

※ 각자가 임의로 선택한 영상신호(또는 음성신호)를

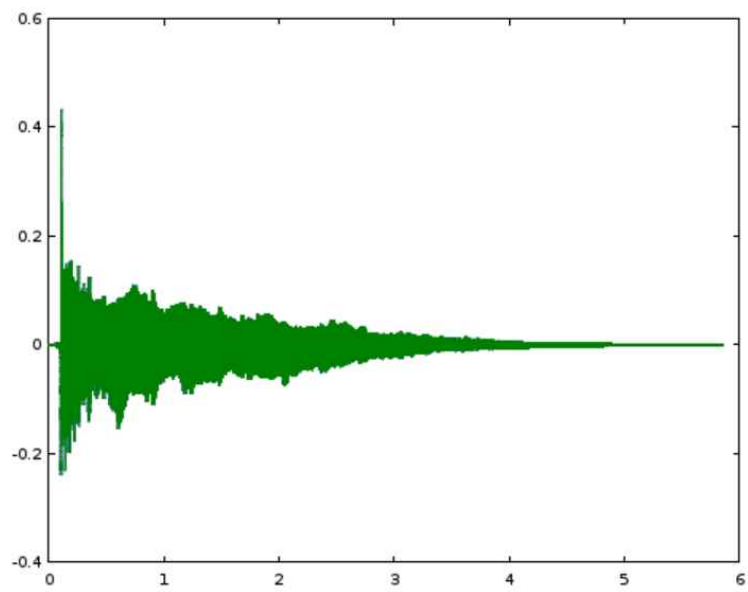
(1) 저주파 통과 필터링을 수행하라.

(2) 고주파 통과 필터링을 수행하라.

(1) 저주파 통과 필터링

| | |
|---------------------------------|-----------------------------|
| [x,fs]=audioread('guitar.wav'); | %fs=샘플링레이트, 오디오파일을 불러옴 |
| [N,a]=size(x); | %오디오파일의 크기를 구함N/fs = 플레이 타임 |
| t=0 : 1/fs : N/fs-1/fs ; | %0부터 1/샘플링단위로 플레이타임 만큼 표현 |
| plot(t, x) | %그래프 출력 |
| X=fftshift(fft(x)); | %오디오 파일 x을 고속 푸리에 변환함 |
| f=-N/2:N/2-1; | %f를 -N/2부터 1단위로 N/2-1 까지 표현 |
| figure | %여러개의 그래프 출력하기 위해 |
| plot(f, abs(X)) | %X의 절대값을 그래프로 출력 |
| axis([-20000 20000 0 600]) | %그래프의 폭 조정 |
| R=12000; | %필터의 크기를 결정 |
| H=zeros(1, N); | %오디오 파일의 크기만큼의 필터 배경을 만듦 |
| H(N/2-R:N/2+R)=1; | %필터의 크기만큼 중간부터 1을 넣음 |
| figure | |
| plot(H) | %필터 그래프 출력 |
| axis([0 250000 -0.2 1.2]) | |
| Y = X .* H ; | %X에 필터를 적용 후 Y에 넣음 |
| figure | |
| plot(t, Y) | %Y를 그래프로 출력 |
| x2=ifft(fftshift(Y)); | %fiit를 통해 복원 |
| figure | |
| plot(t, x2) | %그래프 출력 |
| 원본 | |

