Exercise 5: Ultrasound imaging - signal processing chain

Purpose: To understand the signal processing chain from beamformed (RF) data to grayscale image.

Matlab code: SignalChainData.zip (downloadable from It's learning) contains beamformed RF data and some Matlab code which may be useful when solving the exercises.

Data: The .zip file contains 3 datasets. The data is acquired using a phased array and beamformed in the RF domain.

```
phantomRFdata1.mat
phantomRFdata2.mat
cardiacRFdata.mat.
```

First steps (no hand-in): Download the zip-file containing the ultrasound data and the Matlab functions from the course web-pages, and unpack the files to a new folder. Load the data into the Matlab workspace using the load command. Display the struct p to see the available parameters:

```
load phantomRFdata1;
p
```

The struct is organized with named fields corresponding to values which might be strings or numbers, for instance the transmit center frequency: f0=p.f0_Hz; All the ultrasound data are stored in beam space format. When using a phased array, this means polar coordinates. The geometric information is important to be able to reconstruct the images. The columns in the data matrix are beams in the image, and every row is a depth sample.

Exercises with hand-in: You will be asked to process the above three datasets using the same code. Try to design your code such that only the file name needs to be changed between datasets.

1. Complex demodulation

   a. Create your own Matlab code for complex demodulation using downmixing and low-pass filtering.

   b. Use your code to convert the RF dataset (phantomRFdata1.mat) into IQ data and store it as a new variable. Use the transmit center frequency as demodulation frequency, and design a suitable low pass filter to remove the negative frequency components of the original RF signal.

   c. Find the power spectrum of the signal before and after demodulation. For each case, average the power over all image lines and plot the spectra in the same figure.

2. Envelope detection
   a. Plot the RF signal from the middle image line as a function of depth. Use c = 1540 when you calculate the depth axis. Note that the start depth is stored in the parameter struct.

   b. Find the envelope of your signal (all image lines) and store this as a new variable. Plot the envelope of the middle image line in the same figure as in 2a. You have to multiply the absolute value of the demodulated signal by 2 to obtain the envelope. Can you explain why?

3. Scanconversion and log-compression
   a. Convert your envelope-detected image from beam space to Cartesian space. This can be done using the function

   ```
   scanconvert().
   ```

   The input to scanconvert is an image in beam space and two vectors specifying the position of the samples, r and θ, and optionally two parameters specifying the number of pixels in the output image. The function returns the reconstructed image, and the axes in Cartesian coordinates. You need to create both the r and θ vectors based on information in the parameter struct p.

   b. The code

   ```
   imagesc( 20*log10( abs( myImage) ) );
   caxis([-dyn 0]-gain);
   ```

   visualizes a log-compressed version of the image. The two parameters are gain ("brightness") and dynamic range ("contrast"). Typically the dynamic range for a B-mode image is 30-70 dB, while the gain is adjusted so that the image is sufficiently bright. Try to find a gain and a dynamic range that provides a sharp image with good contrast. To set the correct colormap, use:

   ```
   colormap(gray);  %dyn = dynamic range
   ```

4. Settings
   Repeat exercise 1-3 for phantomRFdata2.mat. Compare the final grayscale images. One acquisition parameter has changed between the two phantom recordings. Which parameter is this, and how can you tell?

5. Fundamental vs. harmonic imaging
   Repeat exercise 1-3 for cardiacRFdata.mat, using two different demodulation frequencies: Fundamental and twice the fundamental frequency. Try to filter the data such that the fundamental frequency is suppressed when creating the harmonic image. Use the same dynamic range, but adjust the gain individually so that they become equally bright. Comment on any differences in the two images.

Good Luck!