

Signal and Systems Project 1

함 태완¹

- 목차 -

1. Problem 1

- 1.1. sinusoidal signal 들의 합에 대한 분석
- 1.2. low-pass filter를 이용한 convolution

2. Problem 2

- 2.1. audio file의 time 및 angular frequency domain에서의 plot
- 2.2. low-pass filter를 이용한 convolution

3. Source Code

- 3.1. Problem 1 (*project1_1.m*)
 - 3.2. Problem 2 (*project1_2.m*)
-

* 해당 프로젝트에서는 MATLAB R2022a가 사용되었습니다.

1. Problem 1

1.1. sinusoidal signal 들의 합에 대한 분석

주어진 3가지 sinusoidal signal들의 합을 $x[n]$ 이라고 한다면 아래와 같이 표현할 수 있다.

$$x[n] = x_1[n] + x_2[n] + x_3[n]$$

$$x[n] = \cos(0.2\pi n) + \cos(0.4\pi n) + \cos(0.6\pi n)$$

이를 MATLAB에서 구현하여 Time domain에서 plot을 한다면 Fig. 1과 같은 결과를 얻을 수 있다. MATLAB에 내장되어 있는 `fft()` 함수와 `fftshift()` 함수를 이용하여 Angular Frequency domain에서 plot을 한다면 Fig. 2와 같은 결과를 얻을 수 있다. Fig. 2 에서는 plot을 통해 LTI system의 property를 확인할 수 있다.