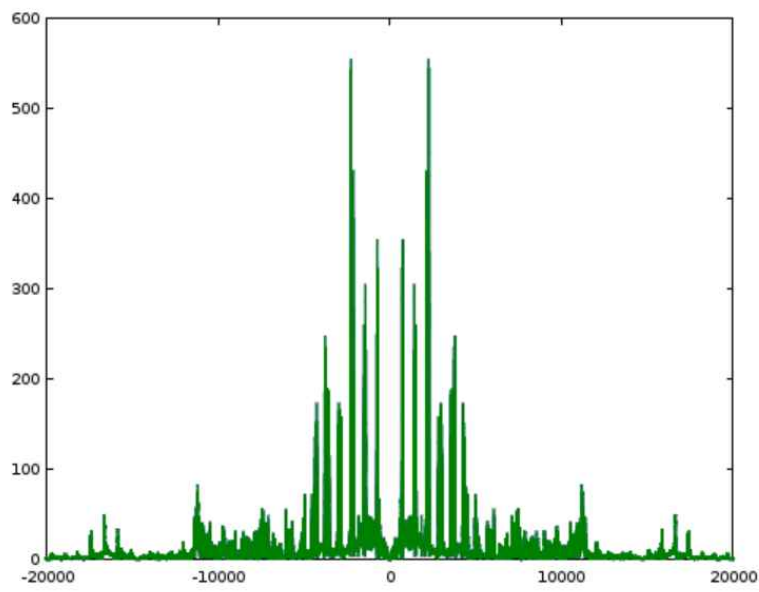
File Edit



A G P R ? [3.583, 0.06392]

