



# REPORT

## -결과 보고서-

-LAB#9 - Audio Equalizer

**제출일** : 2018/11/30

**과목** : 임베디드 신호처리 시스템

**교수님** : 김 수민 교수님

**학과** : 임베디드 시스템

**학번** : 2014146020, 2014146042

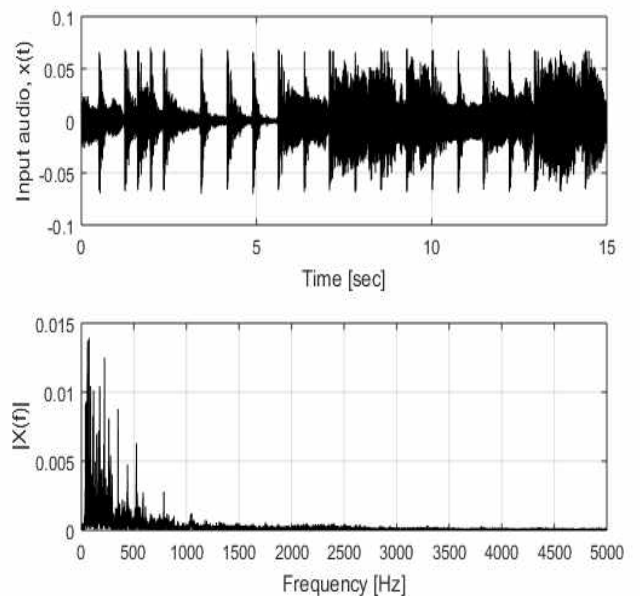
**이름** : 안 종석, 주 성 민

## 제 3 절 실습

### 3.1 Audio file 입력 및 재생

- (1) **실습** 제공되는 샘플 오디오 파일을 Matlab에서 읽어들이어 스펙트럼을 분석하라. (audioread, myfun\_SA 사용)
- (2) **실습 DEMO** 샘플 오디오 신호와 이 신호의 크기 스펙트럼을 그래프에 표시하라. (그림 3 참고)
- (3) **실습** 샘플 오디오 신호를 Matlab에서 재생하여 소리를 확인하라. (sound 사용)

```
1 - clc;clear ALL;clear;
2 - [x, fs]=audioread('Audio_Pop01_15sec.wav');
3 - n = 0:length(x)-1;
4 - t = n/fs;
5 - [f, X] = myfun_SA(t,x);
6 - figure(1)
7 - subplot(211)
8 - plot(t,x);grid on;xlabel('Time [sec]');ylabel('Input audio, x(t)');
9 - subplot(212)
10 - plot(f,abs(X));grid on;xlabel('Frequency [Hz]');ylabel('|X(f)|');
11 - axis([0,5000,0,0.015]);
12 - sound( x, fs )
```



audioread 함수를 사용하여 오디오 파일을 읽어들이어 스펙트럼을 분석하였습니다. 시간 축 신호  $x$ 와 샘플링 주파수  $fs$ 를 얻을 수 있습니다.  $n$ 은 시간축 신호  $x$ 의 샘플 갯수로 설정하고 변수  $t = n \times T_s$  로 시간축 벡터를 설정해주어 읽어들이고  $x$ 와 같이 그려주었습니다. 시간축 신호는 15초동안 소리의 크기에 따라 신호가 변하는 것을 볼 수 있었습니다. myfun\_SA 함수를 사용하여 주파수 축에서 크기 스펙트럼을 관찰하였습니다. 대부분의 신호가 1500Hz 이하에 있고, 모든 주파수 성분에 일정한 잡음이 관찰되는 것 같습니다.

