RESOLUTION AND CONTRAST MEASUREMENTS IN ULTRASOUND SYSTEM

Aim:

To determine the axial and lateral resolution as well as the contrast in ultrasound images taken using different probes

Apparatus:

- Ultrasound probes:
    - **Verasonics** L11-5v: 5MHz to 11MHz linear array of 128 transducers
    - **SonixTouch** L14-5: 5MHz to 14 MHz linear array of 128 transducers

- Phantom – CIRS 040GSE

Methodology:

- Place the phantom on a level surface and attach the water well on the top with provided straps.
- Pour water into the well till the entire surface of the phantom is covered
- Attach the ultrasound probe to the stand and adjust its height till all the transducers are immersed in water. Also ensure that the transducer is parallel to the phantom surface
- Take a B-mode image of the area in the phantom marked as axial/lateral resolution group
- Move the probe and take another B mode image of the area in the phantom marked grayscale
- Export the data from the scanner and perform the resolution and contrast analysis

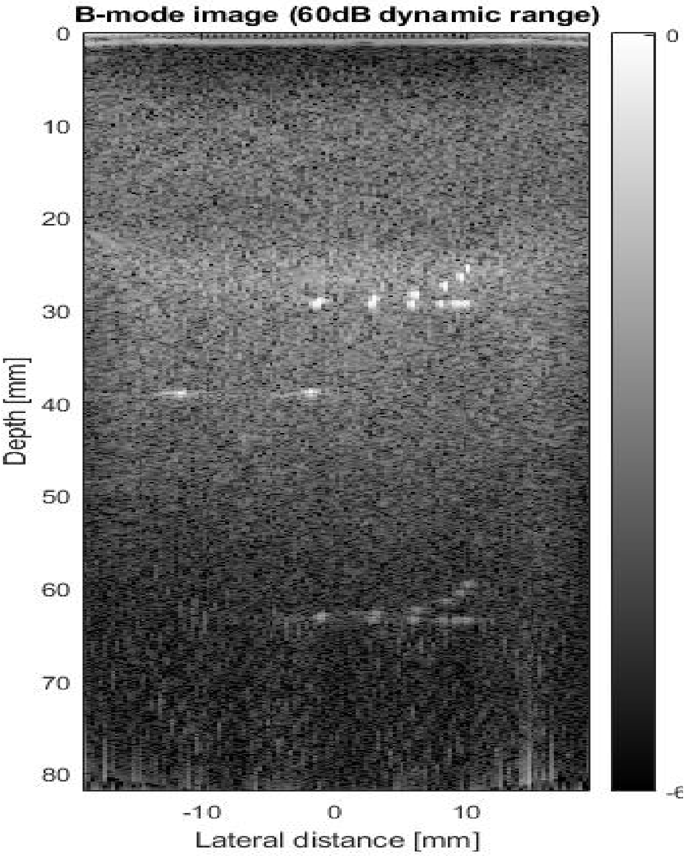
Outcome:

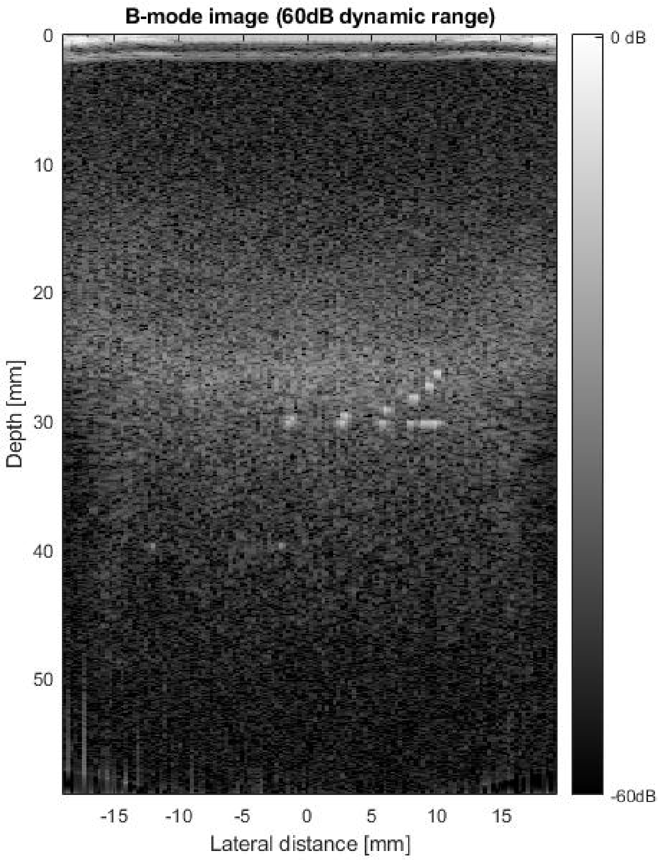
- Report the axial and lateral resolution of both the systems in **mm**
- Report the SNR and CNR values
- Repeat the experiment at different center frequencies and observe& report the effect on resolution and contrast