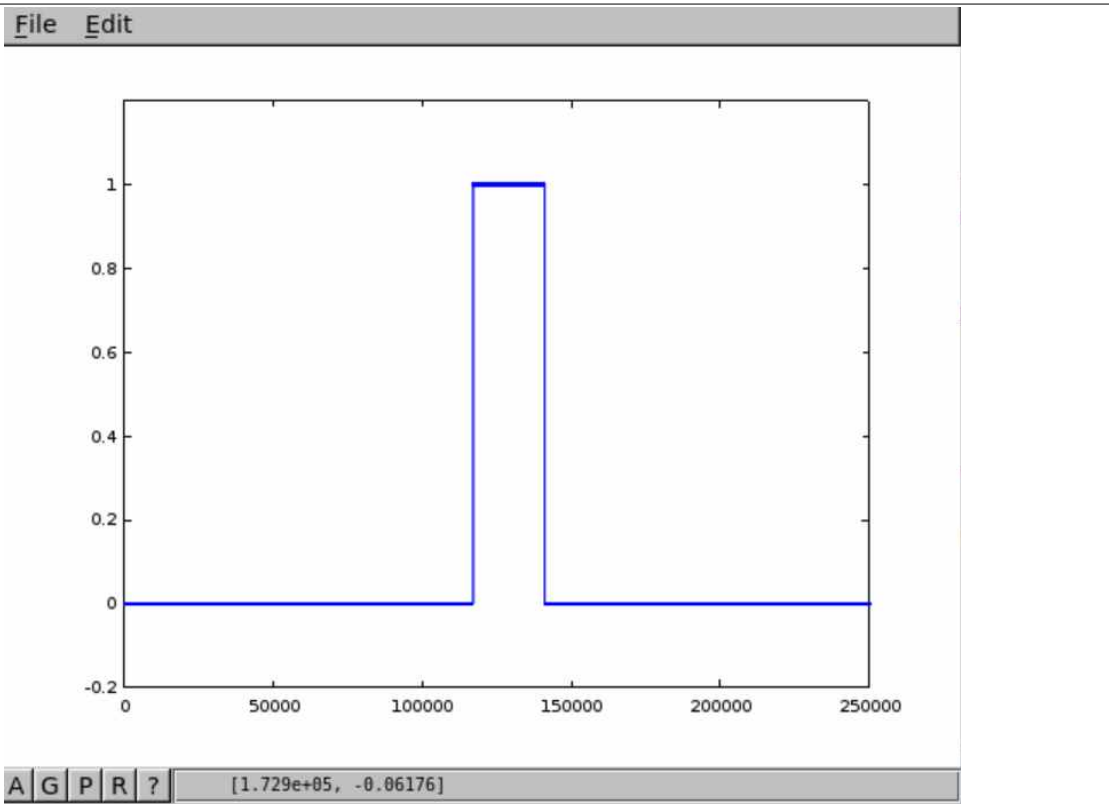
fft시킨 후

File Edit

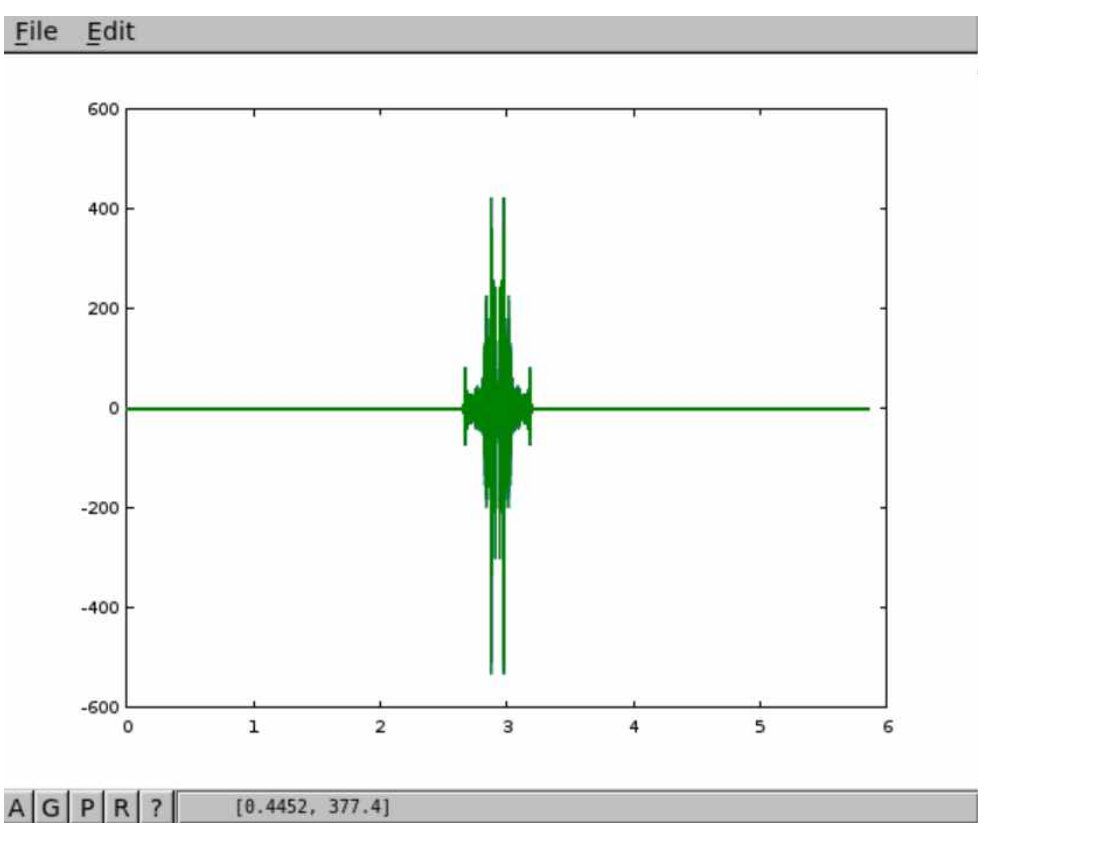


A G P R ? [5825, 437.9]

