## 2D CFAR Implementation

Problem Statement:

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

Logic behind solution:

1. Selection of Training, Guard cells and offset

Various combinations of Training and Guard cells were tried and all found to be working. The offset value impacts the clutter suppersion more as compared to Training and Guard cells. The offset value lower than 4 found to be producing more than one maxima. The final values used can be found below

Training cells in range dimension: 10

Training cells in doppler dimension: 8

Guard cells in range dimension: 3

Guard cells in doppler dimension: 3

Offset: 4.5

1. Implementation steps for the 2D CFAR process

* I started off by calculating the grid size with the help of following formula

grid\_size = (2\*Tr + 2\*Gr + 1),(2\*Td + 2\*Gd + 1)

* To store output an array of zeros named cfar\_rdm having exact same size of Range Doppler Map (RDM) was created.
* A nested for loop was used to leep over the RDM cell by cell.
* A was considered as Cell Under Test (CuT) only when a sufficient buffer for cells at border was left, i.e. the CuT satisfies following conditions:

For Row number (i): i is greater than Td and i is less than num\_rows\_in\_rdm - num\_of\_rows\_in\_grid

For Column number (j): j is greater than Tr and j is less than num\_column\_in\_rdm - num\_of\_column\_in\_grid.

* For every CuT, a temp\_grid of size grid\_size was selected from RDM. A flatten version of temp\_grid flat\_temp\_grid was created.
* Location of CuT was found in flat\_temp\_grid with the formula,

ceil(length(flat\_temp\_grid)/2)

* The value in CuT was fetched and stored in CuT.
* Training cells in flat\_temp\_grid were found in two steps. From beginnin of flat\_temp\_grid to CuT, and from CuT till end of flat\_temp\_grid.
* Noise level from all training cells were averaged and an offset was added to it to calculate threshold value.
* This threshold was calculated with the CuT value and if CuT was found to be greater than threshold then the respective CuT was set to 1 in output array i.e. cfar\_rdm array.
* Now, as cfar\_rdm was a 2D array and CuT\_loc was a location of CuT from a 1D array, following logic was used to find CuT location in cfar\_rdm

Row\_num (i) in cfar\_rdm = i+ ceil(grid\_size(1)/2)

Column\_num (j) in cfar\_rdm = j+ ceil(grid\_size(2)/2)

1. Steps taken to suppress the non-thresholded cells at the edges

* I created cfar\_rdm as an output array which contained all zeros and had same size of RDM array.
* A cell in cfar\_rdm was set to 1 only when Cut > threshold, by doing this i maintained the size of cfar\_rdm and saved some calculations for taking care of non thresholded cells.