

[RT516] Medical Imaging★

Term Project #2 - US

(2021-06-04)

202123008 Jinmin Kim

Phone: 010-6266-6099

Mail: rlawlsals@dgist.ac.kr



Part I : Design of Linear Array Transducers

Question 0.

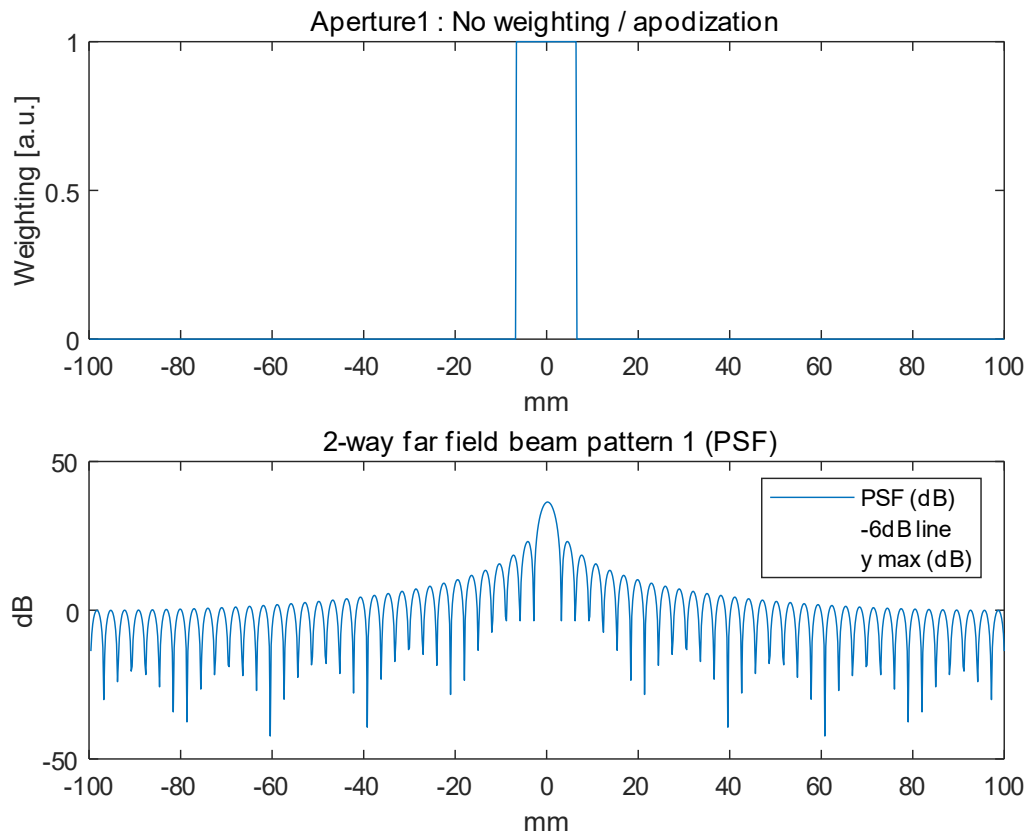
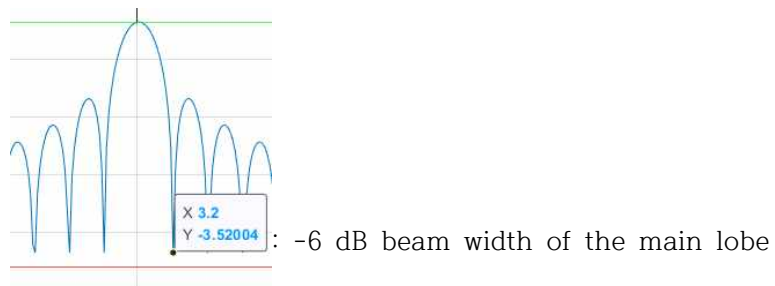


Figure 1. No weighting / apodization

=> weighting과 apodization이 없는 상태에서의 aperture의 그림과 PSF의 근사치입니다.



Question 1.

