoA_.m | Clustering_Tracking.m | radar-target-generation-and-detection.m | +

%% Signal generation and Moving Target simulation
% Running the radar scenario over the time.

for i=1:L
    % *%TODO* :
    %For each time stamp update the Range of the Target for constant velocity
    time=t(i);
    Rt=R+v*time;
    % *%TODO* :
    %For each time sample we need update the transmitted and
    %received signal
    Tx(i) = cos(2*pi*(fc*time+slope*(time^2)/2));
    tau=2*Rt/c;
    Rx(i) = cos(2*pi*(fc*(time-tau)+slope*((time-tau)^2)/2));
    % *%TODO* :
    %Now by mixing the Transmit and Receive generate the beat signal
    %This is done by element wise matrix multiplication of Transmit and
    %Receiver Signal
end
Mix = Tx.*Rx;
```

Range FFT (1st FFT)

We apply the fft on the first column of the **reshaped** and **mixed** signal. After normalization, taking absolute value and truncating first half (OK because sinal is real), we obtain below curve.



2D CFAR

First, I introduce the approach used to compute the output signal:

I used the following values for Tr, Td, Gr, Gd and offset:

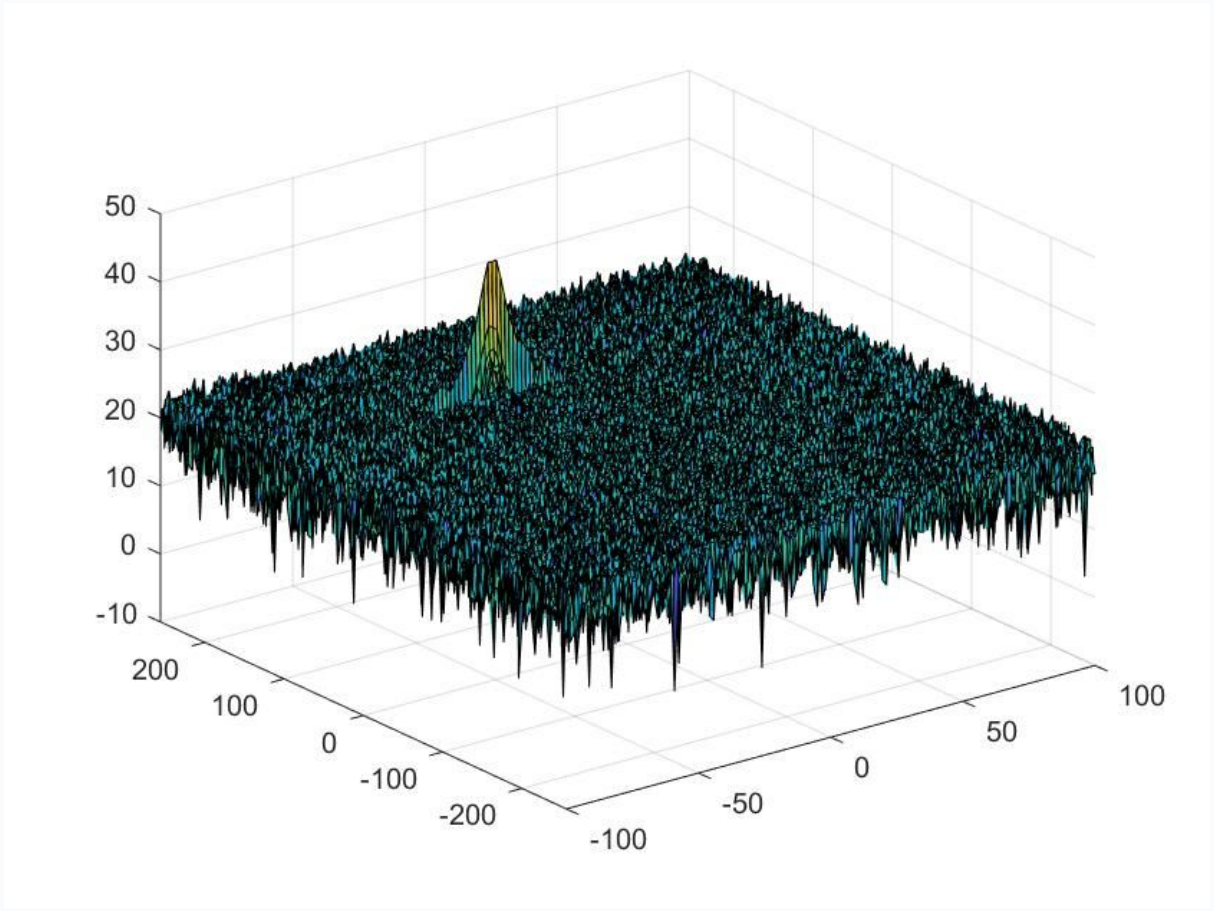
```
%Select the number of Training Cells in both the dimensions.
Tr=8; Td=10;
% *%TODO* :
%Select the number of Guard Cells in both dimensions around the Cell under
%test (CUT) for accurate estimation
Gr=4; Gd=4;
guardKernel=zeros(2*Gr+1,2*Gd+1);
trainKernel=padarray(guardKernel,[Tr Td],1);