¹ 연세대학교 전기전자공학부 학부생

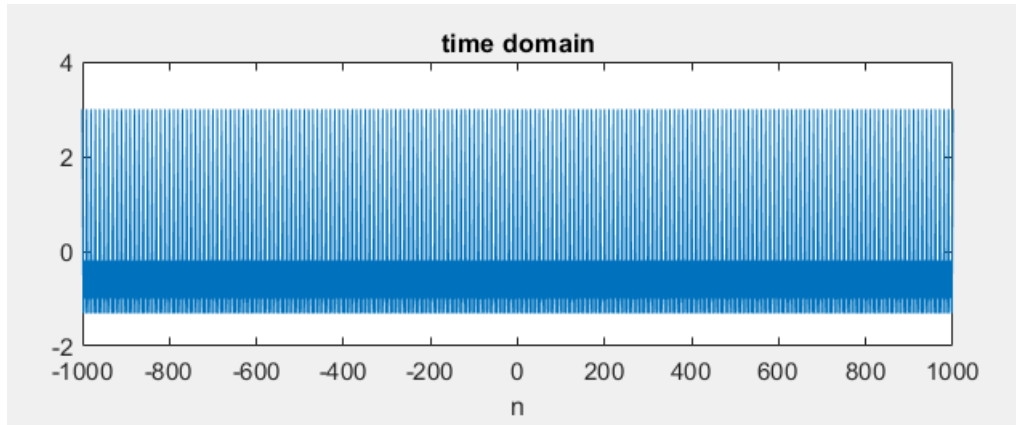


Fig. 1 Time domain에서의 합

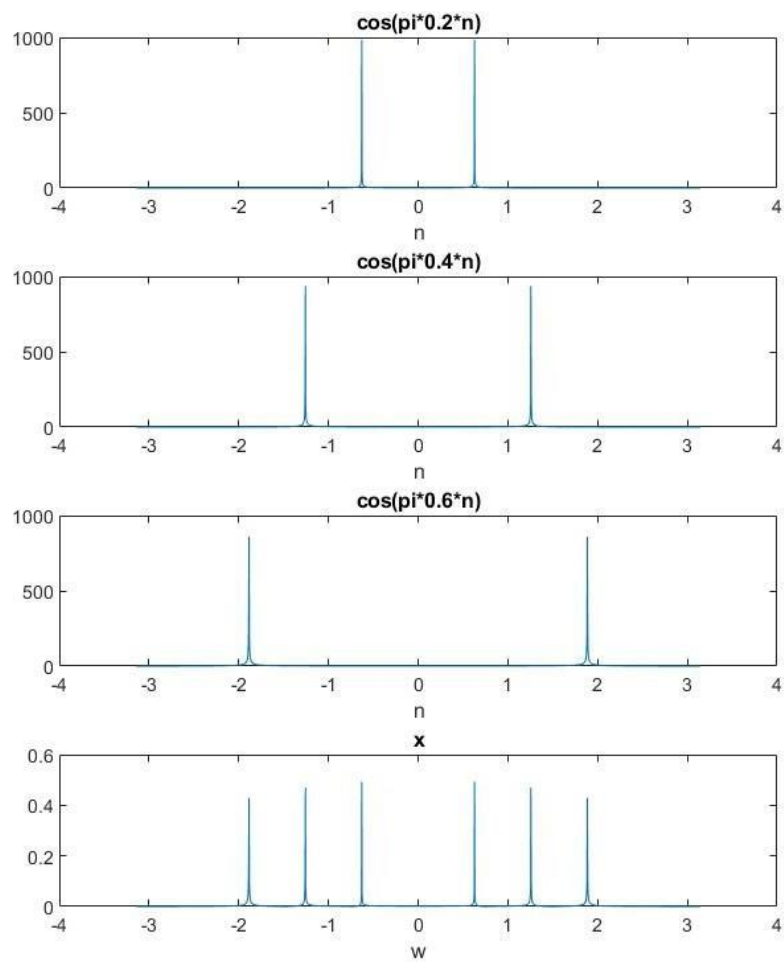


Fig. 2 Angular Frequency domain에서의 합

1.2. low-pass filter를 이용한 convolution

주어진 low-pass filter는 Fig. 3과 같다. 해당 filter는 $-1 \sim 1$ 의 frequency에서 pass를 하기 때문에 convolution 후에는 Fig. 2에서 확인할 수 있는 frequency domain에 있는 6개의 spike 중에서 $-1 \sim 1$ 내에 있는 2개의 spike만 남을 것이라는 추측을 할 수 있다.

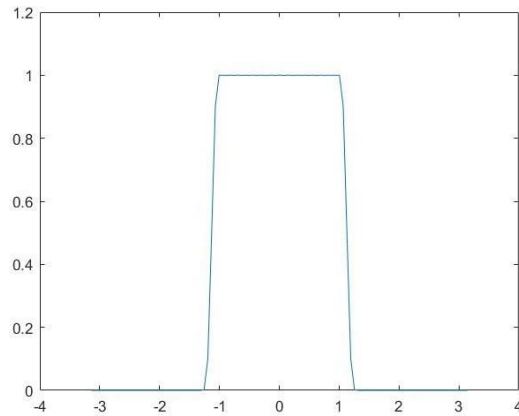


Fig. 3 low-pass filter in frequency domain

실제로 signal x 를 low-pass filter와 convolution을 해보면, Fig. 4와 같은 결과를 얻을 수 있으며 이는 추측과 일치함을 확인할 수 있다.