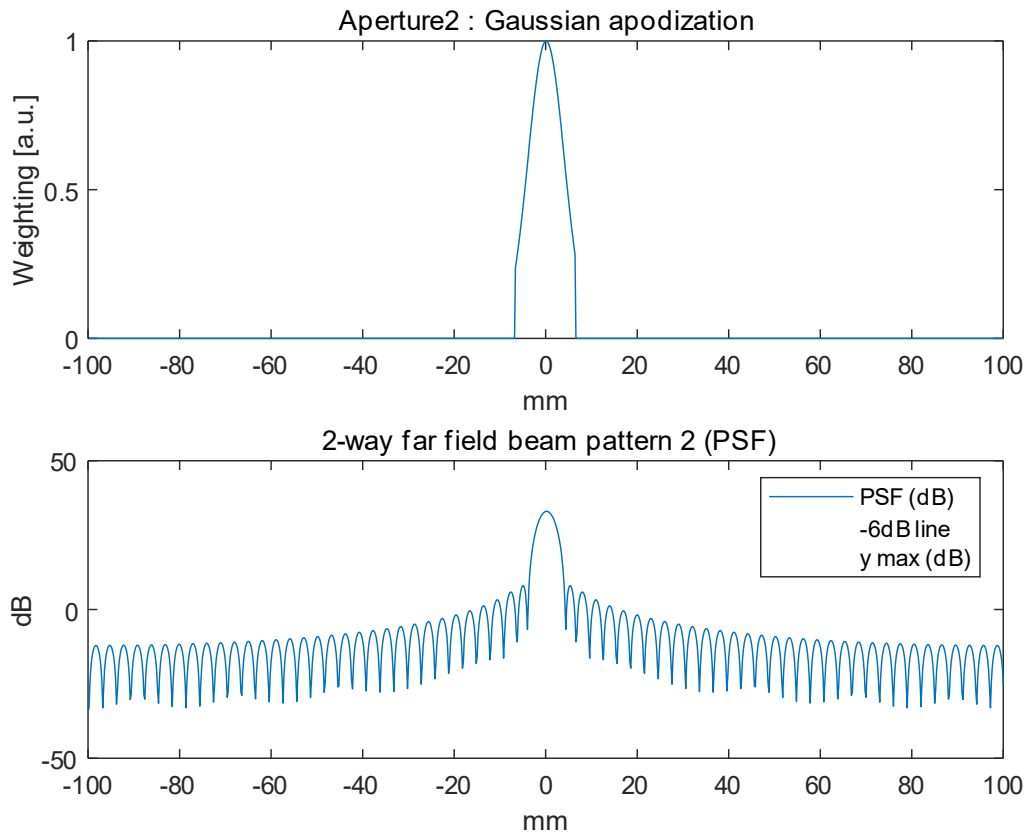
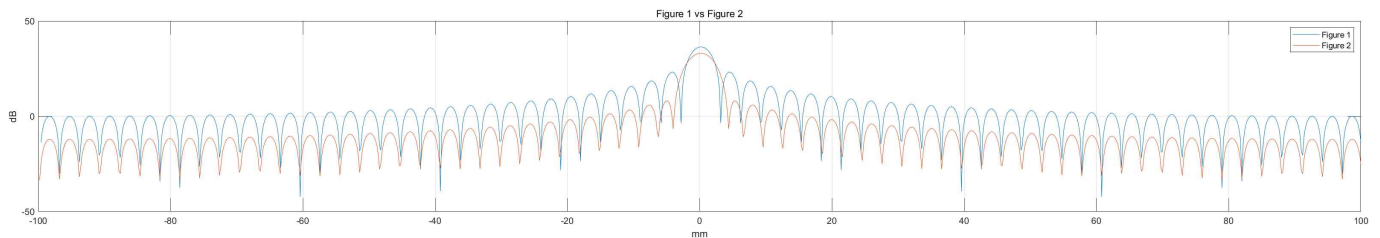
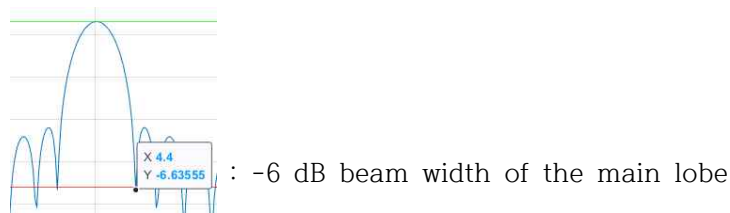


Figure 2. Gaussian apodization

=> aperture의 weighting을 주기 위해 Gaussian apodization을 적용하였습니다. 이를 위해 Matlab의 `gausswin` 함수 라이브러리를 사용하였습니다. 이 라이브러리는 가우스 그래프의 x축 점의 개수를 지정해줄 수 있습니다. 저는 x축 점을 100개 생성하고 center에 가까운 64개의 점을 뽑아서 aperture에 적용하였습니다.