% *%TODO* :
% offset the threshold by SNR value in dB
offset=10;
```

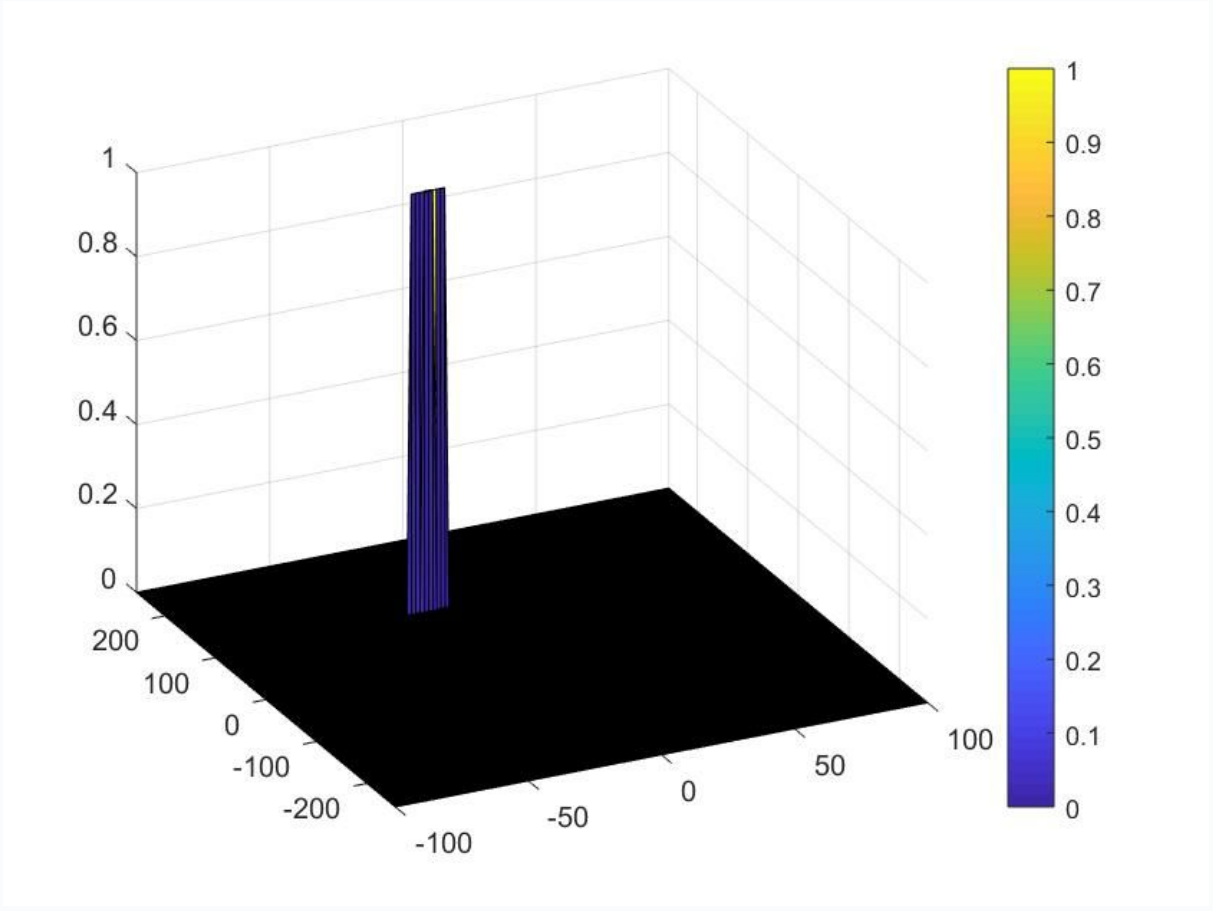
In order to compute the mean of the noise cells on the ***db2pow(RBM)***, we're actually performing a convolution between ***db2pow(RBM)*** and a kernel of size (2xTd+2xGd+1)x(2xTr+2xGr+1) where all green cells are equal to **1** and the others to **0**. For this purpose, I have used the ***conv2*** matlab function.

For this purpose I went through the following steps:

1. **Build noise matrix**
   - Convolution with kernel composed of 1 for training cells and zero elsewhere
   - Normalize the obtained matrix by the kernel sum to get the means
   - Convert to db using ***pow2db***
   - Add offset
2. **Build resulting matrix**
   - If RDM<noise:0, else 1

Below screenshots show the 2d range fft then the final result




1
0.8
0.6
0.4
0.2
0
200
100
0
-100
-200
-100
-50
0
50
100
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0