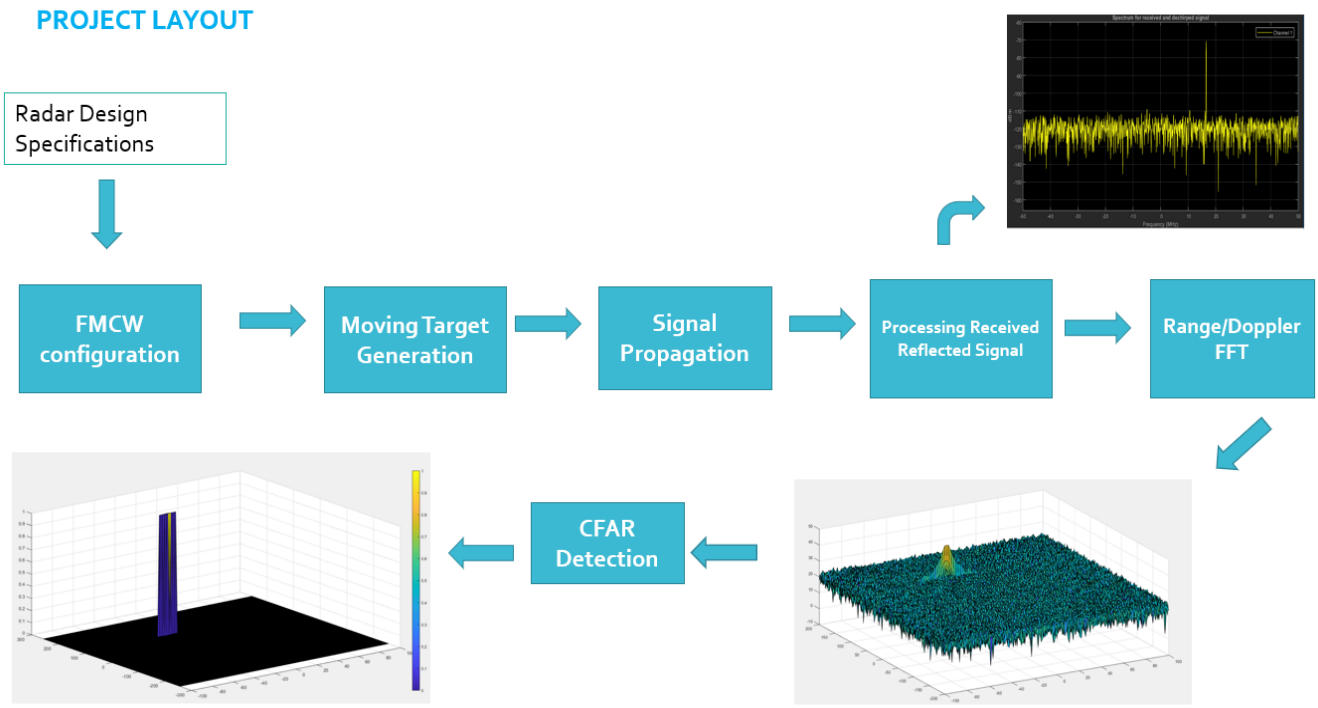
SFND RADAR Target Generation And Detection

The following picture represents the complete project layout.



Here are the each rubrics in alignment with the same.

1 . Using the given system requirements, design a FMCW waveform. Find its Bandwidth (B), chirp time (Tchirp) and slope of the chirp. The defined system requirements of radar below:

Frequency	77 GHz
Range Resolution	1 m
Max Range	200 m
Max velocity	70 m/s
Velocity resolution	3 m/s

The calculated Bandwidth(B), Chirp Duration(Tchirp) and slope (S)

```
% Calculate the Bandwidth (B), Chirp Time (Tchirp) and Slope (slope) of the FMCW
% chirp using the requirements above.

c = 3e8; %speed of light m/s
rangeResolution = 1; %resolution of range m
B = c / (2*rangeResolution);
Tchirp = 5.5 * 2 * max_range / c;
slope = B/Tchirp;
```

FMCW (Frequency Modulated Continuous Wave) for both transmit and recieved signal can be be defined using the wave equations below.