=> Gaussian apodization을 적용시킨 결과, main lobe의 -6dB 빔 폭이 넓어지고(6.4mm -> 8.8mm) sidelobe의 amplitude가 감소하는 결과를 얻을 수 있었습니다. 그리고 peak level of the first sidelobe는 감소(36.3909dB -> 33.0388dB)하는 결과를 얻었습니다.

Question 2.

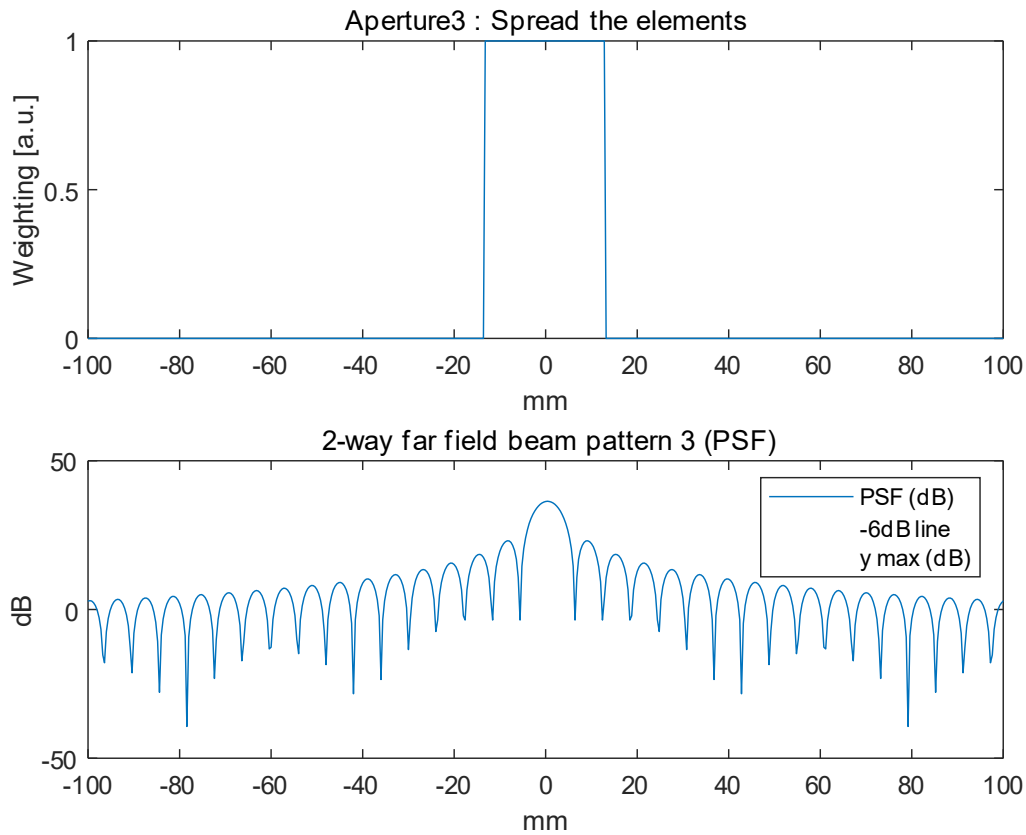
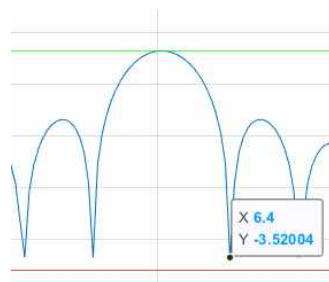
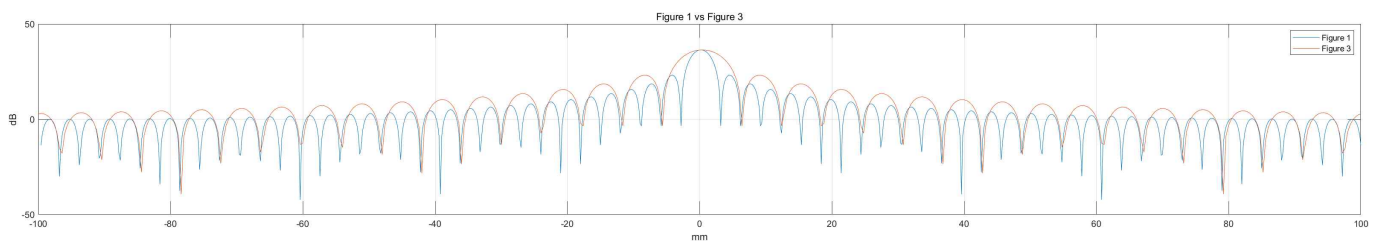