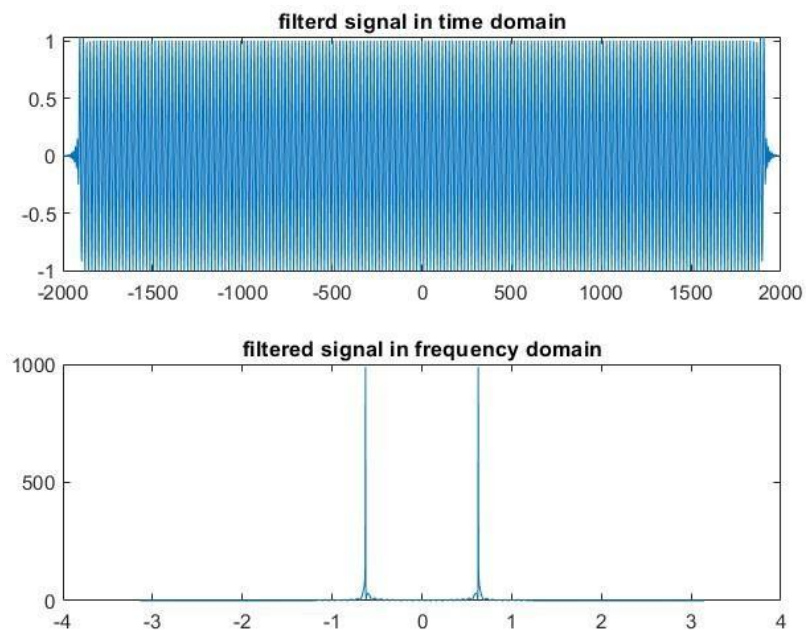


Fig. 4 filtered signal in time and frequency domain

2. Problem 2

2.1. audio file의 time 및 angular frequency domain에서의 plot

주어진 audio file을 time 및 angular frequency domain에서 plot하면 Fig. 5와 같은 결과를 얻을 수 있다.

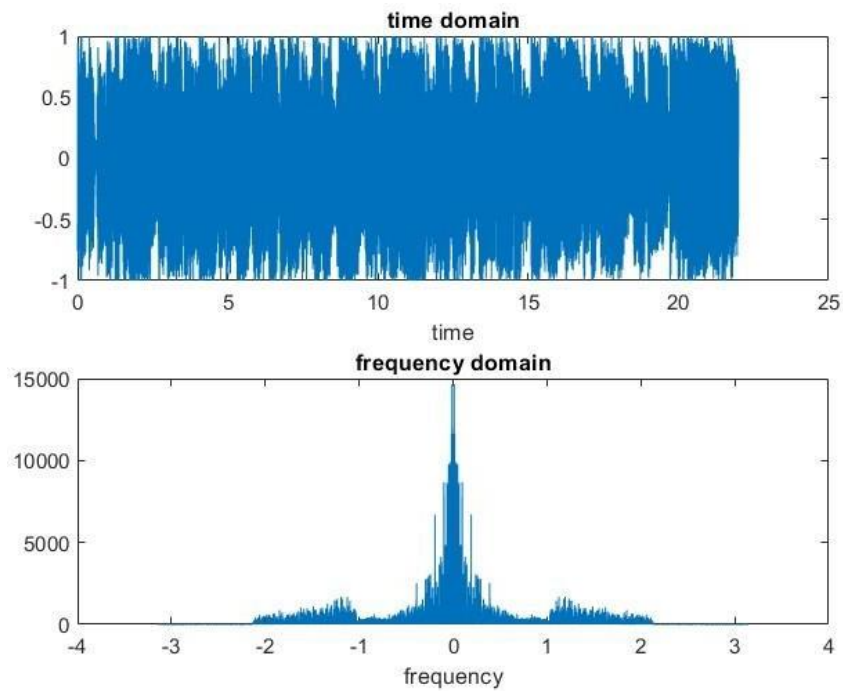


Fig. 5 Plot of audio file in time and frequency domain

2.2. low-pass filter를 이용한 convolution

주어진 low-pass filter는 Fig. 6와 같다. 해당 filter는 $-0.1 \sim 0.1$ 의 frequency만 pass하기 때문에 Fig. 5에서 0 부근의 peak 부분만 convolution 후 남고 나머지는 사라질 것이라는 예측을 할 수 있다.

