Ultrasound Imaging Summer School 2015

Exercise: Synthetic Aperture Sequential Beamforming

Martin C. Hemmsen and Tommaso Di Ianni
Department of Electrical Engineering
Technical University of Denmark

1. Introduction

This exercise is the first of two exercises related to Ultrasound imaging using Synthetic Aperture Sequential Beamforming. The aim of this exercise is to get hands-on implementing the Synthetic Aperture Sequential Beamforming algorithm.

A complete dataset simulated using Field II is provided. The dataset contains fixed focus beamformed rf-data (first stage beamformed data). Your task is to implement the second stage beamformer and generate a B-mode image.

For a comprehensive description of the algorithm, please refer to '*Synthetic Aperture Sequential Beamforming*', J. Kortbek, J. A. Jensen, and K. L. Gammelmark, in Proc. of IEEE Ultrasonics Symposium, 2008.

2. Exercise

This exercise is based on a linear 192 element array, with a pitch of 0.24 mm. Field II was used to simulate the received response from a phantom with 7 point scatters. The scatters are distributed along the axial dimension starting at 1 cm with a spacing of 1 cm.

The simulation model uses the same parameters that you will use in exercise 2, performing an *in-vivo* scan. The important parameters are given in table 1.

Name	Value
Transmit focus	20 mm
Transmit F#	2
Transmit apodization	Boxcar / rect

No. emissions	192
Receive focus	20 mm
Receive F#	2
Receive apodization	Hamming
Acoustic layer	0.812 mm

Table 1

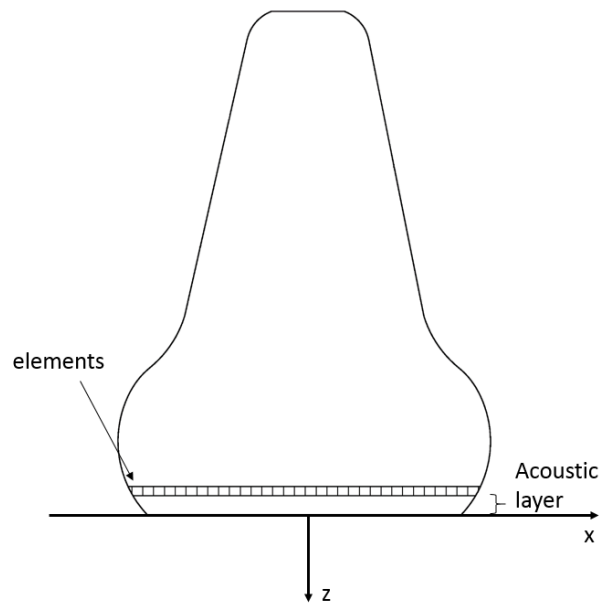The coordinate system used in the simulation, is shown in figure 1.

Figure 1

Note that the origin of the coordinate system is at the shell of the transducer, and not at the elements. The focusing parameters given in table 1 are relative to the elements.

For each emission the received element responses are beamformed with a fixed focus. The scan setup for one of the central scan lines is visualized in figure 2.
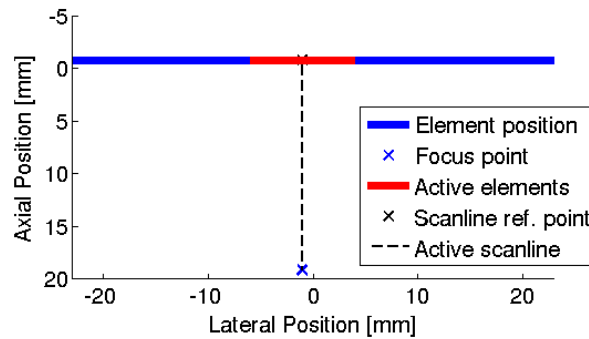

Figure 2

Note that the scan line reference position is located on one of the elements. Also note that the focus point is directly below and 20 mm away from the scan line reference position.

Task 1: Load the file called First_Stage_RF_Data_Beamformed.mat

The file contains two variables, RFdata and xmt. RFdata contains the output from the first stage beamformer. The samples of each scan line are saved along the rows, and each column is a new emission. xmt contains the definitions of the scan lines.

```
xmt =

             no_lines: 192
    scanline_ref_point: [192x3 double]
    scanline_direction: [192x3 double]
         sample_origin: [192x3 double]
                    dr: 2.5667e-05
           focus_point: [192x3 double]
               smpfreq: 30000000
```

The field called sample_origin, represents the position in space of the first sample in RFdata. The field called dr, represents the distance between two samples.

Task 2: Calculate the axial position of all the samples and plot the first stage data as a B-mode image. Your result should look like figure 3. The distance should be relative to the scan line reference position, i.e. consider the acoustic layer.
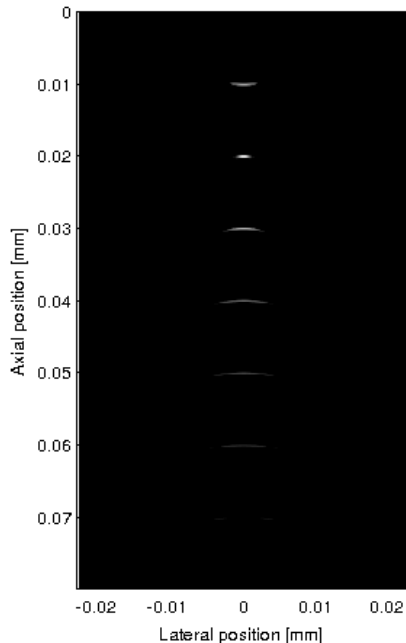

Figure 3

Task 3: Define a set of image points.