Figure 3. Spread the elements

=> aperture의 인접한 요소들 사이에 간격을 기존보다 2배만큼 분리하였습니다. 64개의 element 배열을 유지하면서 d를 2배로 증가시키다보니, $D(=len*d)$ 가 200에서 400으로 증가하였습니다. 그래프의 비교를 위해 axis의 범위를 -100~100으로 적용시켰습니다.



: -6 dB beam width of the main lobe



=> Spread the elements를 적용시킨 결과 main lobe의 -6dB 빔 폭이 정확히 2배 넓어지고(6.4mm -> 12.8mm) 주기가 2배가 되는 결과를 얻을 수 있었습니다. 그리고 peak level of the first sidelobe는 동일(36.3909dB == 36.3909dB)한 결과를 얻었습니다.

Question 3.

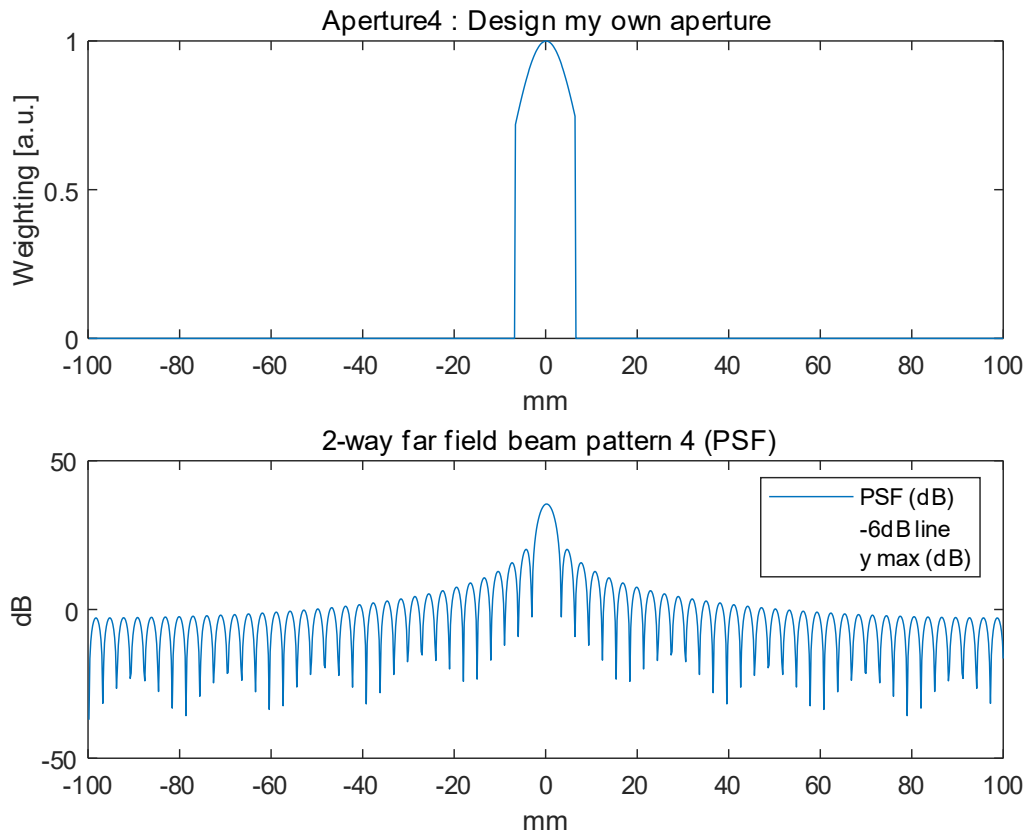
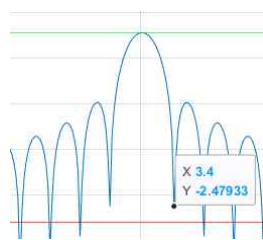
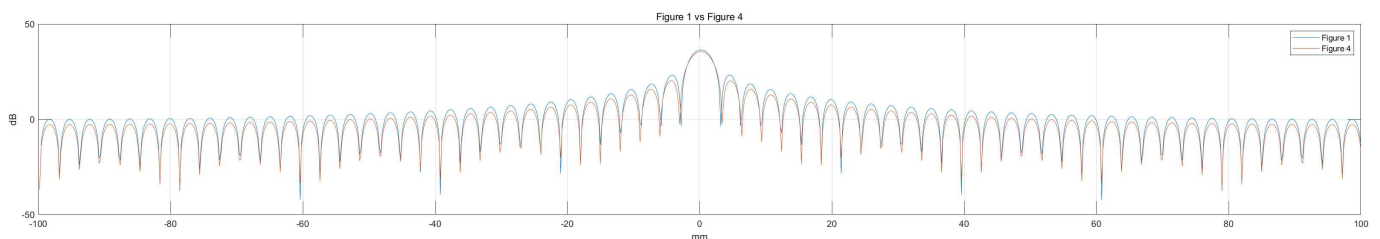


