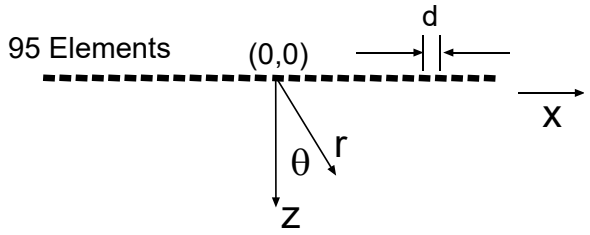
Ultrasound Project

Recommended due date: Oct. 24, 2023



In this project, you will implement the beamforming and scan conversion algorithms for a simulated ultrasound system (above). This system is a little different than most discussed in class in that this system uses only a single transmit element (the (0,0) element at the center of the array) which illuminates all reflectors in the object and the receive array focusing and beam steering is used to provide all lateral localization – that is, we will use one data set to create the entire image of the object. System description:

Number of elements = 95
Operating frequency (f_0) = 5 MHz
Element spacing (d) = $\lambda/2$

The simulated data was generated according to the following assumptions:

Speed of sound (c) = 1.54 mm/μs
No attenuation (no gain compensation needed)
Point receivers (w infinitely small)

The simulated echoes are sampled at f_s = 20 MHz yielding a 1800 point waveform for each of the 95 array elements. I will provide you with this 1800 x 95 matrix. Your task is to take this data and generate a 90° sector scan image using by following the steps described below.

Administrative items (read this!):

- You may choose a partner for this project. You must have a different partner for each project. You may choose not to work with a partner.
- If you work with a partner, please put your partner's name on your solutions.
- I expect you to do your own work, but when you get stuck, you may ask your partner or the instructor for help. Think of this as an exam, but one in which one may ask for help from one other person (and the instructor).
- Getting help from anyone other than the instructor or your partner or copying solutions (even from your partner) will be considered a violation of the engineering honor code. (see https://bulletin.engin.umich.edu/rules/)
- Each person must hand in his/her own solution.
- Title and label the axes on all plots and images (do this by hand if you like). It is important to me to know that you understand what you are looking at.
- Create a single pdf document of your answers. Include the .m files at the end of your pdf document. This should be uploaded to gradescope by the deadline.
- Grading: 60% correct Matlab code, 30% report content, 10% plots, labeling axes, etc.
- Files to get you started can be found on Canvas under Files/US project.
- If you are having problems, please come and see me or Haowei (or send e-mail).
- Hope you feel that you have learn something useful in this project.
- Good luck on your project and have fun!

Questions/Tasks:

Prepare a report describing the tasks below. Hand in answers to questions below, all requested plots/graphs/images and copies of your final ".m" files. For each question/task please write 2 or 3 sentences about what you did, what equation you implemented, and about the appearance of the resultant images, etc. Say something interesting!

a. Make a "wavefield" plot of the data using a linear scales. We will provide software to image the wavefield. The wavefield is defined as a 1800 x 95 image of the raw ultrasonic data. The linear gray scale is setup so that zero pressure is midgray, positive pressure is white and negative pressure is black. (Your comments here can be combined with part n.)
b. Define the number of total beams and the beam spacing to properly sample a 90° sector.
c. Define time delays (i.e., sampling delays $\tau_n{}'$) as a function of range (r) for each element and beam direction (θ). What formulae were implemented?
d. I have placed 2 solitary reflectors along $\theta = 0$ at range/depth locations r = 9 and 18 mm. We create the $\theta = 0$ beam by computing the coherent sum across the array using the delays derived in part c. (shift the pulse by the discretized delay $[\tau_n']$). Before combining data from the detector elements, make plots of delayed shifted pulse from several of the receive elements (e.g. 12, 30 and 47 (middle)) to insure that pulses and waves will add constructively for those two reflectors. Plot time the time vectors before and after the shifts.
e. Create data for the r-sinθ array using all receiving elements for all beams. Use showimage2 function to display results using a 40 dB log scale. Do this for all r up to r_{max} = 69 mm. (Hint: to calculate coherent sum, sum Data($t + [\tau_n']$) across all the elements.)
f. Implement a complex demodulator to convert the data to "baseband" complex representation of the signal for each element (see description in notes, below). For post baseband conversion, plot real and imaginary parts. Is this what you expect?
g. In addition to the time delays ($[\tau_n']$ - part c.), calculate the phase rotations as a function of r for each element and beam direction. Again, what formulae were implemented?
h. Again look at locations r = 9 and 18 mm for the θ = 0 beam to make sure that signals from elements 12, 30 and 47 are phased aligned so that they add constructively.
i. Create data for the r-sinθ buffer for all elements using the baseband converted data, shifted by the time delay and corrected by the phase rotation. Display results r-sinθ for all elements and all beams on 40 dB scale. Is this image better than the result from part e. and why?
j. To look at the issue of aliasing (grating lobes), construct the r-sinθ buffer for all beams using only the odd numbered element from the baseband data. Display results on 40 dB scale. What sorts of artifacts do you see and why?
k. To look at the issue of temporal sampling, keep only every 3rd temporal sample (after demodulation) so that your data array is 600 x 95, and then construct the r-sinθ buffer for all beams. Display results on 40 dB scale. What sorts of artifacts do you see and why?
l. Scan convert the r-sinθ buffer to produce a sector scan image for a 512 by 512 pixel grid. You should display the images over a log scale using both 40 dB and 20 dB using showimage2. Make the pixel dimension Δx=Δz=(70/512)mm, about 137 μm per pixel. Do this for your answers to e., i., j. and k.
m. What is the object? Use the 20 dB image from part i. to answer this question. What artifacts are present? Use the 40 dB display to answer this. What does the point spread function look like? Also discuss differences in the artifacts between the images created by the processing of parts e., i., j. and k.
n. Based on your new knowledge of what the object is, go back to the wavefield plot of part a. and discuss why it looks the way it does. Discuss both the angle and curvature of reflections off of the 4 point objects (two above and two below the main object) and what that says about their location.

