Files and Functions

Project_config.py

This file has all the global variables like ip address of raspberry pi & Local oscillator, Number of ADAR1000, speed of light etc.

Gain_cal_val.py

For Now, this file has gain calibration value. It consists of 4 element arrays and number of arrays is equal to number of adar1000. In near future try to serialize those value in main source code using maybe pickle.

Phase_cal_val.py

For Now, this file has Phase calibration value. It consists of 4 values for each adar1000 in system. In near future try to serialize those value in main source code using maybe pickle.

Top_layer.py

For Now, this file has main function in which adar1000, ad936x and adf4159 is instantiated depending on the configuration in project_config.py file. After initializing all the device to required configuration it ask for calibration and performs monopulse tracking.

Functions.py

This file has all the major functions that are called from either top_layer file or other functions of same file.

1. Sdr_init(my_pluto, my_pll)

   Things passed: - my_pluto = ad936x device (Rx) | my_pll = adf4159 device(Tx)

   This function initializes and sets properties, like Tx & Rx frequency, gain, sample rate etc, of ad936x & adf4159.

2. ADAR_init(adar)

   Things passed: - adar = Single adar1000(1 out of n element of beamformer)

   This function resets and initialize one single adar1000 at a time. It sets tr_source, bypass RAM control, vga enable etc.

3. ADAR_set_gain(adar_list)

   Things passed: - adar_list = list of all n elements/adar1000 of beamformer

   This function sets gain according to the calibrated value and if system is not calibrated it sets it to 127.

4. ADAR_set_RxPhase(adar, Ph_Diff, adar_no)

   Things passed: - Single adar1000, Ph_Diff is phase value swept from -180 to 180 and adar_no is nth number of adar whose phase is currently or will be set in this function.

   e.g:- if adar_no is 1 phase value of 1st adar i.e. channel 1 to 4 is being set.

   If adar_no is 2 then phase value of 2nd adar i.e. channel 5 to 8 is being set.

   In this function phase value is written to adar1000.

5. ADAR_Plotter(adar_list, sdr)

   Things passed: - adar_list = list of all adar and sdr = ad936x device (Rx)

   This function actually plots the output of monopulse tracking experiment depending on the rx/tx source and number of adar1000.

   Phase is constantly updated in this function.

6. Phase_calibration(adar_list, sdr)

   Things passed: - adar_list = list of all adar and sdr = ad936x device (Rx)

   This function one by one set gain of 2 adjacent channels to max and all other gain to 0. The Phase of those channel are set in such a way that they are 180 degrees apart to get gain.

   This function returns a value which is angle at which the gain is minimum/null value.

   Also, all the values are then referenced to single channel i.e. channel 1 and it is stored in Phase_cal_val.py

7. phase_cal_plot(adar_list, sdr, cal_element)

   Things passed: - adar_list = list of all adar, sdr = ad936x device (Rx) and cal_element defines the current pair of adjacent channel that are being calibrated.
   This function sweeps values of 2 adjacent channel 180 degree apart. This is similar to ADAR_Plotter function but only 2 channels are considered at a time.

8. ADAR_set_CalRxPhase(adar_list, Ph_Diff, cal_element)

   Things passed: - adar_list = list of all adar, Ph_Diff is phase value swept from -180 to 180 and cal_element defines the current pair of adjacent channel that are being calibrated.
   This function actually sets the phase.

9. gain_calibration(adar_list, sdr)

   Things passed: - adar_list = list of all adar and sdr = ad936x device (Rx)

   This function sets gain of 1 channel to max at a time and all other channel gain to 0. Performs calibration and stores value in gain_cal_val.py

10. gain_cal_plot(adar_list, sdr, gcal_element)

Things passed: - adar_list = list of all adar, sdr = ad936x device (Rx) and cal_element defines the current pair of adjacent channel that are being calibrated.

This function sweeps phase value of 1 channel from -180 to 180 degree. This is similar to ADAR_Plotter function but only 1 channel is considered at a time.

