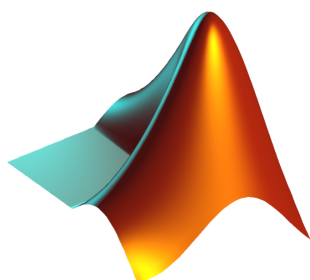


임베디드 신호처리 실습

APF 실습 결과보고서

전자공학부 임베디드시스템

2014146004 김민섭



MATLAB

3.0 MATLAB 명령어 (내부함수)

- $[z, p, k] = \text{buttap}(n)$
: 아날로그 Butterworth 프로토타입 필터의 pole, zero를 구한다.
- $[z, p, k] = \text{cheb1ap}(n, R_p)$
: 아날로그 Chebyshev Type-I 프로토타입 필터의 pole, zero를 구한다.
- $[z, p, k] = \text{cheb2ap}(n, R_p)$
: 아날로그 Chebyshev Type-II 프로토타입 필터의 pole, zero를 구한다.
- $[z, p, k] = \text{ellipap}(n, R_p, R_s)$
: 아날로그 Elliptic 프로토타입 필터의 pole, zero를 구한다.
- $[\text{numt}, \text{dent}] = \text{lp2lp}(\text{num}, \text{den}, w_0)$
: cutoff 주파수가 1[rad/sec]인 LPF를 cutoff 주파수가 w_0 [rad/sec]인 LPF로 변환.
- $[\text{numt}, \text{dent}] = \text{lp2hp}(\text{num}, \text{den}, w_0)$
: cutoff 주파수가 1[rad/sec]인 LPF를 cutoff 주파수가 w_0 [rad/sec]인 HPF로 변환.
- $[\text{numt}, \text{dent}] = \text{lp2bp}(\text{num}, \text{den}, w_0), B_w$
: cutoff 주파수가 1[rad/sec]인 LPF를 중심 주파수가 w_0 [rad/sec]이고 대역폭이 B_w [rad/sec]인 BPF로 변환.
- $[\text{numt}, \text{dent}] = \text{lp2bs}(\text{num}, \text{den}, w_0), B_w$
: cutoff 주파수가 1[rad/sec]인 LPF를 중심 주파수가 w_0 [rad/sec]이고 대역폭이 B_w [rad/sec]인 대역을 저지하는 BPS로 변환.
- $H = \text{freqs}(\text{num}, \text{den}, w)$
: 아날로그 시스템의 전달함수 $H(s)$ 로부터 주파수 응답을 구한다.

3.1 아날로그 프로토타입 필터

Butterworth 필터

- (1) 5차 Butterworth LPF의 전달함수를 구하고 pole-zero plot을 반지름 1인 원과 함께 그래프에 표시하라. (butter 함수 이용)
- (2) 위에서 구한 필터의 주파수 응답의 크기를 그래프에 표시하고 실제 목표에 맞는지 확인하라.
(주파수 응답을 전력에 대해 dB-scale로 표시하라. 즉 $10 \cdot \log_{10}|H(w)|^2$ 으로 환산하여 표시 할 것)

Chebyshev Type-I/II

- (1) 다음과 같은 파라미터를 갖는 Chebyshev Type-I/II의 전달함수를 구하고 pole-zero plot을 반지름 1인 원과 함께 그래프에 표시하라. (cheb1ap, cheb2ap 함수 이용)
 - 차수 : 5차
 - $R_p = 10[\text{dB}]$, $R_s = 30[\text{dB}]$
- (2) 위에서 구한 필터의 주파수 응답의 크기를 그래프에 표시하고 실제 목표에 맞는지 확인하라.
(주파수 응답을 전력에 대해 dB-scale로 표시하라. 즉 $10 \cdot \log_{10}|H(w)|^2$ 으로 환산하여 표시 할 것)

Elliptic 필터

- (1) 다음과 같은 파라미터를 갖는 Elliptic LPF의 전달함수를 구하고 pole-zero plot을 반지름 1인 원과 함께 그래프에 표시하라. (butter 함수 이용)
 - 차수 : 5차
 - $R_p = 10[\text{dB}]$, $R_s = 30[\text{dB}]$
- (2) 위에서 구한 필터의 주파수 응답의 크기를 그래프에 표시하고 실제 목표에 맞는지 확인하라.
(주파수 응답을 전력에 대해 dB-scale로 표시하라. 즉 $10 \cdot \log_{10}|H(w)|^2$ 으로 환산하여 표시 할 것)

=> 실습(DEMO) 3.1에서 보인다.

(실습 3.1) 위 실습의 결과를 그래프에 함께 표시하고 비교하라.

