BME 8910 Bioacoustics and Ultrasonic Imaging
Assignment #2 (TK pts.)
Due 9/30/19** (I got it to you late, but I also didn't think you would start right away)

Assignment 2: Ultrasound Beamforming and some Fourier Transforms

Part 1) We now have an idea that somehow wave propagation from one point to another is related to the Fourier transform. The questions below ask you to draw relative outputs from several different scenarios so you may want to read all of the scenarios before starting (18 pts.)

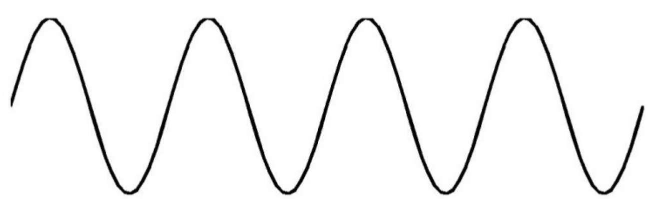
a.) If you have two really small ultrasound transducers what do you expect the pressure field to look like far away from your two transducers. Make a sketch. (3 pts.)
b.) Make a new sketch of the pressure wave far away from the transducers if I move the transducers closer together. (3 pts.)
c.) Make a third sketch of the ultrasound transducers located farther apart. (3 pts.)
d.) What if I only have a single ultrasound transducer, but it's not really small. What if the transducer is just kind of small? Make a sketch. (3 pts.)
e.) What if the ultrasound transducer is even bigger? (3 pts.)
f.) What if you reintroduce the second transducer and it's the same size as the transducer in d? Make a sketch. (3 pts.)

Part 2) Here are some more questions about Fourier Transforms. Feel free to look at equations (including the FourierTransform (I still need to post this, but it seems like you guys are generally comfortable with Fourier Transforms) hand out posted on brightspace), but do not use any calculations to arrive at your answer. (12 pts.)

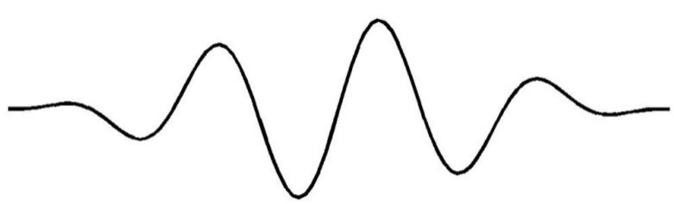
a.) Sketch the shape and write an equation for a Gaussian function. (3 pts.)
b.) What's the Fourier Transform of a Gaussian Function? (3 pts.)
c.) What kind of function do you get when multiplying any Gaussian Function by any other Gaussian Function? You can do the algebra for this if you need to. (If you do the algebra think back on how you can manipulate polynomial equations to "complete the square".) (3 pts.)
d.) What is the functional form of a Gaussian function convolved with a Gaussian function. Don't try to do this analytically. Just think about what you know about Gaussian multiplication, convolutions and Fourier transforms. (3 pts.)

Part 3) A hydrophone is an acoustic device for measuring pressure waves in fluids. One can translate a hydrophone in water to map out the spatial extent of a pressure field. What would you expect the source of the pressure waves to look like if the pressure field measured by the hydrophone looks like the following: (12 pts.)

a.) (3 pts.)



b.) Note: examine this function carefully. It is not a sinc function (3 pts.)



c.) Assuming the hydrophone measurements in part a and b are performed at the same depth and lateral locations (whatever those may be), do you expect the acoustic source used to make part a or part be to be wider? (3 pts.)

d.) What specific aspect of the above pressure fields would provide information about the exact location of the sources relative to the hydrophone measurements? (3 pts.)

Now things get a bit more real…

There are two MAT-files posted on brightspace (pointTargetData.mat and anechoicCystData.mat). Each MAT-file contains a structure called veraStrct. The contents of the structure can be accessed by typing the name into the command window. The variable elementSpacingMM is the pitch. The variable XMTspacingMM is the distance between beams. In this case the beam moves over one transducer element each time. The sampling rate is given and the time sample that corresponds to 0 seconds (i.e. the onset of receive) is given as well. The data is ~2400x128x128. The first dimension is the time/depth dimension for an individual channel. The second dimension indexes the channels, and the third dimension indexes the lateral location of each beam. In this particular data set I've only given you the active receive aperture. The transducer has 256 elements, but only 128 are used at a time.