You are now ready to define a set of image points. Define a coarse grid while you develop the algorithm, and then later on you can increase the number of pixels.

Define a 3x101 image with 3 image lines symmetrically distributed around x equal to 0.
Hint: Use meshgrid

Task 4: Second stage beamforming

Each sample in first stage beamformed scan line contain information from a set of spatial positions indicated by the arcs showed in Figure 4. With this in mind, a high-resolution image can be obtained by refocusing the scan lines. Perform the second stage beamforming to obtain a high-resolution image.
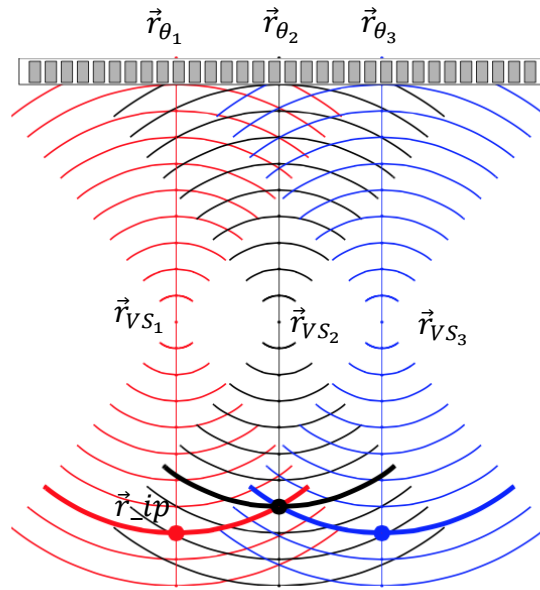

Figure 4

Use Equation 1 and 2 to calculate the high-resolution image.

A single sample, h, representing the image point at location, $\vec{r}_{ip}$, is calculated as

$$h(\vec{r}_{ip}) = \sum_{k=1}^{N} \mathcal{W}(l_{\theta_k}, \vec{r}_{ip}) l_{\theta_k}(t_{d_k}(\vec{r}_{ip})),$$

Equation 1

where $l_{\theta_k}(t_{dk}(\vec{r}_{ip}))$ is the sample at time tdk from the scan line, l, with propagation direction θ_k. The variable W is an apodization function with N values, which controls the weighting of the contribution from each of the N first stage scan lines. The time delay, t_{dk} for the individual scan lines are found from the round trip time-of-flight,

$$t_{d_k}(\vec{r}_{ip}) = \frac{2}{c}\left(|\vec{r}_{VS} - \vec{r}_{\theta_k}| \pm |\vec{r}_{ip} - \vec{r}_{VS}|\right),$$

Equation 2

where r_{θ_k} is the scan line reference position, and $\vec{r}_{VS}$ is the position of the virtual source. Plus/minus referes to if the image point is below or above the virtual source (minus if above/shallower depth).

The variable W is an apodization function determining the weighting of the individual scan lines. The weight for the scan line, l_{θ_k}, contributing to a specific image point can be calculated from

$$\mathcal{W}(l_{\theta_k}, \vec{r}_{ip}) = \begin{cases} 0.54 + 0.46\cos(2n\pi), & \text{if } n \text{ is } \leq 0.5 \\ 0, & \text{if } n \text{ is } > 0.5 \end{cases}$$

for the case of a desired Hamming apodization, where n can be calculated from

$$n = \frac{d(l_{\theta_k}, \vec{r}_{ip})}{\Delta},$$

and is given as the ratio between the distance from the image point to the scan line, and a desired aperture width, Δ. The aperture width is relative to the axial distance from the image point to the virtual source, divided by the F#.

Use this template:

```
%% Task 4
IQdata = hilbert(RFdata);

f_num = 2;
c = 1540;

% Loop over all scan lines
for k = 1:xmt.no_lines
    for x_id = 1:size(x_img,2) % loop in lateral direction
        for z_id = 1:size(x_img,1) % loop in axial direction
```

```
            % determine axial distance from image point to virtual source
            dz = ;

            % determine lateral distance from image point to scan line
            dx = ;

            % determine apodization value
            delta = dz/f_num;
            n = dx/delta;
            if(n <= 0.5)

            else

            end

            % determine time of flight (tof)
            if(z_img(z_id,x_id) >= xmt.focus_point(k,3))

            else

            end

            % interpolate with tof into the IQdata
            yi = interp1(z, IQdata(:,k),zi);
            if(~isnan(yi))

            end
        end
    end
end
```

Task 5: Display a high-resolution image 500 x 500 pixels

Display the high-resolution image obtained with the second stage beamformer with the correct axes. Data must be converted to a logarithmic scale (Log compressed) using:

$$y = 20 * \log_{10}(abs(x))$$

and converted to 256 gray tones:

$$y = \left[\left(1 + \frac{x}{dynamic_{range}} \right) * 255 \right]$$

where the range is limited to $dynamic_{range}$ and [ ] is a rounding function.