Thank you very much for sharing the links regarding the range-azimuth heatmap plots. I went through them plus some other related E2E forum questions about this topic, but I still have problem extracting a meaningful range-azimuth heatmap plot.

I need to plot the range-azimuth(angle) heatmap from the raw data I recorded using “mmWave Studio” and **not** the cloud Demo visualizer. I am using AWR1843BOOST with 3 Tx and 4 Rx antennas in TDM mode, which generates 12 virtual antennas.

1) Please correct me if I am wrong in any part of understanding this. I have arranged the data into 12 virtual Rxs. For each receiver I have 256 ADC samples, 128 chirps, and 100 Frames. Based on the picture below from the “spuim4b” and “swra554a” documents (figures are provided below) I have considered that Tx1 and Tx3 correspond to Rx1-Rx4 and Rx9-Rx12, respectively, and are used to extract the azimuth data while Tx2 corresponds to Rx5-Rx8 which are used for the elevation data. Is this correct?

Then I arrange Rx1-Rx4 and Rx9-Rx12 into a matrix of size 8*256 to account for the 8 virtual azimuth antennas and Rx5-Rx8 into a matrix of size 4*256 to account for virtual elevation antennas. Is this correct?

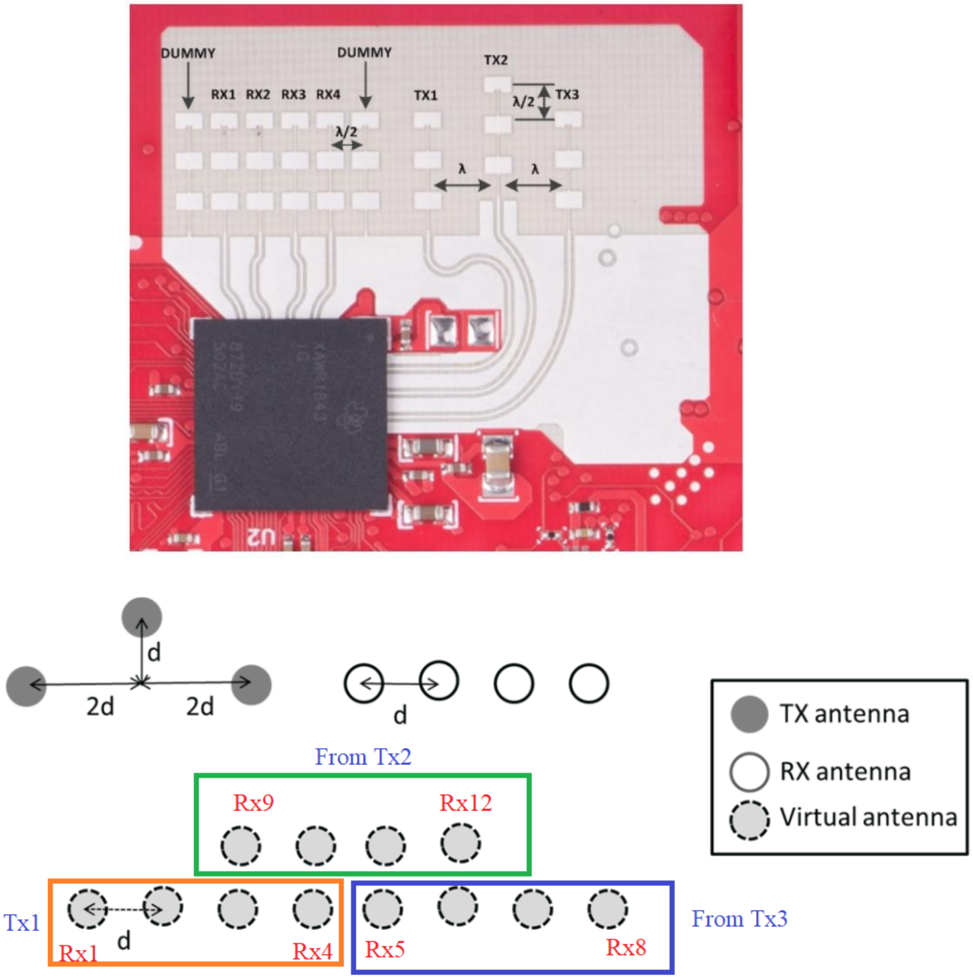

Fig. 1

2) I form the 8*256 matrix for each chirp of each frame, then to obtain range-azimuth heatmap I take a 2D FFT once along 256 samples (fast time) to obtain range and once across the 8 antennas to obtain the angle. Is this the right approach? When I do this and plot the one-dimensional range and angle plots separately (plot(theta, columns of the matrix) and plot(range, rows of the matrix)), I obtain the expected correct results. But when I use plot the heatmap of the same matrix, the results are off. Any idea why this happens? Below I provide more information.