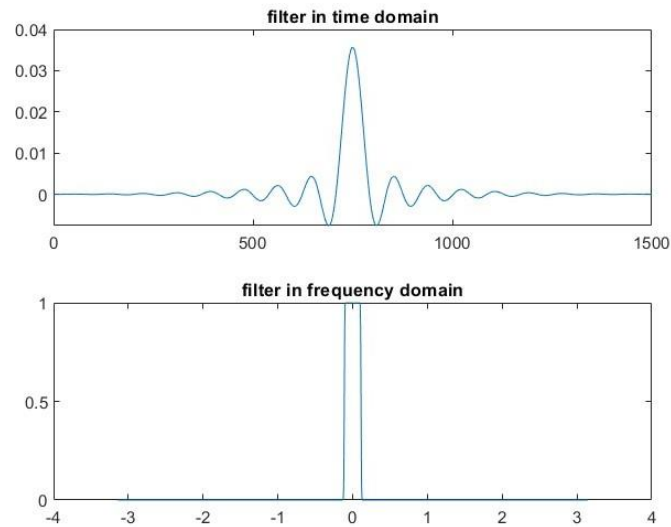


Fig. 6 low-pass filter

실제로, audio signal을 low-pass filter와 convolution을 한다면 Fig. 7과 같은 결과를 얻을 수 있다. 예측한대로, $-0.1 \sim 0.1$ 내의 frequency만 남고 나머지 frequency는 사라졌다. filter를 적용한 audio signal을 출력하여 원본 signal과 비교하면 상대적으로 낮은 frequency만 남은 것을 확인할 수 있다. 원본 오디오 파일은 높은 음들이 있어 상큼한 느낌을 주는 반면에, filter를 거친 오디오 파일은 높은 음들이 제거되어 둔탁한 느낌을 준다.

