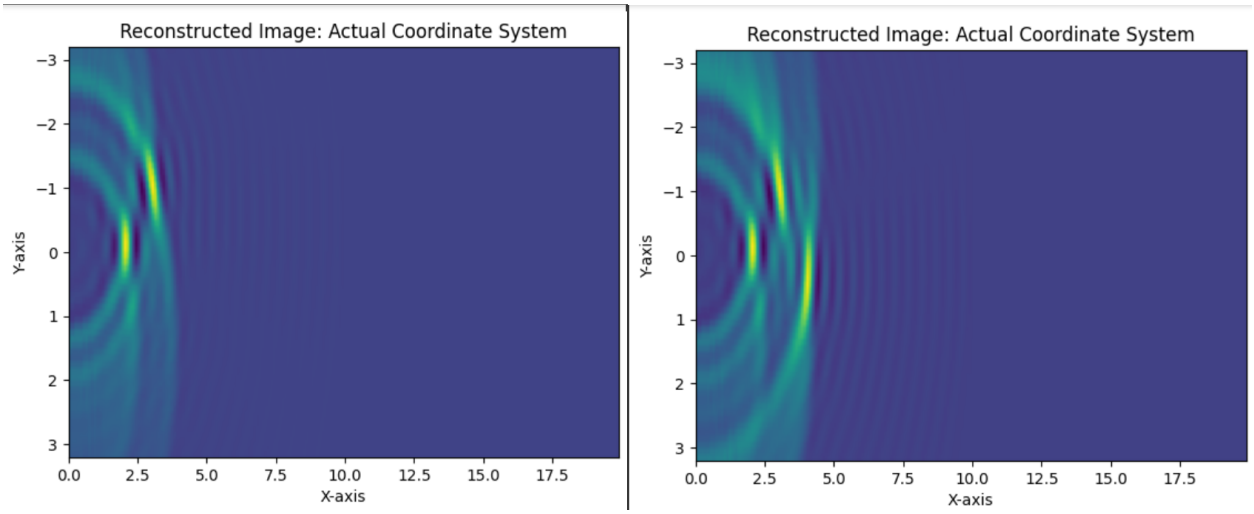
Sound Localisation : EE23B060

November 6, 2024

1 Introduction

The code involves reconstructing an image using the Delay-and-Sum (DAS) algorithm that is commonly used in Ultrasound image reconstruction.

2 Heatmaps of rx2.txt and rx3.txt



The locations of obstacles in actual physical coordinates are approximately

- For rx2.txt data: (2, 0) and (3, -1)
- For rx3.txt data: (2, 0), (3, -1) and (4, 0.5)

3 Answer to Questions

QUESTION 1: The plots below show two example sinc pulses. How will you generate pulses that look like this? Which parameter should be changed? What effect do you think this will have on the final image?

- SincP value: The SincP value adjusts the width of the sinc pulse. Larger the SincP value, narrower the pulse, while a smaller SincP value results in a wider pulse with fewer oscillations.
- Range of X-axis: The product of dist_per_samp and Nsamp determines the range of x-axis values. For instance, keeping Nsamp constant and increasing dist_per_samp from 0.05 to 0.1 will double the x-axis range we capture in the plot.

- The smoothness of the graph depends on Nsamp, higher the sample points, smoother the curve.
- Effect on the final image: Higher SincP value causes a narrow pulse, with higher number of oscillation. This results in the final image being sharper. While a smaller SincP value results in a very distorted final image.

QUESTION 2: Does it make sense to reconstruct up to Nsamp? What value is more reasonable as an upper limit for the x-axis here?

- Nsamp is not representing any distance, it signifies the number of samples collected per microphone. Reconstructing it upto Nsamp doesn't completely make sense, but it can be seen as having `dist_per_samp = 1` while calculating the points in the x-axis for the location of an unknown obstacle.
- A more reasonable upper limit for the x-axis would be: `Nsamp * dist_per_samp`. This represents the total distance of the Nsamps, giving us the optimum maximum distance along the x-axis. This would represent the physical distance instead of number of samples.