Figure 4. Design my own aperture

=> aperture를 설계하기 위해 여러 parameter들을 바꾸어 보았는데, 이상적인 aperture를 위한 목표 2가지를 생각하였습니다. 첫 번째는 'main lobe의 peak가 높으면서 width가 좁아야한다'이고 두 번째는 'sidelobe의 dB가 낮아야한다'입니다. 첫 번째 목표를 위해서 벡터의 사이즈를 조정해보았는데, 높은 peak에서 그래프의 모양이 왜곡되는 문제점이 있었습니다. 따라서 두 번째 목표인 sidelobe의 dB를 낮추기 위한 aperture의 weighting을 조절하였습니다. 결론적으로 [Fig. 2]와 유사하지만 gaussian window의 alpha값을 조절하는 방식을 사용하였습니다.



: -6 dB beam width of the main lobe



=> Design my own aperture를 적용시킨 결과 main lobe의 -6dB 빔 폭이 소폭 넓어지고(6.4mm -> 6.8mm) peak level of the first sidelobe는 소폭 감소(36.3909dB -> 35.5012dB)한 결과를 얻었습니다. 그리고 [Fig. 1]에 비해서 sidelobe의 dB이 감소한 결과를 얻을 수 있었습니다.

Part II : Reconstruct raw RF data into an image

Question 1.

```
54 - for a_line = 1:41
55 -     for element = 1:elemnum
56 -         %-----
57 -         %YOU NEED TO ADD CODE HERE TO COMPUTE THE DELAY VALUES!
58 -
59 -         temp = element * a_line * k / c / fs; % temp is the delay value. I have to write the code to calculate focal delay.
60 -
61 -         %-----
62 -         delay(a_line,element,k) = temp; %compute focal delay for this line, element, and depth
63 -     end
```

```
93 - for element = 1:elemnum
94 -     %-----
95 -     %YOU NEED TO ADD CODE HERE TO DO SUMMATION
96 -
97 -     Summ = 0; %this is where beamforming (summation part) is applied
98 -
99 -     %-----
100 -     final_image(sample-9,Aline,k) = Summ; %this is where beamforming (summation part) is applied
101 - end
```

=> 코드를 완성해야 하는 부분이 2개 있었는데, 구현하지 못했습니다.

Question 2.

Question 3.

Question 4.

=> 마찬가지로 Question 1을 해결하지 못하여 구현하지 못했습니다.