BME 8910 Bioacoustics and Ultrasonic Imaging
Assignment #2 (TK pts.)
Due 9/30/19** (I got it to you late, but I also didn't think you would start right away)
The Verasonics scanner

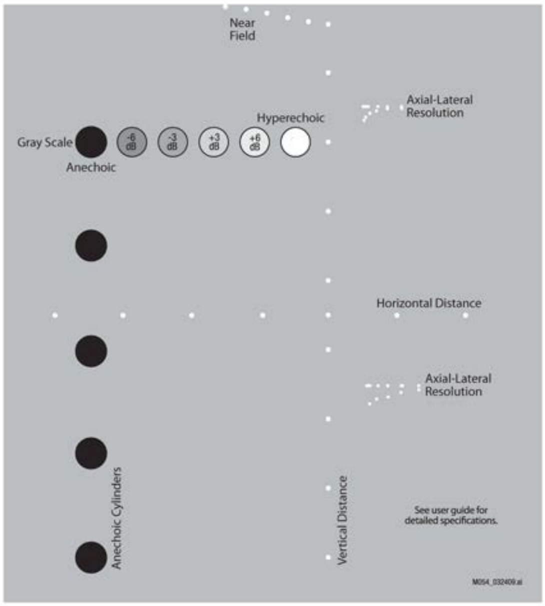


Fig. 2—The structures inside the phantom used for data acquisition are shown above. Data were acquired of the first anechoic lesion (i.e. the black circles on the left) and of the vertically oriented point targets in the middle of the phantom. This visualization doesn't reflect the speckle generated by random scattering. **Actually, your data were derived from a different phantom, but I kept this to give you a sense of what kind of structures might appear in an ultrasound phantom.**

The data for this assignment were acquired using a Verasonics scanner. Verasonics produces flexible, commercially available ultrasound scanners designed for prototyping unique ultrasound pulse sequences. The Verasonics in general has lower image quality compared to clinical scanners made by companies like GE, Philips, Siemens, Toshiba, etc. The decrease in quality is particularly apparent when looking at real time images. As a research tool for rapid prototyping the decrease in signal quality is considered acceptable. One of the primary uses of the Verasonics is that it allows for individual receive channels to be acquired per transmit event. (Acquiring single channel data on a clinical scanner is possible, but it represents a significant challenge with unavoidable drawbacks.)

Raw Ultrasound Data

Ultrasound channel data for this assignment can be found on brightspace. The data used here were acquired from an old ATL L12-5 50mm transducer. This is a linear array transducer. The sequence was developed with this class in mind so it is a little simpler than what you would find on a clinical scanner. Basically, each line of the image (A-line) is formed by focusing straight down with a symmetric aperture. This means the center of the beam is right in between transducer elements 64 and 65.

The data that makes up the crux of this assignment was acquired from the same CIRS phantom used in the ultrasound demo during the first week of class. **I recommend starting and debugging your code with the point target data set because it is easy to see the wavefronts originating from the bright point targets. It is easy to look at the wavefront from point targets to determine whether the delays are applied correctly. i.e. the wavefront of a point target should look flat. I also recommend starting early. It'll take a few tries to get everything right.**

For the following problems make sure to create your images with appropriately scaled axes. Basically instead of just running imagesc(data), use imagesc(x,y,data); axis image;

Part 4.) **Single Fixed Focus Beamforming**

In this problem you will implement an ultrasound image with a single receive focus. We call this a fixed focus system. The image data extends down to about 8 cm so 4 cm is a reasonable depth for your focus. You'll need to apply the appropriate delays for each beam. Apply the delays discretely, which means that you will need to uniformly upsample the data (i.e. linearly (or spline-ly?) interpolate evenly spaced points between each original sample). Next write out the explicit delay equation that you will use for

BME 8910 Bioacoustics and Ultrasonic Imaging
Assignment #2 (TK pts.)
Due 9/30/19** (I got it to you late, but I also didn't think you would start right away)
this beamforming problem. (Keep in mind that you'll want to adjust the applied delays so that the "maximum" delay is zero.) Use your equation and convert your time-delays into discrete shifts keeping in mind your upsampled sampling frequency. Finally, apply the discrete delays appropriate for each channel by taking away the correct number of samples.