I need to obtain Range azimuth heatmap of the form provided in “swra549” document (like the figure below). But my outputs do not make much sense.

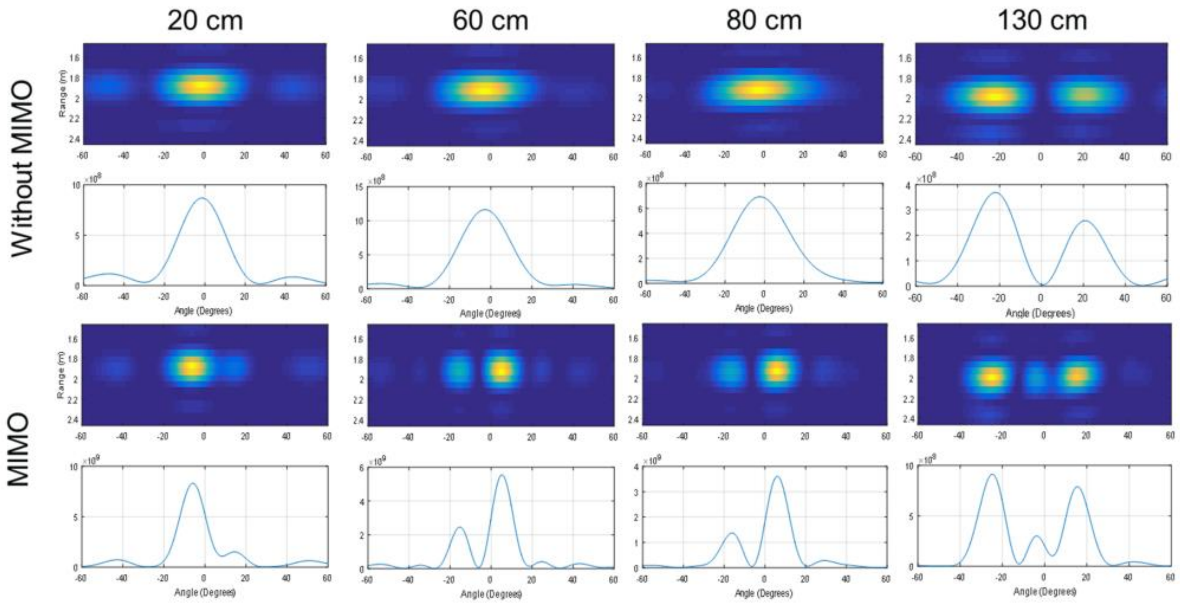

Fig. 2. This figure is from the TI document “swra549”.

I found a code shared in a TI E2E forum (which is provided for the Demo visualizer users and I tried to follow a similar approach). The code is shared below. In the following, x is the 8*256 (azimuth virtual receivers * ADC samples) matrix.

```
NUM_ANGLE_BINS = 1000;
x1 = fft(x,NUM_ANGLE_BINS);
theta = asind([-NUM_ANGLE_BINS/2+1 : NUM_ANGLE_BINS/2-1]'*(2/NUM_ANGLE_BINS));
range = 0:c/(2*BW):f(end)*c/(2*cr);
x1 = x1';
x1 = fftshift(abs(x1));
x1=flip(x1,2);
x2 = fft(x1,length(x1));
x2 = x2';
QQ = fftshift(abs(x2),2);
QQ=QQ.';
imagesc(theta,rng,abs(QQ))
```

I tried this code on simulation data first since I could not relate what I saw from my measurement data. Although, the one-dimensional results (the plots below the range azimuth heatmap in the figure above) for range and angel estimation work perfectly, my range azimuth heatmap with this approach has some problem. I have developed the algorithms for six different angle estimation methods and all work perfectly but not the range azimuth heatmap. I provide an example below. Any idea why this happens? Am I taking the FFT along wrong directions?