11. ADAR_set_gCalRxPhase(adar_list, Ph_Diff, gcal_element)

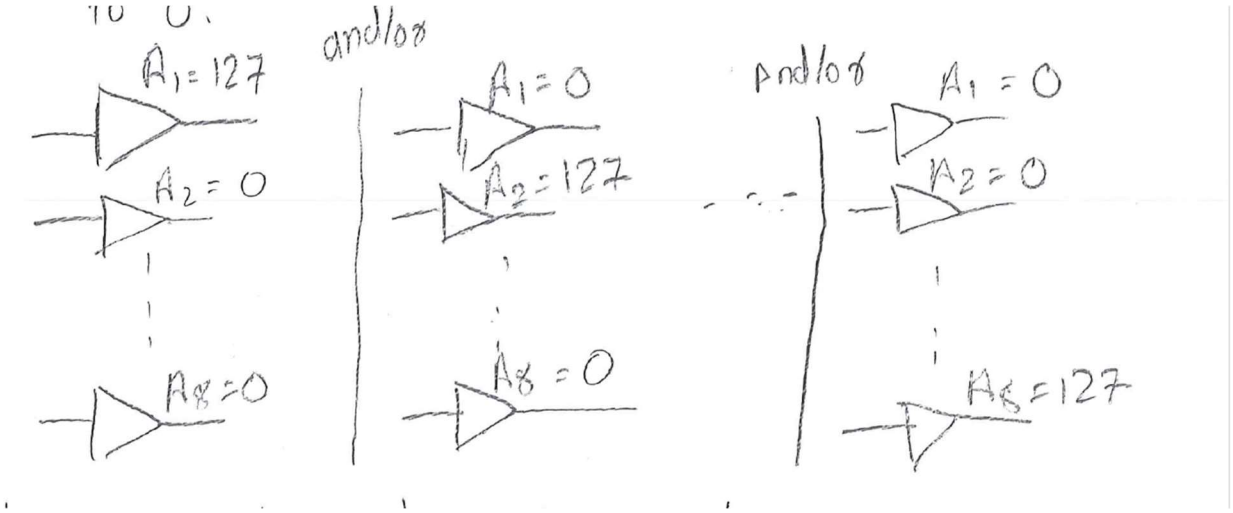
Things passed: - adar_list = list of all adar, Ph_Diff is phase value swept from -180 to 180 and cal_element defines the current pair of adjacent channel that are being calibrated.

This function actually writes phase values to adar1000.

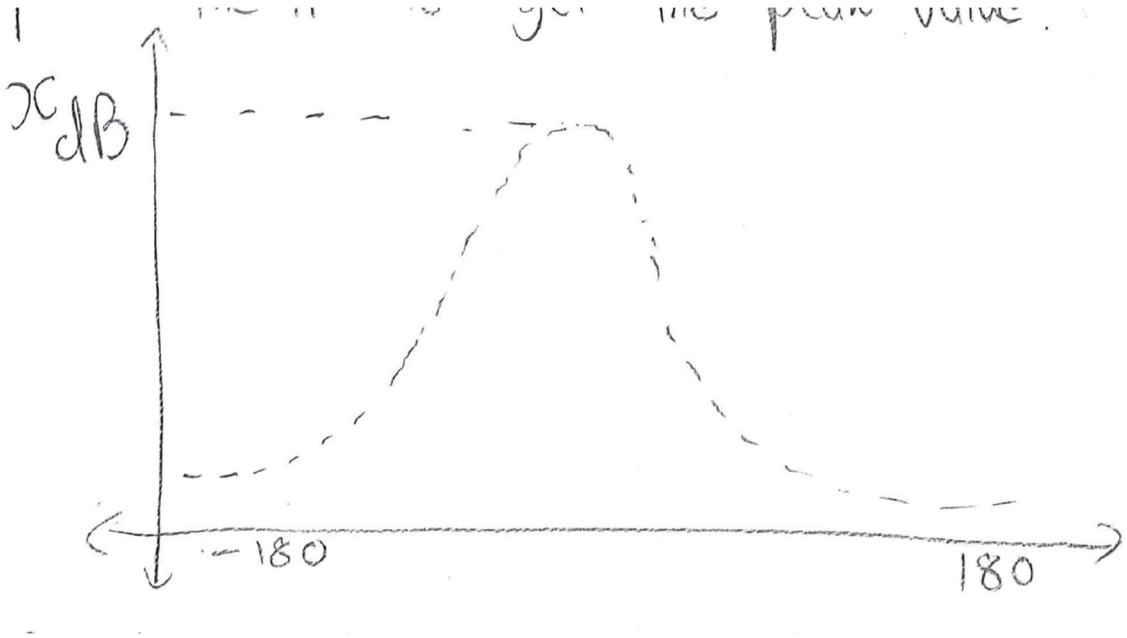
Gain Calibration:-

Steps

1. Set gain of any 1 channel at a time to maximum and all other to 0.



2. Sweep Phase value of that channel from -180 to 180 degree and plot arrayangle vs arraygain to get the peak value.



3. Repeat the above steps for all 8 channel and get value of ‘x’ for each one of them. We will get an array of 8 elements.
   For my setup the value of x ranged from -29 to -27db.

4. We cannot have gain more than 127, So set the channel with minimum value of x to maximum gain i.e. 127. Scale all other channel gain accordingly. For me -29 was minimum so,

   -29 == 127

   -27==?