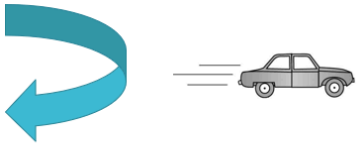
Modeling Signal Propagation for the Moving Target scenario

Slope (α) = Bandwidth / Tchirp

Transmit signal (Tx) $\cos(2 * \pi * (f\_c * t + \alpha * t^2/2))$

Range , time_delay = τ

Receive signal (Rx)
(time delayed version of the transmit signal) $\cos(2 * \pi * (f\_c * (t - \tau) + \alpha * (t - \tau)^2/2))$



Subtracting (Mixing or Dechirping) the receive signal with the transmitter signal gives the frequency shift

Tx.*Rx
(element wise matrix multiplication) $\cos(2 * \pi * \left(2 * \alpha * \frac{R}{c} * t + 2 * f\_c * v/c * t\right)$

Range Doppler

2. Simulate Target movement and calculate the beat or mixed signal for every timestamp.

The initial target position and constant velocity is set as below.

```
%speed of light = 3e8
%% User Defined Range and Velocity of target
% define the target's initial position and velocity. Note : Velocity
% remains contant
range = 110;
velocity = -20;
```

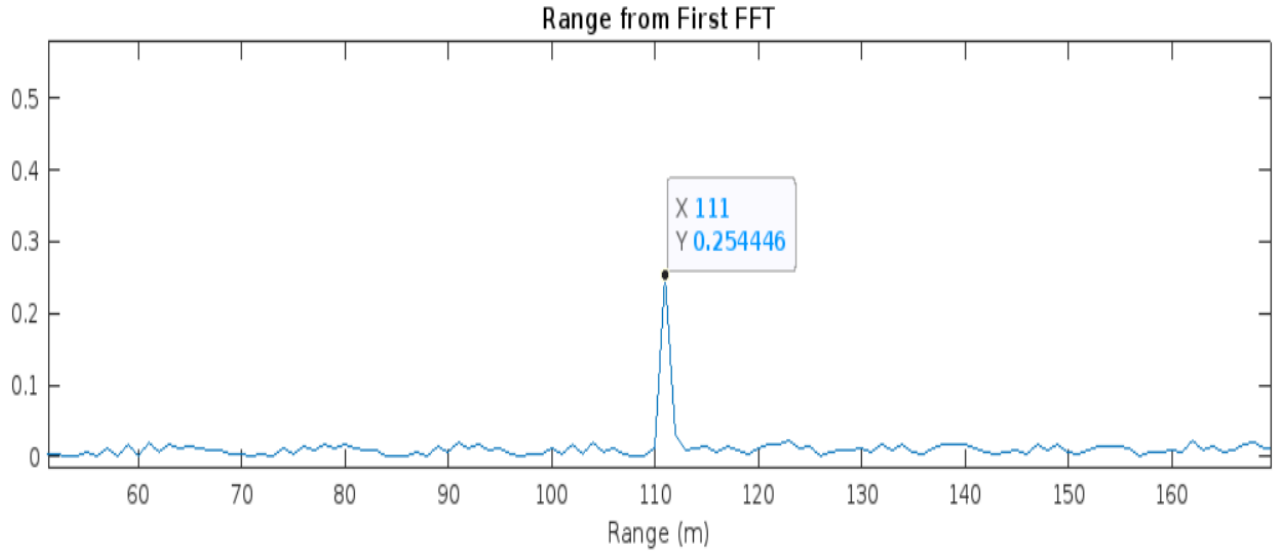
Next, the range and delay_time is simulated using motion equations as below.

```
for i=1:length(t)

    %For each time stamp update the Range of the Target for constant velocity.
    r_t(i) = range + velocity*t(i);
    td(i) = (2 * r_t(i))/ c;

    % *%TODO* :
    %For each time sample we need update the transmitted and
    %received signal.
    Tx(i) = cos(2 * pi * (fc * t(i) + (slope * t(i)^2) / 2));
    Rx(i) = cos(2 * pi * (fc * (t(i) - td(i)) + (slope * (t(i) - td(i))^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
    Mix(i) = Tx(i).*Rx(i);

end
```