Three targets are considered at -10, -35, 55 degrees at 5, 7, and 10 m distances respectively.

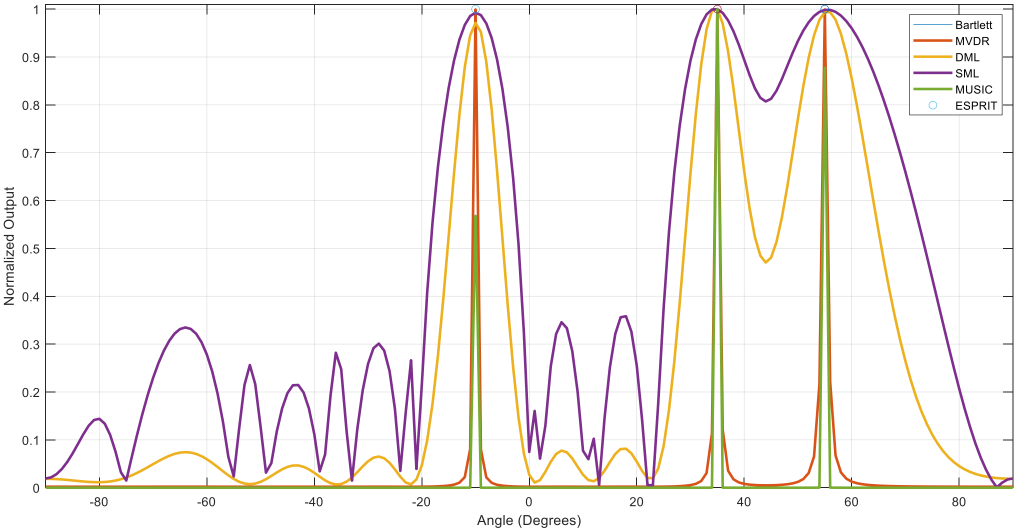

Fig. 3 Angle estimation with 6 different algorithms, all work correctly

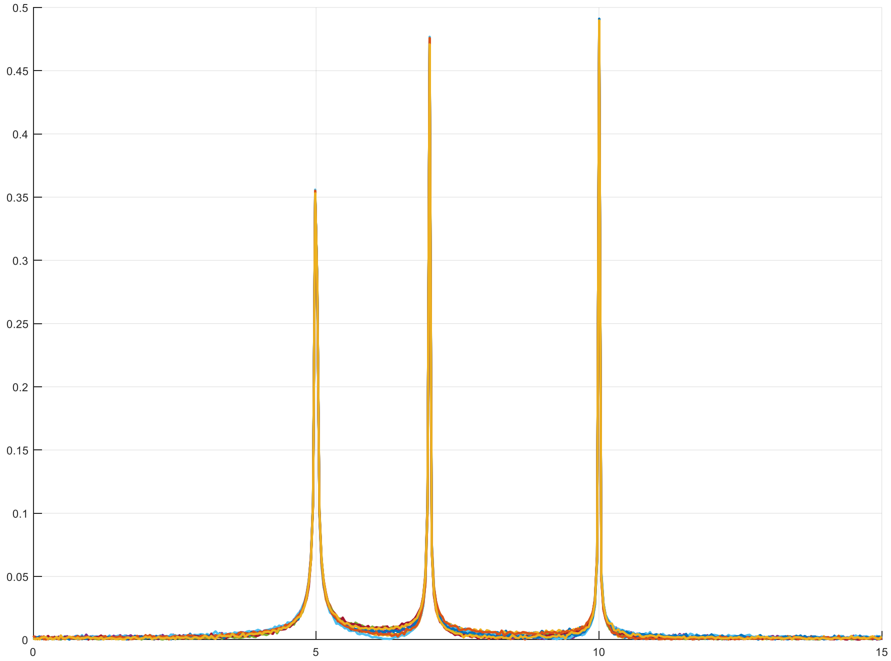

Fig. 4 Range FFT gives the correct results

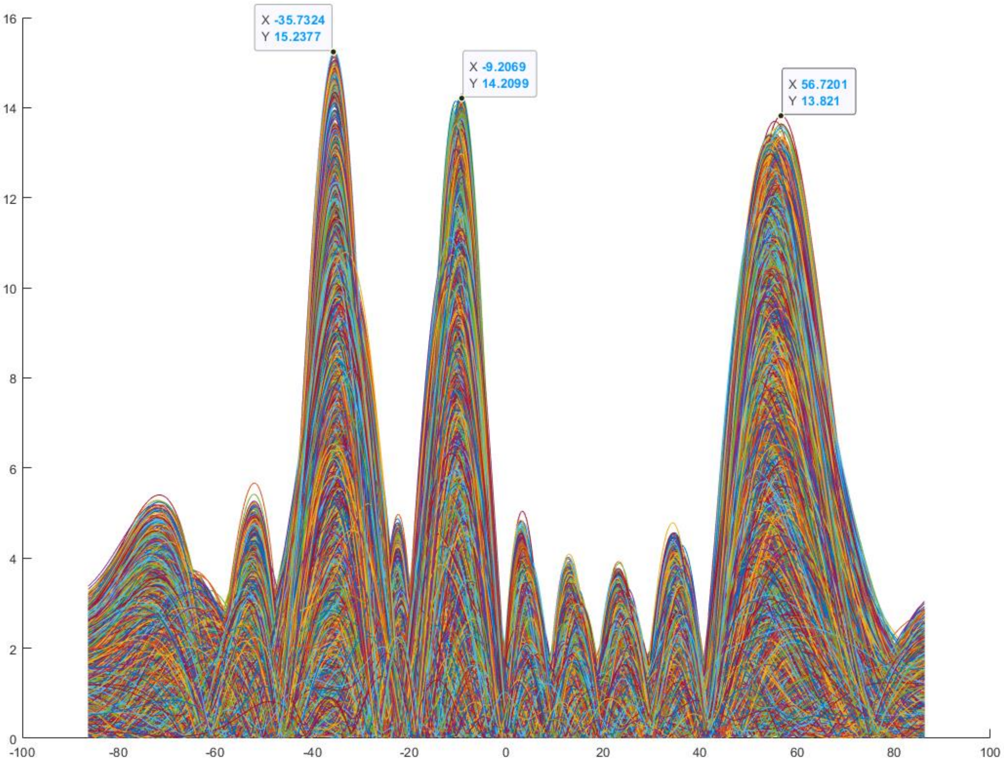

Fig. 5. Angle FFT taken for each chirp of each frame, the FFT is taken along the antennas (along the columns of the 8*256 sized matrix) and the angle is estimated correctly.

Although both angle and range in one dimensional calculation are calculated correctly, when I get the 2D FFT based on the code above once along ADC sample and once along the virtual antennas, the following heatmap shows up which does not show the reflections in the correct spots.