   = 118.24 = 118

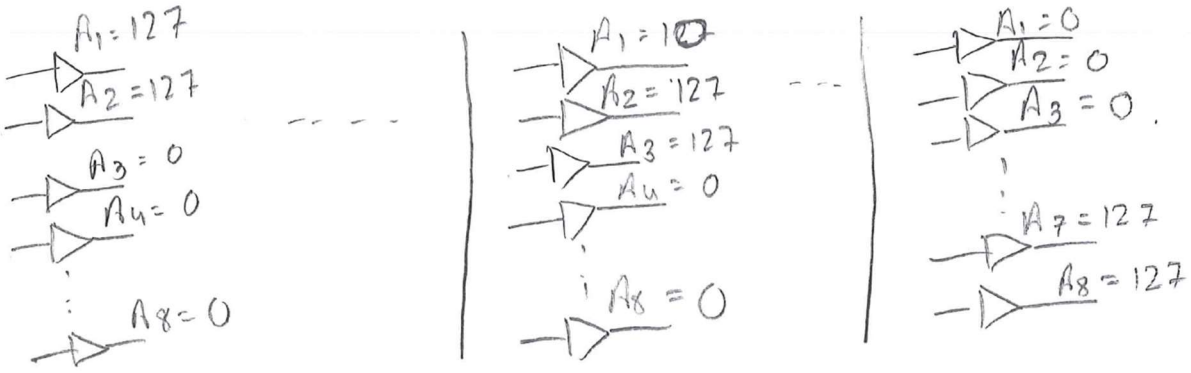
   This will give an 8 element array with values ranging from 115 to 127 with one fixed channel gain at 127.

5. Set gains of all the channel according to this values to make sure maximum of each channel is same.

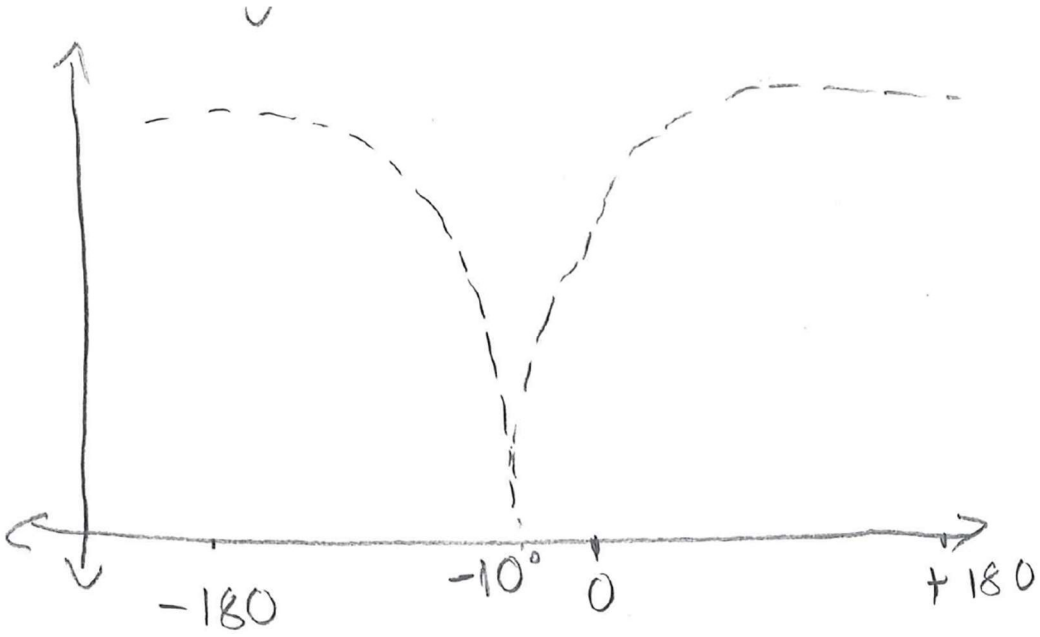
Phase Calibration:-

Steps

1. Set the gain of any two adjacent channel to max, for now, later change it to their max i.i. Calibrated gain values and all other channel to 0.



2. Sweep the phase of those two adjacent channel from -180 to 180 degree in such a way that they are aalways 180 degree apart. Plot this output i.e. array-angle vs array-gain to get a NULL



3. Repeat this for all adjacent channels to get 7 angle/NULL values. This values need to be subtracted from it's adjacent element to get a NULL at 0. This will get a 7 element array.
4. Trace back all the channels with respect to single channel i.e. 1st channel for our case.
   It can be understood as follows:-
   Channel 1 is 0 degree out of phase w.r.t itself.
   Channel 2 is "1st value of array" degree out of phase w.r.t channel 1.

Channel 3 is "2nd value of array" degree out of phase w.r.t channel 2.
Therefore channel 3 is "1st value of array - 2nd value of array" out of phase w.r.t channel 1.
And so on…

5. Continue this to get 8 calibration values in which 1st element is always 0.