Files for ultrasound project:

`Data23.mat`	RF ultrasound data of Mystery Object
`template_beamform.m`	A Matlab script template for doing the receive-only beamforming reconstruction (for questions A-K)
`template_scancon.m`	A Matlab script template for doing the scan conversion for questions l. through n.
`showimage3.m`	image display utility, also converts skinny matrices as 'panels'

Hints:

1. Beamforming signal reconstruction part e. - Small delay to shift envelope/baseband signal, use eqn:

$$v(t) = \frac{1}{N}\sum_{n=1}^{N} v_n\,(t + [\tau'])$$

Where v_n is Data.

2. In part f: One way to baseband convert the data is the multiply the sampled signal for each element by $\cos(\omega_0 t) + i\sin(\omega_0 t) = e^{i\omega_0 t}$ and then low pass filter the resultant signal. An easy way to low-pass filter each element's signal is the take the `fft` (e.g. using function `ft`), multiply by a rect filter keeping only the low frequencies (zeroing out the high frequencies), and then taking the `ifft` (e.g. using function `ift`).

3. Beamforming signal reconstruction step i. – Add phase rotation to step 1, use eqn

$$b(t) = \frac{1}{N}\sum_{n=1}^{N} b_n\,(t + [\tau_n'])e^{\{\mp i\omega_0\tau_n'\}}$$

Where b_n is databb after baseband conversion. The sign ($\pm$) of the phase depends on the sign used in the baseband conversion (e.g. $e^{\pm i\omega_0 t}$).

4. showimage3.m help you display the image in panels (2nd argument), "showimage2(Data,4)" will split the time dimension across 4 panels which is useful because the time dimension is much longer than the array dimension. This function will also produce images in log scale as well (3rd argument), e.g. "showimage3(rsdata,4,40)" will make 4 panels with 40 dB from min to max gray values. You can also add arguments to scale the axes to particular units, if you like.

5. You are given three .m files. You only need to fill the lines for template_beamform.m and template_scancon.m. showimage3.m is complete and can be used as functions directly, more or less as implemented in the templates.

6. For scan conversion, Matlab has existing functions (`interp2`) to perform the bilinear interpolation we discussed in class. So your task is simple, find out the axes values (r-sinθ) for using `interp2`. However, `interp2` function is very sensitive. Type "`help interp2`", will give you something like this:

```
ZI = INTERP2(X,Y,Z,XI,YI)
```

Note that X, Y and Z should all be the same dimensions and X and Y should be produced using

ndgrid or meshgrid. X is the matrix of x coordinates of the input data, Y is the matrix of y coordinates of the input data, and Z is the input data. XI and YI are matrices of coordinates for the output array, but importantly, are given in the coordinate system of the input data. Note that XI, YI, and ZI should all be the same dimensions.

A related function is `griddata`. Some may find this easier to use.

7. General Matlab Hints. Some of you may be new to Matlab, so here are some useful Matlab functions. These functions may help you, but not necessarily needed for this assignment.

```
help    = get help on Matlab commands
load    = load data into memory
size    = get size of a matrix
length   = get length of a vector
ones    = create a vector or matrix filled with 1's
zeros    = create a vector or matrix filled with 0's
sqrt    = compute square roots
exp     = compute exponentials
abs     = compute absolute value (or magnitude of complex value)
sum     = sum the elements of a vector/matrix
angle    = compute phase of complex value
find    = find indices of non-zero elements of a vector/matrix
floor,ceil,round,fix = round or truncate values to integers
meshgrid,ndgrid = transform 2 vectors into 2 array grids
fft, ifft = forward and inverse discrete Fourier transforms
ft, ift = utility functions (previously provided) with fftshifts
```

Warning! In Matlab :

```
 '  (single quote) is Hermitian (conjugate) transpose
 .' (period single quote) is transpose
This is an important distinction when using complex values.
```

In Matlab, all math functions are geared toward matrix math, so

```
 *  (asterisk) is matrix multiplication
 .* (period asterisk) is element-by-element multiplication
The same is true for / vs ./
```

Matlab runs fastest when the number of loops is minimized.