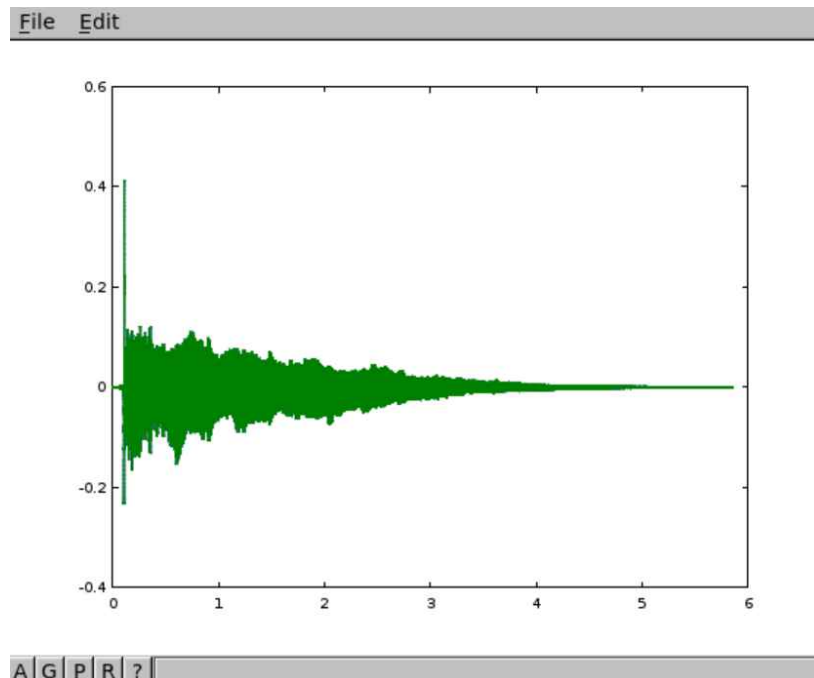
저주파 필터



fft한 결과를 필터링 한 후



ifft로 복원한 모습



설명

audioread()함수를 통해서 Xftp4를 이용하여 서버에 올린 guitar.wav 파일을 불러왔습니다. size()함수를 통해 N에 오디오파일의 크기를 구했습니다. N/fs는 플레이 타임이고 t가 0부터 시작했기 때문에 N/fs-1/fs까지 t를 설정했습니다. fftshift는 0을 가운데에 두어서 좌우 방향 데이터를 분석할 때 용이하고 fft를 사용해서 고속 푸리에 변환을 했습니다. R은 필터의 크기를 정하는데 $H(N/2-R:N/2+R)=0$ 에 있듯이 파일의 크기의 중앙에서 $\pm R$ 만큼의 필터를 만듭니다. 푸리에 변환시킨 X를 필터를 통과시키고 ifft로 복원시키면 필터링 된 오디오 파일을 구할 수 있습니다. octave에서 audiowrite()를 통해서 Xftp4에 오디오 파일을 뽑았습니다.