QUESTION 3: The (x, y) coordinates corresponding to the maximum amplitude (yellow colour) is approximately (30, 22). Explain why this is the correct expected position for the given obstacle.

- The maximum amplitude appears at (30,22) in the heatmap, while the actual obstacle location is (3,-1). These two points mean the same. While reconstructing, the points defined in the grids don't represent the actual physical coordinates, instead represent indices within the defined grid. The values these "boxes" or locations in the grid give us the physical coordinates (approximately).
- At this point, the signal's delays gets added up(constructive interference) and has the highest value for this sum, i.e., maximum amplitude in the grid.

QUESTION 4: What is the maximum obstacle x- and y- coordinate that you can use and still have an image reconstructed?

- The maximum coordinates for positioning an obstacle:
  - **X-axis**: The maximum distance that sound can travel is `Nsamp * dist_per_samp`, in order to sample a point there.
  - **Y-axis**: The microphone array along the Y-axis goes from `- Nmics * pitch / 2` to `Nmics * pitch / 2`.The maximum Y-coordinate is the uppermost microphone, i.e., `Nmics * pitch / 2`.

QUESTION 5: What happens if C is different - if C is decreased it looks like the image becomes sharper. Can you explain why intuitively?

- Lowering C increases the `time per sample`, this allows us to see better in smaller time intervals as differentiating between closer points becomes effective, this results in a sharper image.

QUESTION 6: What happens if Nmics is increased or decreased? Do the experiments with Nmics = [8, 32, 64] and Nsamp = [50, 100, 200] (all combinations).

- Increasing `Nmics` improves spatial resolution, and increases the amount of data points and thus produces a clear and more accurate reconstruction. While lower values produces blurrier pictures.

- Increasing `Nsamp` improves temporal resolution, this also increases the amount of data points, producing better resolved images. The images for all possible combinations of the given Nmics and Nsamp are attached in this report.

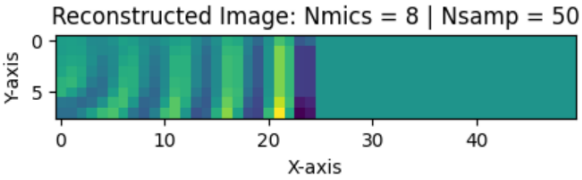
4 How to run my code

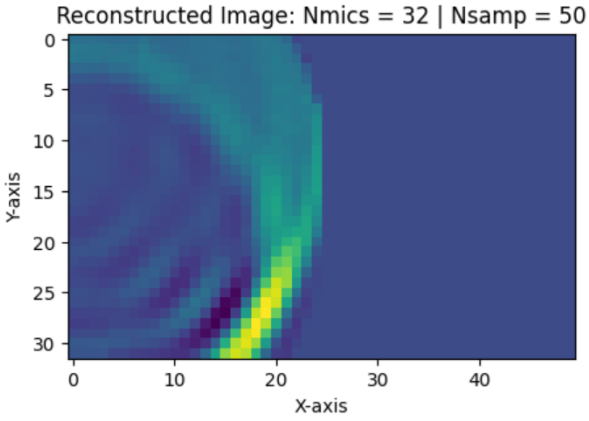
:

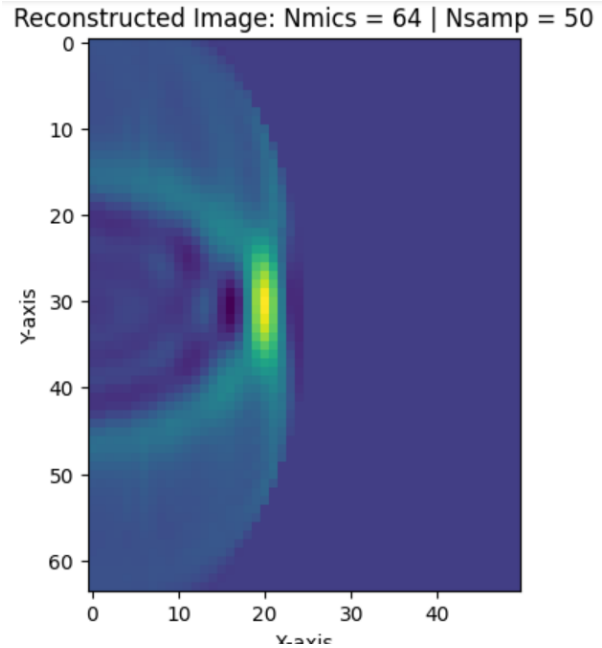
- The first part of the code plots the sinc pulse. Changing the parameters, and calling the function **plotting** will generate the pulse accordingly.
- The second part of the code, generates different plots for a given obstacle, the function **obstacle locator** can be called with any obstacle to plot its heatmaps.
- The third part of the code, loads data from a file and generates different heatmaps. The path to the file must be given correctly in the function call of **obstacle locator**.
- Running the entire python file at once shouldn't give any issues.

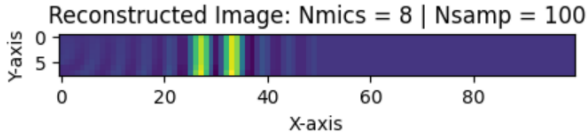
5 References

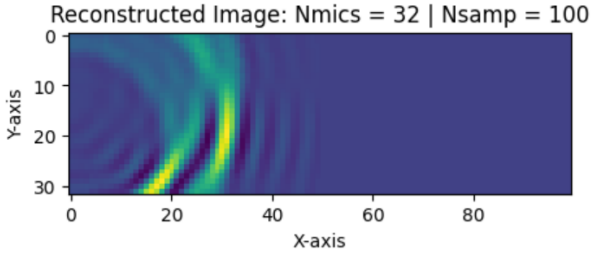
Introduction to Beamforming Part 1 - Medium

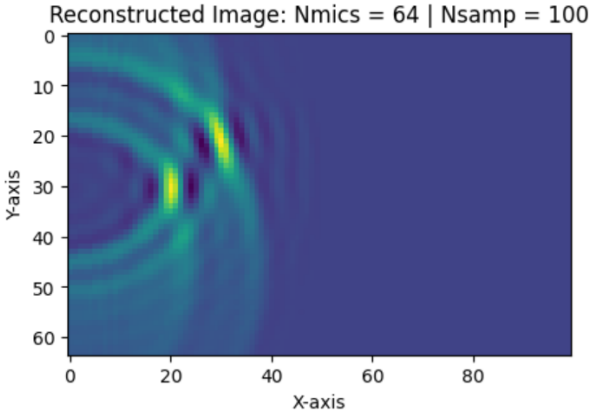

Reconstructed Image: Nmics = 8 | Nsamp = 50
Y-axis
X-axis

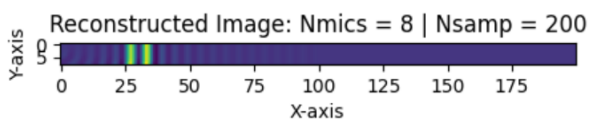

Reconstructed Image: Nmics = 32 | Nsamp = 50
Y-axis
X-axis

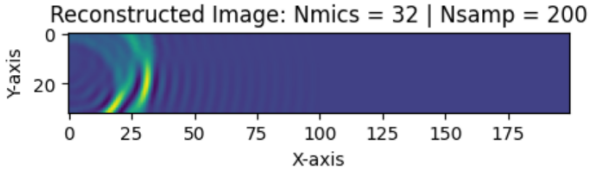

Reconstructed Image: Nmics = 64 | Nsamp = 50
Y-axis


Reconstructed Image: Nmics = 8 | Nsamp = 100
Y-axis
X-axis


Reconstructed Image: Nmics = 32 | Nsamp = 100
Y-axis
X-axis


Reconstructed Image: Nmics = 64 | Nsamp = 100
Y-axis
X-axis


Reconstructed Image: Nmics = 8 | Nsamp = 200
Y-axis
X-axis


Reconstructed Image: Nmics = 32 | Nsamp = 200
Y-axis
X-axis