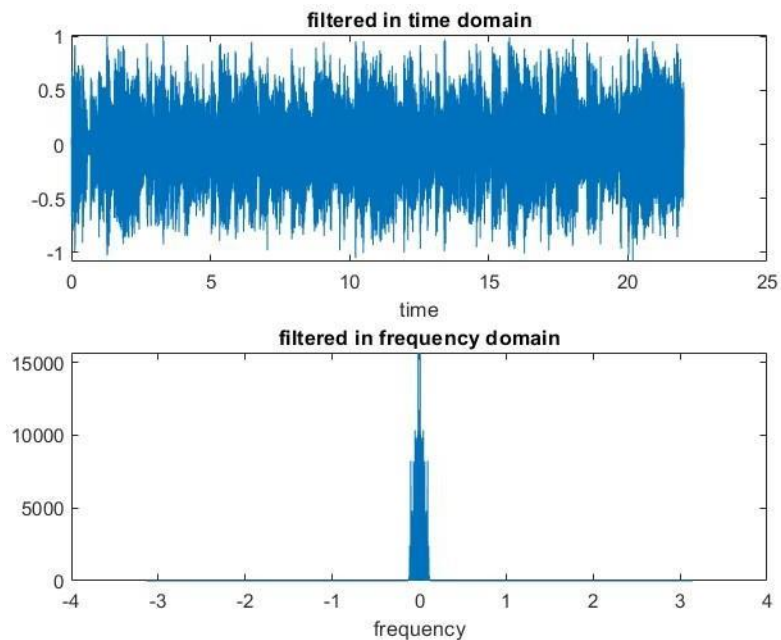


Fig. 7 filtered audio signal

3. Source Code

3.1. Problem 1 (*project1_1.m*)

```
n = -1000 : 1000; % n = -1000 ~ 1000, 2001 points
l = length(n);
w = linspace(-pi, pi, l); % w = -pi ~ pi

% Generating signal in each domain
x_t = cos(pi*0.2*n) + cos(pi*0.4*n) + cos(pi*0.6*n); % Added final signal in time domain
x_f = fftshift(fft(x_t)); % Added final signal in angular frequency domain

figure(1);
subplot(4,1,1); plot(w,abs(fftshift(fft(cos(pi*0.2*n))))); xlabel('n');
title('cos(pi*0.2*n)');
subplot(4,1,2); plot(w,abs(fftshift(fft(cos(pi*0.4*n))))); xlabel('n');
title('cos(pi*0.4*n)');
subplot(4,1,3); plot(w,abs(fftshift(fft(cos(pi*0.6*n))))); xlabel('n');
title('cos(pi*0.6*n)');
subplot(4,1,4); plot(w,abs(x_f)/l); xlabel('w'); title('x');

% (1-a) Plotting signal in each domain
figure(2);
subplot(2,1,1); plot(n,x_t); xlabel('n'); title('time domain') % plotting signal in time domain
subplot(2,1,2); plot(w,abs(x_f)/l); xlabel('w'); ylabel('magnitude'); title('angular frequency domain'); % plotting signal in frequency domain

% (1-b) Convolution with FIR filter
h_lpf1 = firceqrip(100,1/pi,[0.00057565 1e-4],'passedge'); % FIR lowpass filter with pass angular frequency w = -1 ~ 1
fil_h = fftshift(fft(h_lpf1));
figure(4); plot(linspace(-pi, pi, length(fil_h)), abs(fil_h)); % plotting filter in frequency domain

y = conv_filter(x_t, h_lpf1); % filtering by using convolution

% plotting filtered signal in each domain
figure(5);
subplot(2,1,1); plot(linspace(-2000,2000,length(y)), y); title('filtered signal in time domain')
subplot(2,1,2); plot(linspace(-pi, pi, length(x_t)+length(h_lpf1)-1), abs(fftshift(fft(y)))); title('filtered signal in frequency domain');

% convolution function
function [y] = conv_filter(x, h)
    m = length(x);
    n = length(h);
    X = [x, zeros(1,n-1)];
    H = [h, zeros(1,m-1)];
    y = zeros(1, m+n-1);

    % using the definition of linear convolution
    for i = 1:m+n-1
        for j = 1:i
            y(i) = y(i) + X(j)*H(i-j+1);
        end
    end
end
```

3.2. Problem 2 (*project1_2.m*)

```
[music, fs] = audioread('pj_music.wav'); % read audio file
t = (1 : length(music)) / fs; % make time interval for plotting in time domain
w = linspace(-pi, pi, length(music));
music_f = fftshift(fft(music)); % music signal in frequency domain

% (2-a) Plotting music in each domain
figure(1);
subplot(2,1,1); plot(t, music); xlabel('time'); title('time domain'); % plotting in time domain
subplot(2,1,2); plot(w, abs(music_f)); xlabel('frequency'); title('frequency domain') % plotting in frequency domain

% (2-b) Convolution with Low-pass filter
h_lpf2 = firceqrip(1000,0.1/pi,[0.00057565 1e-4],'passedge'); % FIR Lowpass filter with pass angular frequency w = -0.1 ~ 0.1
h_f = fftshift(fft(h_lpf2)); % Low-pass filter in frequency domain

% plotting low-pass filter
figure(2);
subplot(2,1,1); plot(linspace(0,1500,length(h_lpf2)), h_lpf2); title('filter in time domain');
subplot(2,1,2); plot(linspace(-pi,pi,length(h_f)), abs(h_f)); title('filter in frequency domain');

fil_music_t = conv(music, h_lpf2); % filtered music in time domain with low-pass filter
fil_music_f = fftshift(fft(fil_music_t)); % filtered music in frequency domain with low-pass filter
t_f = (1 : length(fil_music_t)) / fs; % time domain
w_f = linspace(-pi, pi, length(fil_music_f)); % angular frequency domain

% plotting results of filtering in time and frequency domain
figure(3);
subplot(2,1,1); plot(t_f, fil_music_t); xlabel('time'); title('filtered in time domain');
subplot(2,1,2); plot(w_f, abs(fil_music_f)); xlabel('frequency'); title('filtered in frequency domain');

% write filtered music file to wav audio file
audiowrite('filtered_result.wav',fil_music_t,fs);
```