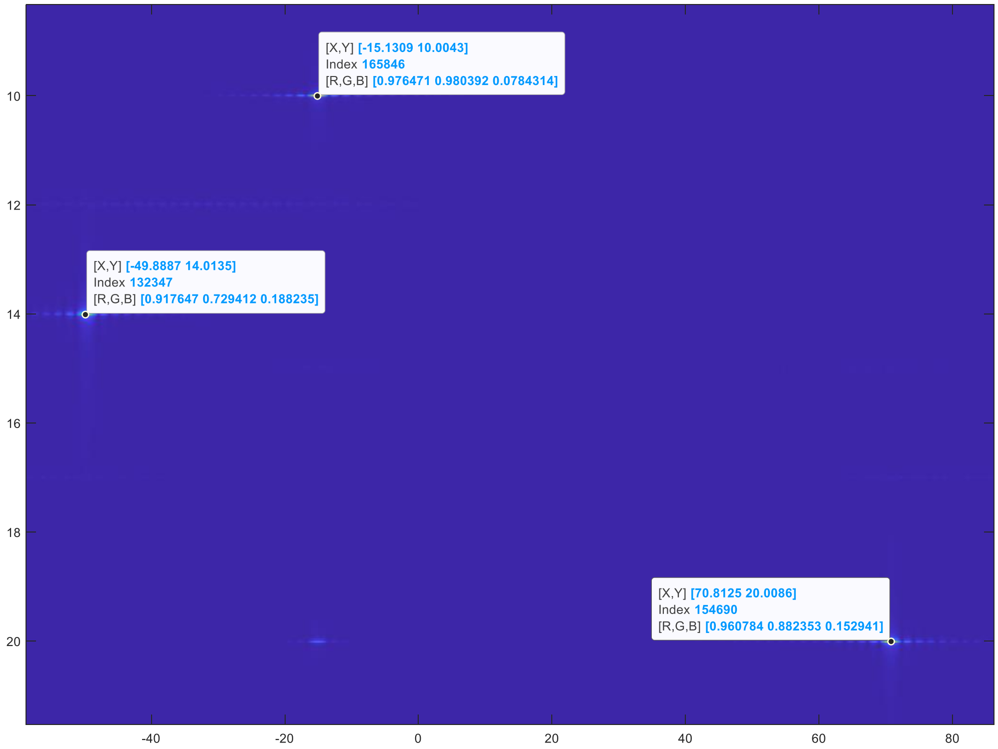

Fig. 6. Incorrect results

Although one dimensional range and angle FFTs are correct, heatmap of the same data shows the range in twice what is expected, and the angle is also deviated. Instead of -10, -35, and 55 it shows -15, -50, and 70 degrees respectively. This has nothing to do with increasing the number of angle point or virtual antennas, I have tried both.

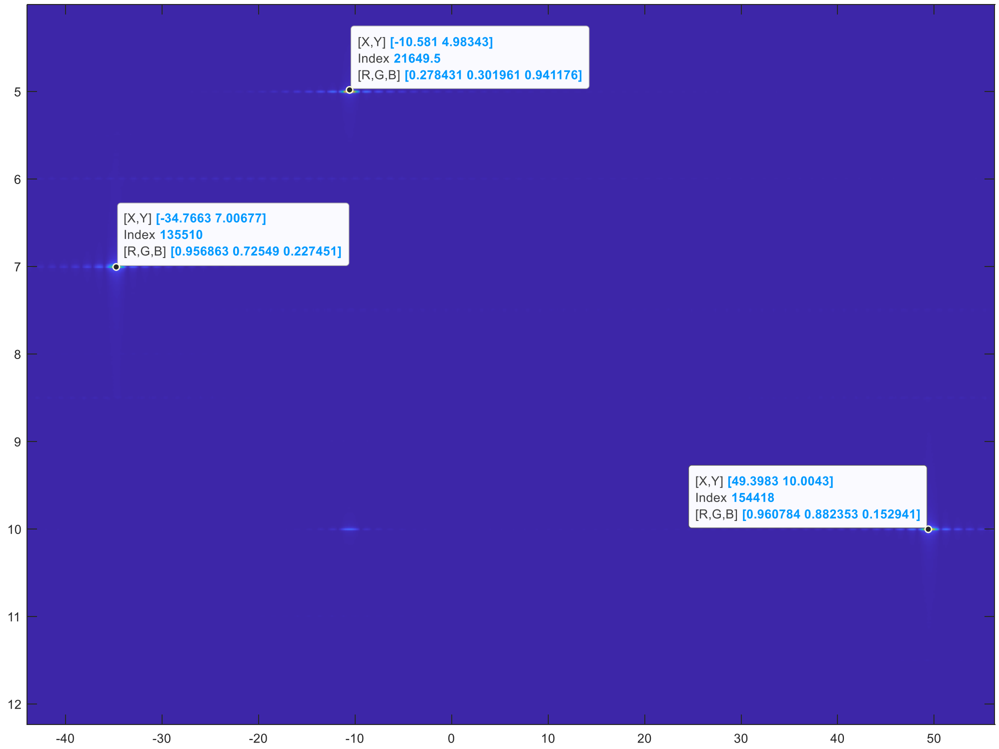

Fig. 7. Correct results after manipulating the axes.

When I divide theta by 1.43 and divide range by 2 (imagesc(theta/1.43,rng(4002:end)/2,abs(QQ(4002:end,:)))) then I get the right answer. Any idea why this happens?