## 3.2 디지털 audio equalizer 설계

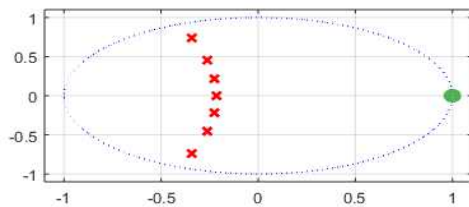
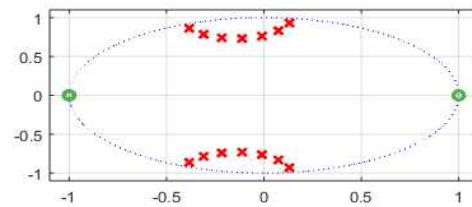
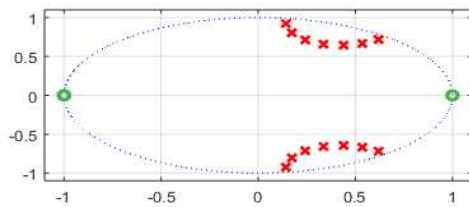
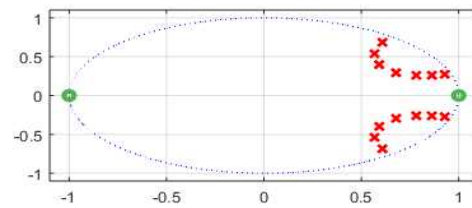
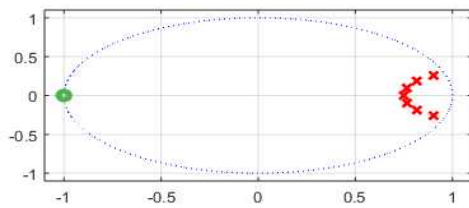
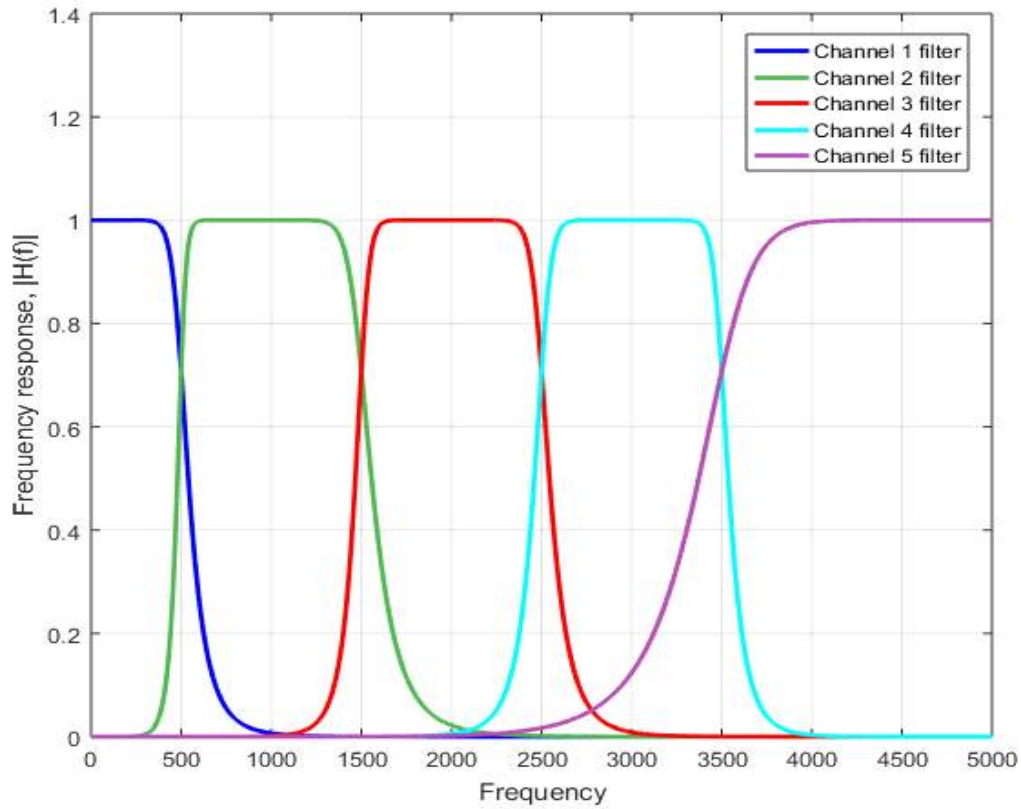
(1) **실습** 다음과 같은 사양의 IIR Butterworth 필터를 설계하라. (butter 사용)

- (Ch-1) 7차 LPF, Cutoff 주파수 500Hz
- (Ch-2) 12차 BPF, 통과대역 500Hz ~ 1500Hz
- (Ch-3) 12차 BPF, 통과대역 1500Hz ~ 2500Hz
- (Ch-4) 12차 BPF, 통과대역 2500Hz ~ 3500Hz
- (Ch-5) 10차 HPF, Cutoff 주파수 3500Hz

```
1 -   clc;
2 -   clear;
3 -   fs = 11025;
4 -   Omega = 0:0.001:pi;
5
6 -   W_Ch1 = (2*500)/fs;%Ch-1
7 -   W_Ch2 = [(2*500)/fs , (2*1500)/fs];%Ch-2
8 -   W_Ch3 = [(2*1500)/fs , (2*2500)/fs];%Ch-3
9 -   W_Ch4 = [(2*2500)/fs , (2*3500)/fs];%Ch-4
10 -  W_Ch5 = (2*3500)/fs;%Ch-5
11
12 -  [num_ch1, den_ch1] = butter(7, W_Ch1, 'low');
13 -  [num_ch2, den_ch2] = butter(6, W_Ch2, 'bandpass');
14 -  [num_ch3, den_ch3] = butter(6, W_Ch3, 'bandpass');
15 -  [num_ch4, den_ch4] = butter(6, W_Ch4, 'bandpass');
16 -  [num_ch5, den_ch5] = butter(10, W_Ch5, 'high');
```

fs는 1번에서 오디오 신호의 샘플링 주파수 값으로 결과를 보기 위해 임의로 설정해주었습니다. 만약 다른 오디오 샘플을 사용한다면 2번 문제에서도 똑같이 aaudioread 함수를 사용하면 됩니다. butter함수를 사용하기 전 각 채널에 맞게  $W_n$ 인자를 설정 해주었습니다.  $W_n = \frac{\Omega}{\pi} = \frac{2\pi\hat{f}}{\pi} = 2\hat{f} = 2\frac{f_0}{f_s}$  수식에 따라서 값을 설정해준 뒤 각 채널에 맞는 필터를 구현하였습니다.

- (2) 실습 DEMO 위에서 설계한 다섯 필터의 주파수 응답의 크기와 pole-zero plot을 그래프에 표시하라. (그림 4, 5 참고)



다섯 개의 채널에 해당하는 주파수 응답의 크기와 pole-zero plot 결과입니다. 주어진 성능을 만족시키면서 신호의 거의 모든 대역이 한 번 이상 필터에 들어가게 된다고 볼 수 있습니다. pole-zero plot을 확인해보면 모든 pole이 단위원 안에 있기 때문에 안정하다고 볼 수 있습니다.

이전부터 pole-zero plot이 가지고있는 여러 가지 정보를 배우면서, 신호처리에 있어서는 빠지면 안될 정말 중요한 요소라는 생각이 들었습니다. 주파수 응답의 크기를 근사화할 수 있고, 안정성을 판별할 수도 있고, 심지어 극점이 0점에 없다면 IIR 필터(시스템)이라는 정보까지, 단순하지만 엄청나게 함축적인 pole-zero plot은 어렵다고 느꼈던 주파수 차원을 매력적으로 느끼게 해주었습니다.