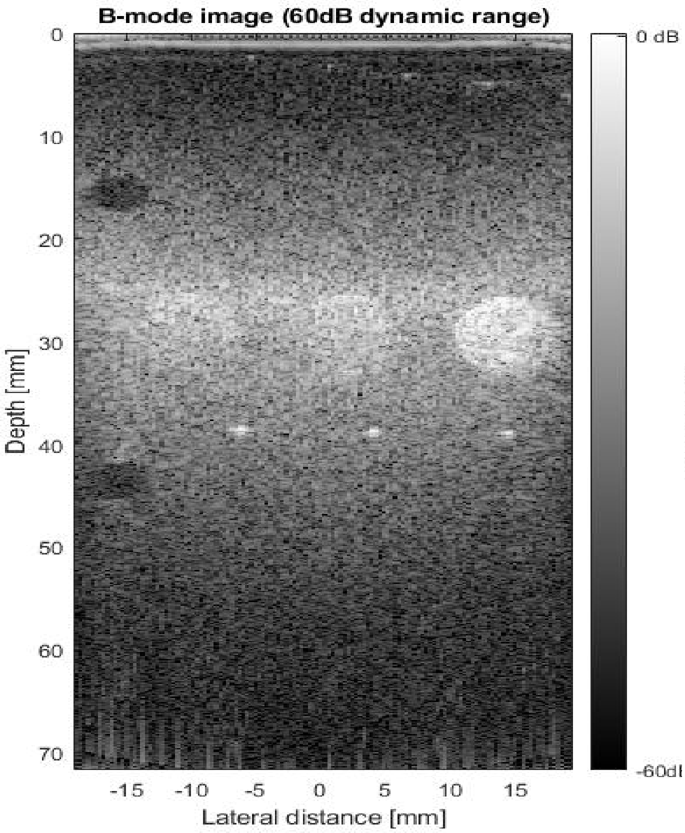
Results:

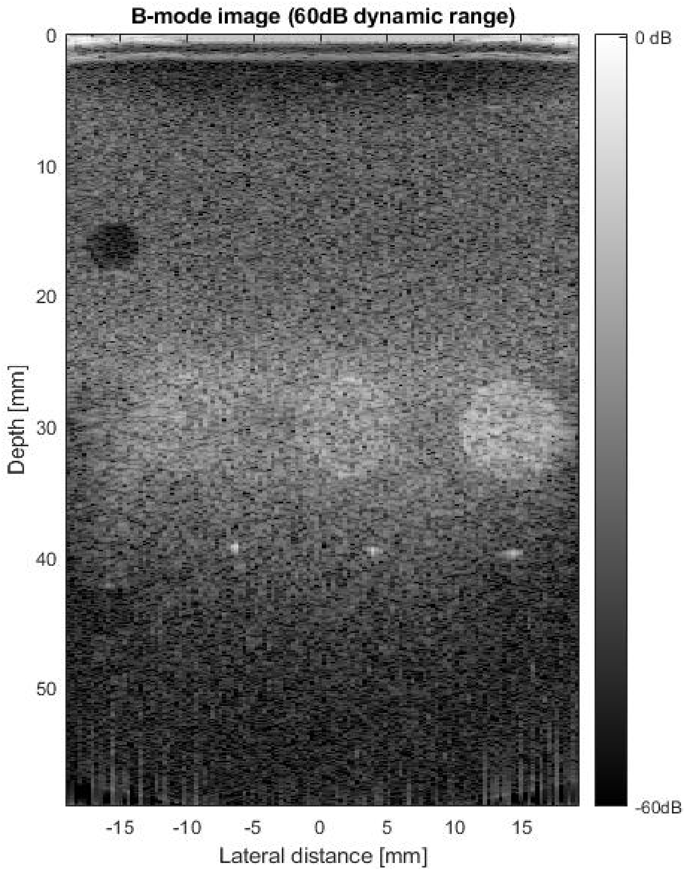
	5 MHz	**10 MHz**
Axial Resolution [mm]	2	1
Lateral Resolution [mm]	1	1
CNR [dB]	+3	+6

- We can see that as frequency increases, the axial resolution increases.
- However, lateral resolution does not change with frequency change as it depends on the beam width and focusing depth.
- Finally, CNR increases with the increase of frequency because attenuation increases as frequency increases, which degrades the contrast the of the image.