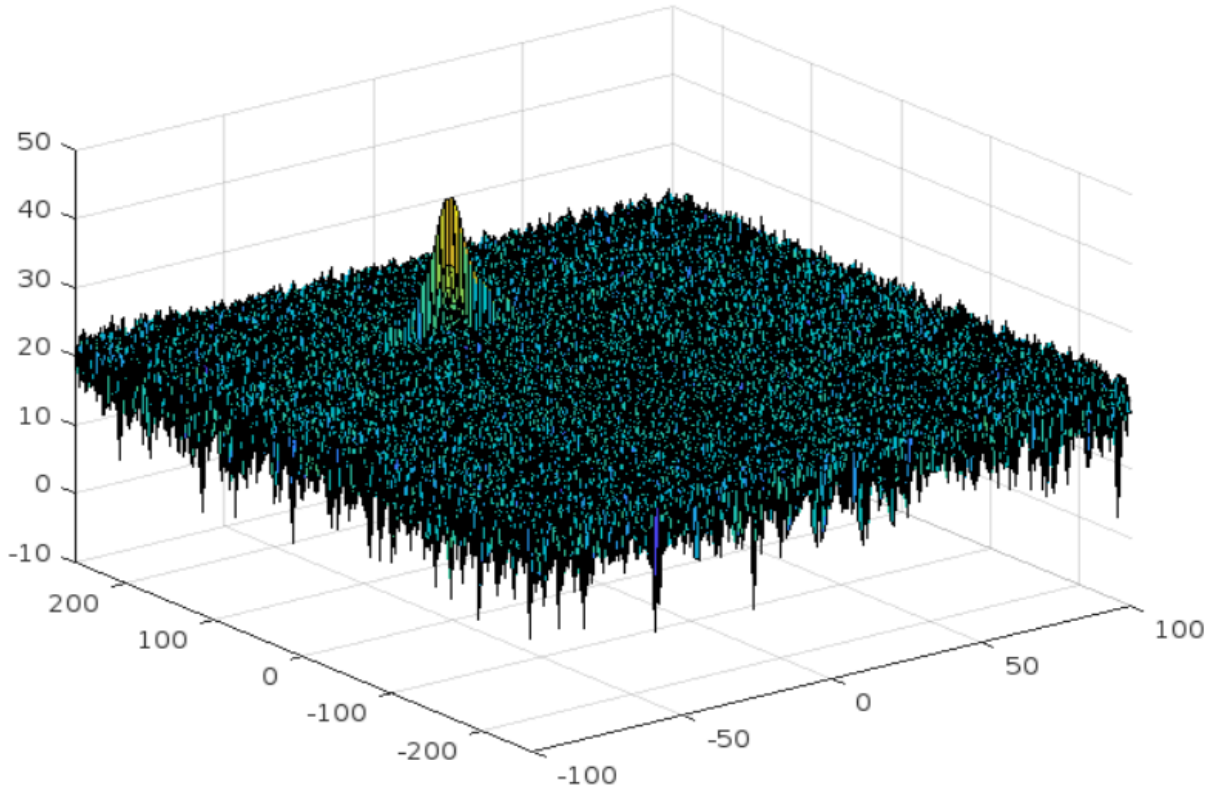
Using the time stamp and simulated delay time, the Tx and Rx are generated and mixing them both using Tx*.Rx gives us the beat signal , which will be utilised by RangeFFT or 1D-FFT for range measurements

3. Implement the Range FFT on the Beat or Mixed Signal and plot the result.

The recieved beat signal is further processed by 1D - FFT as below, matlab has an inbuilt function which does all the maths. The output is shown, where the peak is clear w.r.t to the range we have initially set and the value is also in the range of +/- 10m