Make an image of the channel data. Use colormap gray and saturate your image so that the wavefronts are clear. Also zoom in enough that you can identify the wavefronts clearly. (5 pts.)
Select one of the middle lateral locations (~beam 64 or 65), apply the delays and then make an image of the delayed (but unsummed) channel data **near a point target**. (5 pts.)
Apply the delays to all beams and sum across the channels (not across depth), and then make a compressed B-Mode image. i.e. (20*log10(abs(Hilbert(imageData))). Do this for both the anechoic cyst data and the point target data (10 pts.)
Good, now we're all beamformers! (0 pts. For this assignment but +10 pts. towards winning at life.)

Part 5.) **Continuous Delay and Sum**
Repeat part 4, but calculate a new delay profile for each depth sample. This is true dynamic receive delayed data. We'll talk more in class about how to do this effectively. Use interp1 to delay the data appropriately. delayedChannel = interp1(timeArray+delay, channelData, timeArray, 'linear'). Make compressed envelope images for both cases. Briefly describe the differences in the images made in questions 2-4. (25 pts.)

Part 6.) **Parallel Receive Beamforming**
Adapt the dynamic receive beamforming code from part 5 to perform parallel receive beamforming. For this case you'll need to consider where the lateral focus should be, and you'll need to eliminate some of the beams in the third dimension. Make images with 2, 4, 8, and 16 parallel beams. How many parallel receive beams can you introduce before you start to notice artifacts in your images (20 pts.)?

Part 7) **Apodization**

a.) Now use your code from part 5 in assignment 1 (i.e. dynamic receive beamforming), but this time we'll implement apodization and determine how it affects lesion contrast. Start with your code for dynamic receive beamforming. Next, using the cyst data make an image with a rectangular aperture, and make two more images with window functions of your choice. Make sure to specify your choices. Matlab will automatically generate most of the window functions you can find on Wikipedia, and you can find the Matlab specific names by using `help window'. When you make your 3 images display all of them with a constant dynamic range, and this time restrict your data's dynamic range. Typically displaying 50 dB is a good start. If you scale your data to the maximum value in the image then plotting the data as follows is a good way to do this: imagesc(lat, axl, logCompressedScaledImage,[-50 0]). Notice that I also include the lateral and axial dimensions of the window. These are the lateral and axial positions of all of your depth samples and beam locations. If you create these arrays correctly and then use the `axis image' command in Matlab you will create a correct image. This is important because in diagnostic ultrasound, physicians need to know correct spatial relationships. (10 pts.)

b.) Now calculate the contrast and contrast to noise ratio for your lesion for the 3 different apodization windows. You can select arbitrarily shaped regions of a matrix using matlab's

BME 8910 Bioacoustics and Ultrasonic Imaging
Assignment #2 (TK pts.)
Due 9/30/19** (I got it to you late, but I also didn't think you would start right away)

roipoly command. So something like this: maskLesion = roipoly; then outline the lesion. Repeat for a background region. Try to make the background as close in depth and size to the lesion as possible. Report your contrast and CNR's for each case. (10 pts.)

Part 8) **Aperture growth**

a.) Next keep your code from above, pick an apodization window that you like, and implement aperture growth. This means that the size of your aperture expands through depth so that your F/# is constant. To do this you'll need to pick an F/#, 2 is often a good choice. An easy way to implement this in software is to make a binary mask. Make your mask matrix 0 in the regions where the channels should be zeroed out and 1 in the region that should be included in the beamforming. You'll multiply this mask by your delayed channel data before summing. Report your F/# and make an image of your mask. (10 pts.)
b.) Apply your aperture growth mask to your point target data. Make an image of the point targets with and without aperture growth. Show your image and describe any changes from the point targets without aperture growth (10 pts.)