B-mode image (60dB dynamic range)
0
10
20
30
40
50
60
70
80
Depth [mm]
-10
0
10
Lateral distance [mm]
0
-6


B-mode image (60dB dynamic range)
0
10
20
30
40
50
Depth [mm]
-15
-10
-5
0
5
10
15
Lateral distance [mm]
0 dB
-60dB


B-mode image (60dB dynamic range)
0
10
20
30
40
50
60
70
Depth [mm]
-15
-10
-5
0
5
10
15
Lateral distance [mm]
0 dB
-60dI


B-mode image (60dB dynamic range)
0
10
20
30
40
50
Depth [mm]
-15
-10
-5
0
5
10
15
Lateral distance [mm]
0 dB
-60dB

MATLAB Code

```
% Load your data (replace 'data_5MHz.mat' and 'data_10MHz.mat' with actual filenames)
load('5mhz.mat'); % Assume this file contains beamformed_data_5MHz and other necessary variables
load('10mhz.mat'); % Assume this file contains beamformed_data_10MHz and other necessary variables
% Parameters (you might need to adjust these based on your data)
db_range = 60; % Dynamic range in dB
samp_sep = 0.1; % Example value, replace with actual sampling separation in mm
pitch = 0.1; % Example value, replace with actual pitch in mm
% Process and display images for both 5 MHz and 10 MHz data
no_lines_5MHz = size(beamformed_data_5MHz, 2); % Replace with the actual number of columns in your data
no_lines_10MHz = size(beamformed_data_10MHz, 2); % Replace with the actual number of columns in your data
process_bmode_image(beamformed_data_5MHz, no_lines_5MHz, db_range, samp_sep, pitch, 'B-mode image (5 MHz)');
process_bmode_image(beamformed_data_10MHz, no_lines_10MHz, db_range, samp_sep, pitch, 'B-mode image (10 MHz)');
roi_axial = [start_idx:end_idx]; % Replace with actual indices for axial resolution area
roi_lateral = [start_idx:end_idx]; % Replace with actual indices for lateral resolution area
roi_background = [start_idx:end_idx]; % Replace with actual indices for background area
% Perform analysis for both 5 MHz and 10 MHz data
analyze_resolution_and_contrast(beamformed_data_5MHz, roi_axial, roi_lateral, roi_background);
analyze_resolution_and_contrast(beamformed_data_10MHz, roi_axial, roi_lateral, roi_background);
% Function definitions must be at the end of the file
function process_bmode_image(beamformed_data, no_lines, db_range, samp_sep, pitch, title_str)
hilbert_transformed_data = zeros(size(beamformed_data));
for line = 1:no_lines
hilbert_transformed_data(:, line) = abs(hilbert(beamformed_data(:, line)));
end
env_dB = 20 * log10(hilbert_transformed_data);
env_dB = env_dB - max(max(env_dB));
```

```
log_compressed_data = (env_dB + db_range) / db_range;
pixel_data = uint8(255 * log_compressed_data / max(log_compressed_data(:)));
depth = (0:size(pixel_data, 1) - 1) * samp_sep;
x = ((1:no_lines) - no_lines / 2) * pitch;
figure;
imagesc(x * 1000, depth * 1000, pixel_data);
xlabel('Lateral distance [mm]');
ylabel('Depth [mm]');
axis('image');
colormap gray;
cb = colorbar;
cb.YTick = [0 255];
cb.YTickLabel = {['-', num2str(db_range), 'dB'], '0 dB'};
title(title_str);
end
function analyze_resolution_and_contrast(image_data, roi_axial, roi_lateral, roi_background)
% Compute axial and lateral resolution from ROI data
axial_resolution = compute_resolution(image_data(roi_axial));
lateral_resolution = compute_resolution(image_data(roi_lateral));

% Compute SNR and CNR
signal_mean = mean(image_data(roi_axial), 'all');
noise_mean = mean(image_data(roi_background), 'all');
noise_std = std(image_data(roi_background), 0, 'all');
snr = 20 * log10(signal_mean / noise_std);
cnr = abs(signal_mean - noise_mean) / noise_std;
fprintf('Axial resolution: %.2f mm\n', axial_resolution);
fprintf('Lateral resolution: %.2f mm\n', lateral_resolution);
fprintf('SNR: %.2f dB\n', snr);
fprintf('CNR: %.2f\n', cnr);
end
function resolution = compute_resolution(roi_data)
% Placeholder function for computing resolution
```

```
% Implement the actual resolution calculation based on your specific method
resolution = mean(roi_data); % Replace with actual computation
end
```

Conclusion:

Resolution and contrast analysis forms the basis for the performance evaluation of any medical imaging system. Careful analysis of these parameters can be beneficial in enhancing system performance and diagnostic capability.