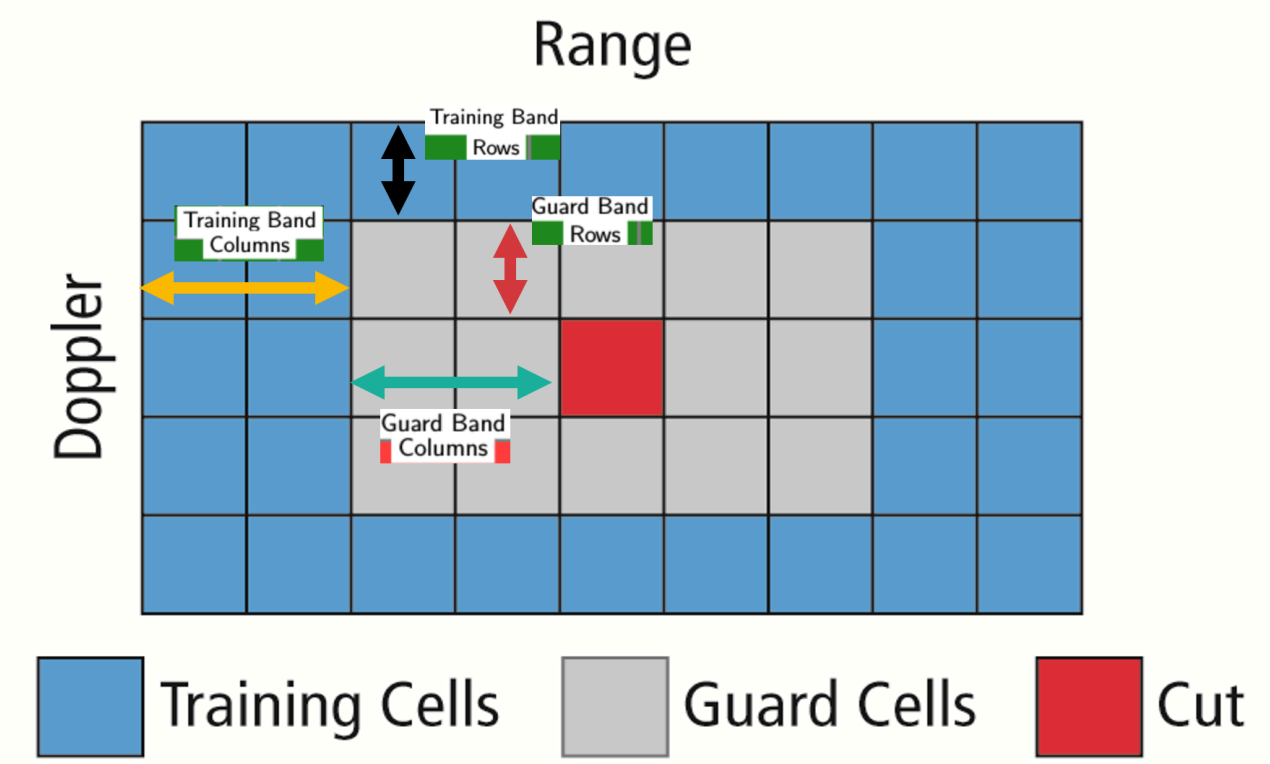
The 2D-FFT of Range Doppler Map(RDM) is already implemented , and the result is gotten as below. We can nicely gauge we are very close to the output, but one more step on refining this is to be done which is 2d-CFAR



4.Implement the 2D CFAR process on the output of 2D FFT operation, i.e the Range Doppler Map

The 2D -CFAR - Constant False Alarm Rate is the technique which helps us in finding Dynamic threshold, that segregates noise from target peaks, considering the homogenous nature of noise around the peaks.

The process involves, 3 category of cells as picture below,



The training cells are the choosen cells around Cell Under Test(CUT), from which the values of training cells are summed and averaged, which is further added with an offset value to fianlize the actual threshold value. Note the gaurd cells are excluded from the calculation

The gaurd cells helps avoiding leaking of the target signal to training cells, which can impact noise calculation.