(2) 고주파 통과 필터링

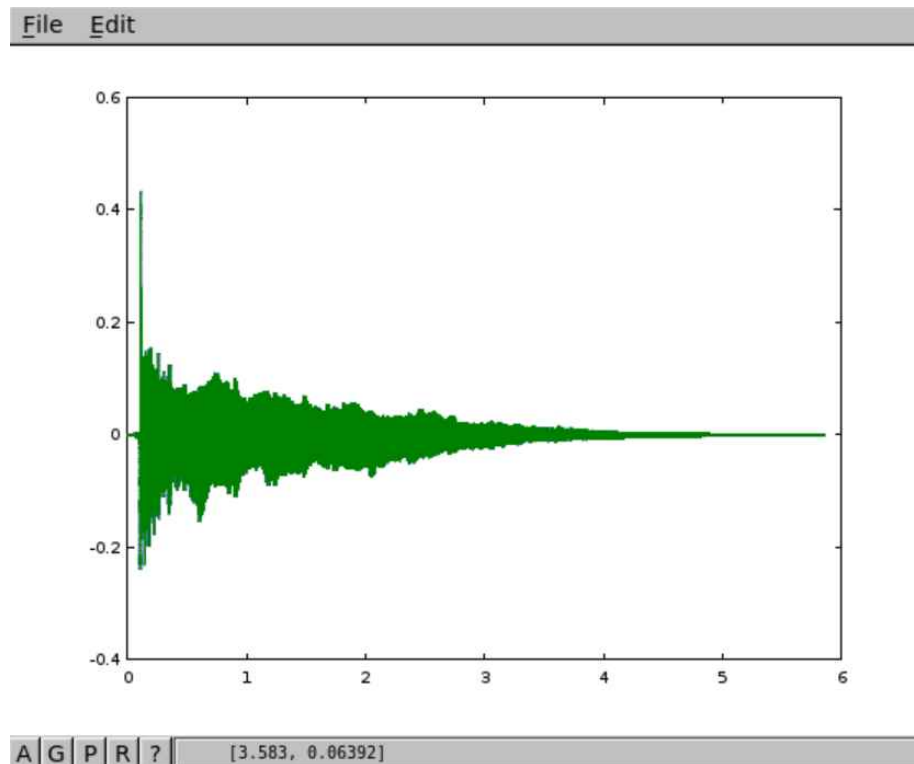
| | |
|---------------------------------|-----------------------------|
| [x,fs]=audioread('guitar.wav'); | %fs=샘플링레이트, 오디오파일을 불러옴 |
| [N,a]=size(x); | %오디오파일의 크기를 구함N/fs = 플레이 타임 |
| t=0 : 1/fs : N/fs-1/fs ; | %0부터 1/샘플링단위로 플레이타임 만큼 표현 |
| plot(t, x) | %그래프 출력 |
| X=fftshift(fft(x)); | %오디오 파일 x을 고속 푸리에 변환함 |
| f=-N/2:N/2-1; | %f를 -N/2부터 1단위로 N/2-1 까지 표현 |
| figure | %여러개의 그래프 출력하기 위해 |
| plot(f, abs(X)) | %X의 절대값을 그래프로 출력 |
| axis([-20000 20000 0 600]) | %그래프의 폭 조정 |
| R=12000; | %필터의 크기를 결정 |
| H=ones(1, N); | %오디오 파일의 크기만큼의 필터 배경을 만들 |
| H(N/2-R:N/2+R)=0; | %필터의 크기만큼 중간부터 0을 넣음 |

```

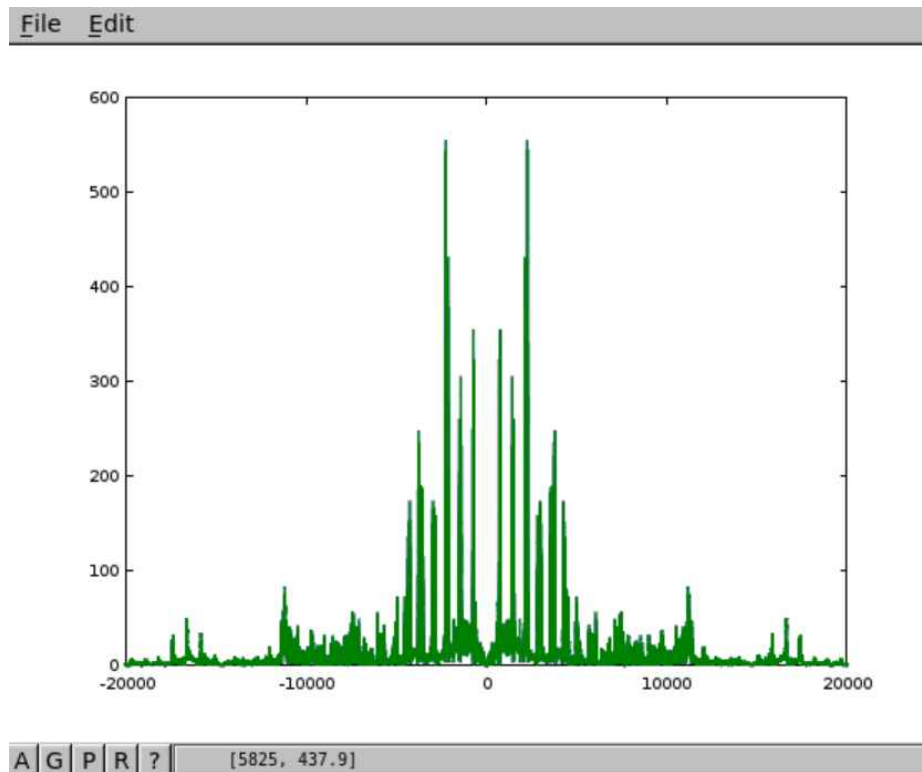
figure
plot(H)                                %필터 그래프 출력
axis([0 250000 -0.2 1.2])
Y = X .* H';                           %X에 필터를 적용 후 Y에 넣음
figure
plot(t, Y)                             %Y를 그래프로 출력
x2=ifft(fftshift(Y));                  %fiit를 통해 복원
figure
plot(t, x2)                            %그래프 출력

