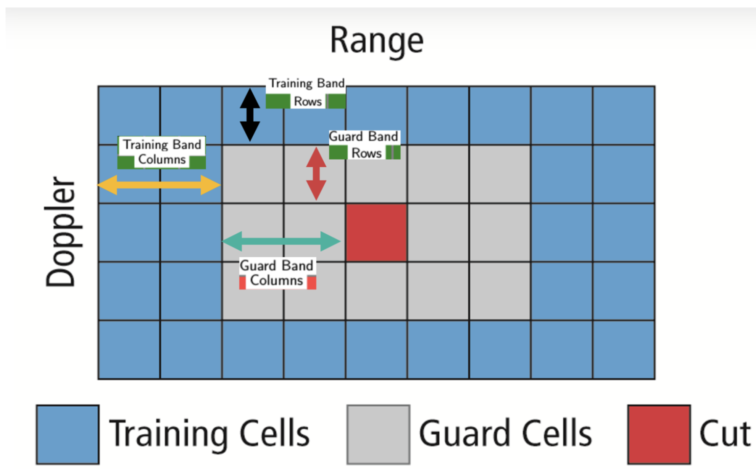
Radar Target Generation and Detection
Project Writeup

2D CFAR Process implementation overview

The idea behind implemented 2D CFAR algorithm is to process all the cells in given range doppler map and separate signal values from noise data. This is achieved by calculating the average level of noise across all range doppler map cells and using it as a threshold (with additional buffer) for identification of all the signal cells which will have the signal values above this threshold. In order to process all cells of the map in computationally efficient way, the sliding window approach is used, when we define the subset set of cells that for the window of fixed size (2xTr+2xGr+1) x (2xTd+2Gd+1), where CUT is the cell under test and located in the center of sliding window, mean the value of this cell is going to be compared with calculated average value of noise, to decide whether this cell contains the signal data, Tr and Td are the number of columns and rows accordingly of training cells, used for noise level calculation, Gr and Gd - number of columns and rows of cells, which paly a role of buffer between training cells and cell under test in order to minimize the bias of CUT value on calculation of noise level.



Sliding Window of 2D CFAR algorithm

The implemented algorithm of 2D CFARconsist of following steps:

1. Iterate over all cells of the range doppler map using a sliding window approach with a strides by row and column equal to one cell accordingly. The sliding window size and structure are fixed and match above description
2. In each sliding window iteration step, iterate over all cells inside the current sliding window, and compute the average value among all the training cells, by calculating the total sum of values of these cells and dividing by their number. The calculated average value of training cells with added the fixed offset value will be a noise threshold level for this sliding window.
3. If the value of the current Cell Under Test is bigger than the calculated value of the noise threshold then the value of the corresponding cell in the range-doppler map is set to value 1 otherwise it is set to value 0.
4. Afterwards shift the sliding window by one cell and repeat steps 3 and 4 fow cell of new sliding window
5. Continue this process until all cells in range-doppler map are processed
6. Afterwards suppress all the values of the cells which are located in the edges of range-dopler map, and therefore were not processed in a role of CUT cells because of the margins of Tr+Gr and Td+Gd in the left/right and top/bottom sides of the map accordingly

Selection of Training, Guard cells and offset

The constant values for number of training cell rows and columns (Tr and Td) were selected in a way that allows to calculate the stable level of noise across whole range-doppler map without sacrificing the signal detection sensitivity and in the same type preventing the false interpretation of higher spike of noise as a real signal. Tr=6, Td=6 have shown best results for given input data.
Number of guard cell rows and columns Gr and Gd had to be as small as possible to minimize the loss of effect of the values inside these cells on the calculated level of noise, but at the same time must be big enough to prevent the impact of target signal on the noise level estimation. Gr=2 and Gd=2 satisfied these criteria for given input data.
The offset value plays a role of a safety buffer on top of calculated average level of noise, which additionally helps to prevent the false positive signal detections caused by noise spikes. In the same time it must be as small as possible to prevent suppression of real signal values which are not much bigger than average noise estimation. For given input data, the offset value 8 was chosen as the minimal value for satisfaction of stable signal detection results.

Suppressing the non-thresholded cells at the edges

Last step of above described algorithms for 2D CFAR is intended to suppress all the values of the cells of range-doppler input data map, which are located in the ‘margin’ edges, and therefore were not processed as CUT cells. These margins are formed by the training and guard cells of the sliding window accordingly Tr+Gr and Td+Gd in the left/right and top/bottom sides of the map. Since these cell can also theoretically contain a signal data and wont be set to 1 during the CUT cell value processing, these must be suppressed afterwards. Therefore after the execution of the main part of the algorithm, all cells of the map, whose values remain bigger than 1 or smaller than 0 (apparently only those inside the margins) are set to 0.