This Training_Cell and Gaurd cell window is slided across the entire RDM to dynamically calculate the threshold.

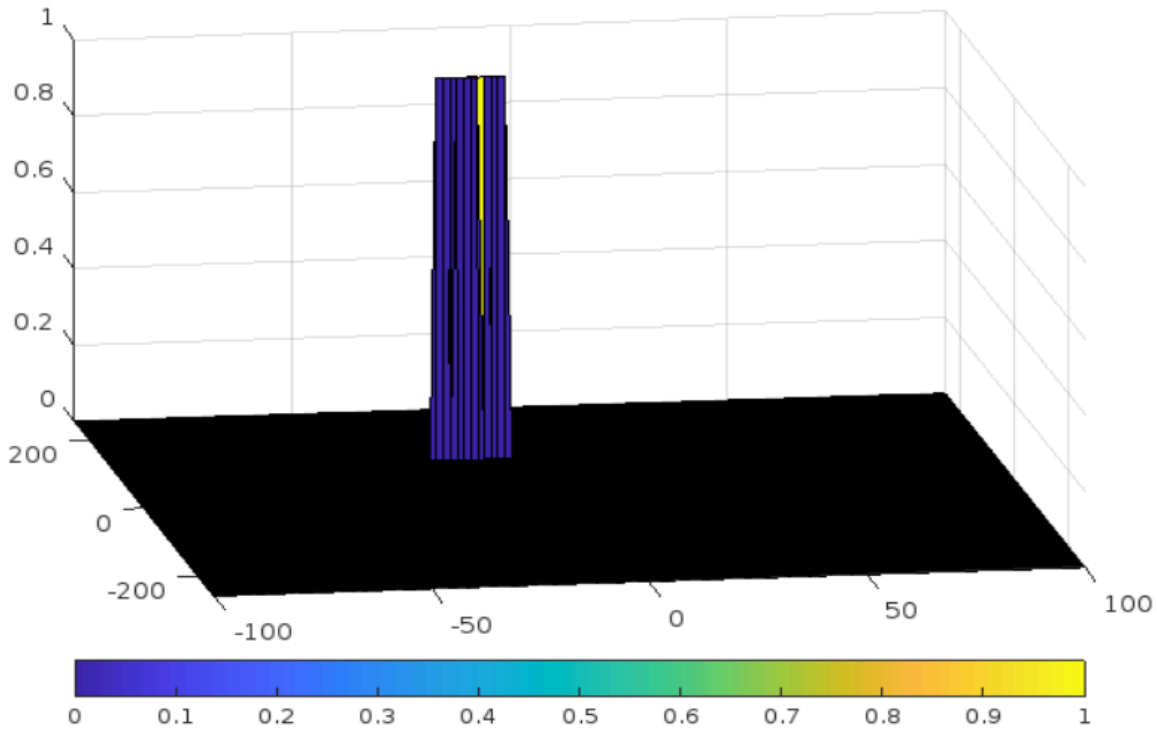
The following code snippet is CA(CellAveraging)-2DCFAR: where,

```
%Select the number of Training Cells in both the dimensions.
Tr = 14;
Td = 14;

% *%TODO* :
%Select the number of Guard Cells in both dimensions around the Cell under
%test (CUT) for accurate estimation
Gr = 5;
Gd = 5;

% *%TODO* :
% offset the threshold by SNR value in dB
offset = 9;

% *%TODO* :
%Create a vector to store noise_level for each iteration on training cells
noise_level = zeros(1,1);
rangeSize = 2*(Tr+Gr) + 1;
dopplerSize = 2*(Td+Gd) + 1;
```