```

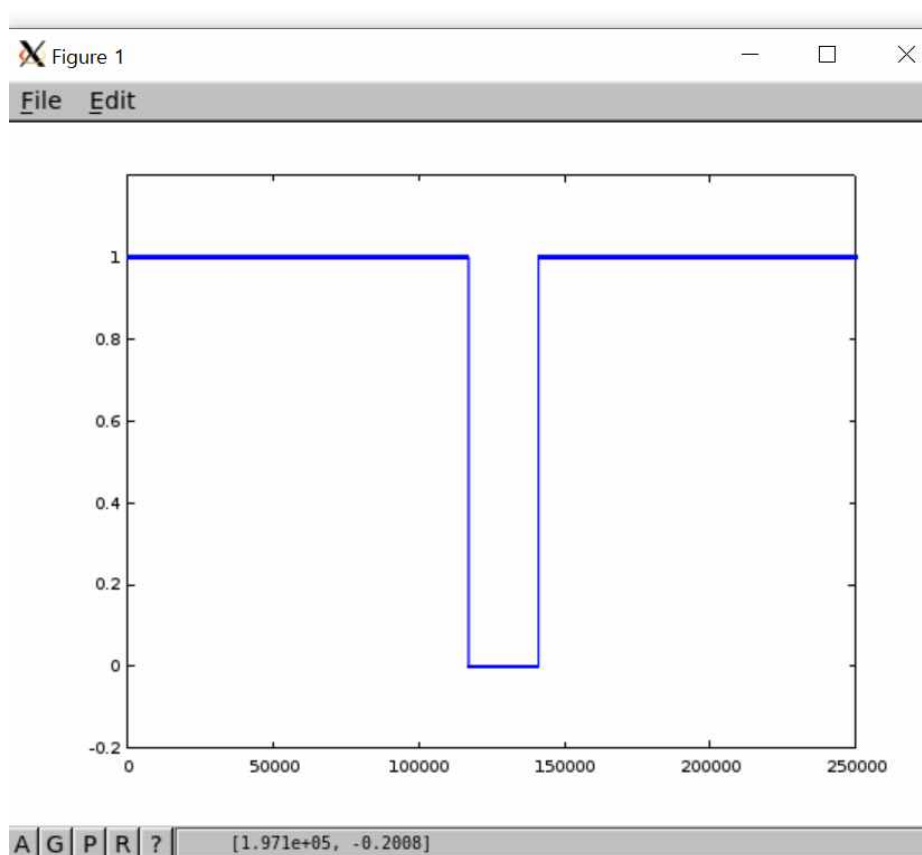
원본



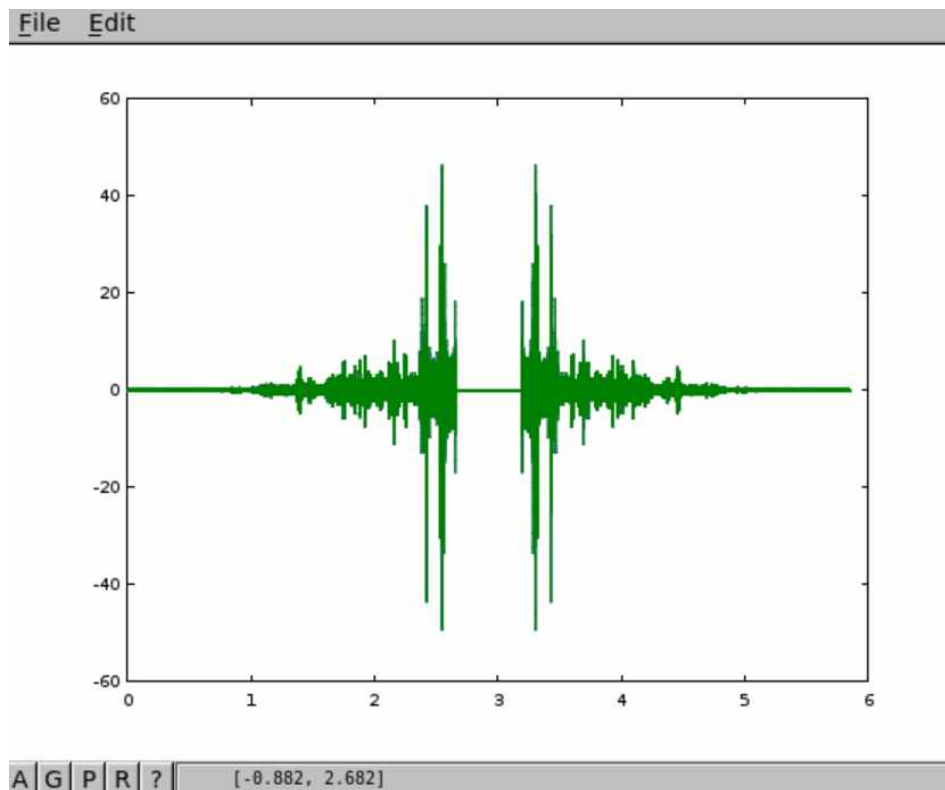
fft 시킨 후



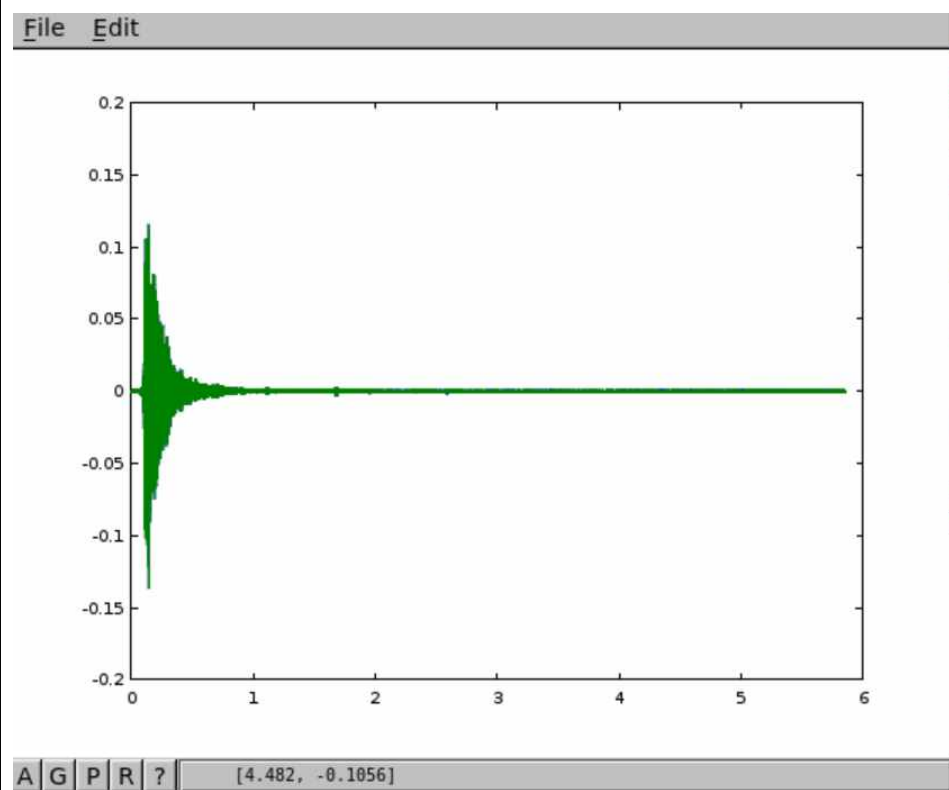
고주파 필터



fft시킨 결과를 필터링한 모습



ifft로 복원한 모습



설명

audioread()함수를 통해서 Xftp4를 이용하여 서버에 올린 guitar.wav 파일을 불러왔습니다. size()함수를 통해 N에 오디오파일의 크기를 구했습니다. N/fs는 플레이 타임이고 t가 0부터 시작했기 때문에 N/fs-1/fs까지 t를 설정했습니다. fftshift는 0을 가운데에 두어서 좌우 방향 데이터를 분석할 때 용이하고 fft를 사용해서 고속 푸리에 변환을 했습니다. R은 필터의 크기를 정하는데 $H(N/2-R:N/2+R)=1$ 에 있듯이 파일의 크기의 중앙에서 $\pm R$ 만큼의 필터를 만듭니다. 푸리에 변환시킨 X를 필터를 통과시키고 ifft로 복원시키면 필터링 된 오디오 파일을 구할 수 있습니다. octave에서 audiowrite()를 통해서 Xftp4에 오디오 파일을 뽑았습니다.