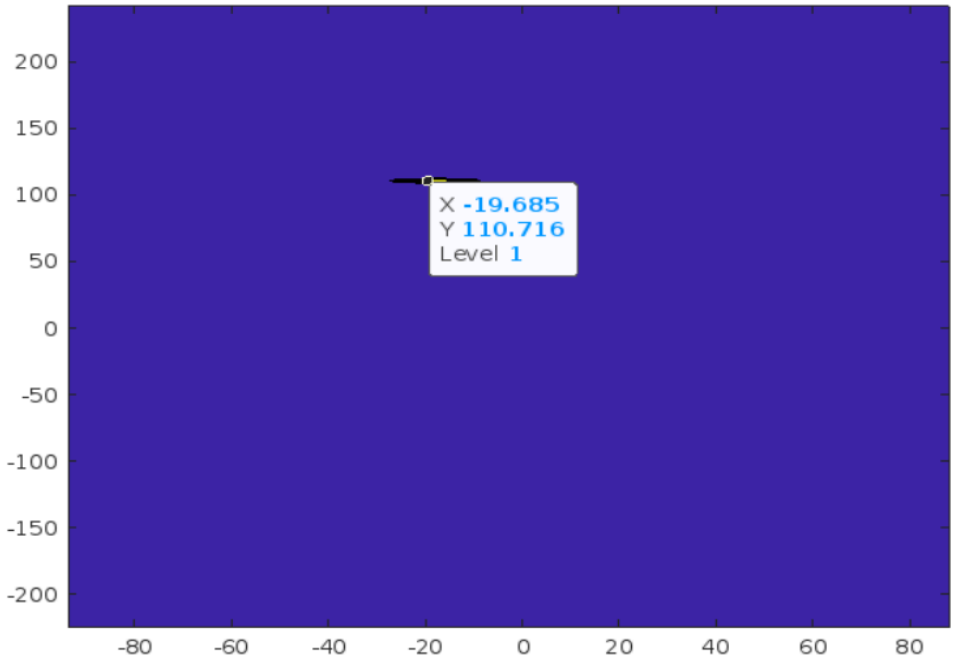
- The Tr - Training Cell numbers along Range, Td- Training Cell numbers along Doppler Gr - Gaurd Cell numbers along Range, Gd- Gaurd Cell numbers along Doppler are defined
- Offset value is set
- Corresponding window size using Training, Gaurd cells and CUT are formed by Grid Size = (2Tr+2Gr+1)(2Td+2Gd+1).
- Gaurd Cell region + CUT can be defined by (2Gr+1)(2Gd+1).
- The Training cells are finally (2Tr+2Gr+1)(2Td+2Gd+1) - (2Gr+1)(2Gd+1)

```
% CFAR
trainingCellSize = rangeSize * dopplerSize - (2 * Gr + 1)*(2 * Gd + 1);
cfarSignal = zeros(Nr/2, Nd);
for i = 1+Tr+Gr:(Nr/2 - (Gr+Tr))
    for j = 1+Td+Gd:(Nd - (Gd+Td))
        unifiedSumOfTrainingAndGaurdCells = sum(db2pow(RDM(i-(Tr+Gr):i+Tr+Gr, j-(Td+Gd):j+Td+Gd)), 'all');
        sumOfGaurdCells  = sum(db2pow(RDM(i-Gr:i+Gr, j-Gd:j+Gd)), 'all');
        noiseTot = unifiedSumOfTrainingAndGaurdCells - sumOfGaurdCells;

        threshold = noiseTot / trainingCellSize;
        threshold = db2pow(pow2db(threshold) + offset);

        signal = db2pow(RDM(i,j));
        if (signal <= threshold)
            cfarSignal(i,j) = 0;
        else
            cfarSignal(i,j) = 1;
        end
    end
end
```

- Average of Training cells are computed and summed with the offset
- Each RDM cell signal level is checked against the threshold, if its greater than threshold output is assigned 1 else 0.
- if we continue for all the possible cells, we will get the final map as below, where the peak is clearly segregated from noise.



Follwoing is the Contour graph in 2D for clear focus.


200
150
100
50
0
-50
-100
-150
-200
-80
-60
-40
-20
0
20
40
60
80
X -19.685
Y 110.716
Level 1