-① 실습 3.1 코딩

[illegible]

```

46
47 — subplot(2,2,1);
48 — plot(real(buttap_z),imag(buttap_z),'o','MarkerSize',20,'LineWidth',4)
49 — hold on;
50 — plot(x,'k','MarkerSize',20,'LineWidth',4)
51 — plot(real(buttap_p),imag(buttap_p),'xr','MarkerSize',20,'LineWidth',4)
52 — hold off;
53 — xlabel('Real(s)');
54 — ylabel('Image(s),Butterworth');
55 — grid on;
56
57 — subplot(2,2,2);
58 — plot(real(cheb1ap_z),imag(cheb1ap_z),'o','MarkerSize',20,'LineWidth',4)
59 — hold on;
60 — plot(x,'k','MarkerSize',20,'LineWidth',4)
61 — plot(real(cheb1ap_p),imag(cheb1ap_p),'xr','MarkerSize',20,'LineWidth',4)
62 — hold off;
63 — xlabel('Real(s)');
64 — ylabel('Image(s),Chebyshev_I');
65 — grid on;
66
67 — subplot(2,2,3);
68 — plot(real(cheb2ap_z),imag(cheb2ap_z),'o','MarkerSize',20,'LineWidth',4)
69 — hold on;
70 — plot(x,'k','MarkerSize',20,'LineWidth',4)
71 — plot(real(cheb2ap_p),imag(cheb2ap_p),'xr','MarkerSize',20,'LineWidth',4)
72 — hold off;
73 — xlabel('Real(s)');
74 — ylabel('Image(s),Chebyshev_II');
75 — grid on;
76
77 — subplot(2,2,4);
78 — plot(real(ellipap_z),imag(ellipap_z),'o','MarkerSize',20,'LineWidth',4)
79 — hold on;
80 — plot(x,'k','MarkerSize',20,'LineWidth',4)
81 — plot(real(ellipap_p),imag(ellipap_p),'xr','MarkerSize',20,'LineWidth',4)
82 — hold off;
83 — xlabel('Real(s)');
84 — ylabel('Image(s),Elliptic');
85 — grid on;
86
87 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
88 — figure(2);
89
90 — subplot(2,2,1);
91 — plot(f_w,buttap_freq_response,'MarkerSize',20,'LineWidth',4);
92 — axis([0 2 -40 10])
93 — ylabel('Butterworth');
94 — grid on;
95

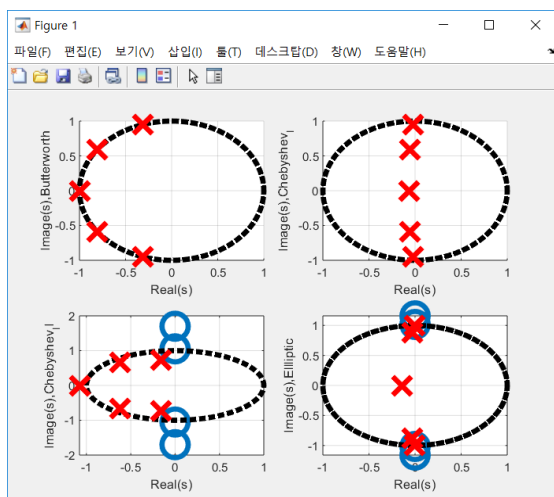
```

```

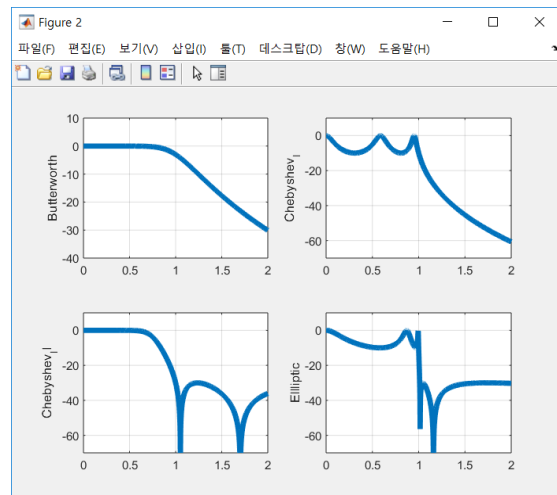
96 — subplot(2,2,2);
97 — plot(f_w,cheblap_freq_response,'MarkerSize',20,'LineWidth',4);
98 — axis([0 2 -70 10])
99 — ylabel('Chebyshev_I');
100 — grid on;
101
102 — subplot(2,2,3);
103 — plot(f_w,cheb2ap_freq_response,'MarkerSize',20,'LineWidth',4);
104 — axis([0 2 -70 10])
105 — ylabel('Chebyshev_II');
106 — grid on;
107
108 — subplot(2,2,4);
109 — plot(f_w,ellipap_freq_response,'MarkerSize',20,'LineWidth',4);
110 — axis([0 2 -70 10])
111 — ylabel('Elliptic');
112 — grid on;

```

② 실습 3.1 코딩의 결과_ 그래프



▲ (1)의 결과 그래프



▲ (2)의 결과 그래프

-③ 실습 3.1 설명

Figure (1)

subplot(221) : Butterworth 필터는 zero가 없고 pole이 원주상에 위치한다.

subplot(222) : Chevyshev Type-I는 zero가 없고 pole이 y축(허수축) 가까이에 위치한다.

subplot(223) : Chevyshev Type-II는 zero는 y축(허수축) 가까이에, pole은 원주 가까이에 위치한다.

subplot(224) : Elliptic 필터는 zero와 pole이 모두 y축 가까이에 위치하고, zero는 원주와 단위원 외부에, pole은 원주와 단위원 내부에 위치한다.

⇒ 시스템의 차수는 pole이 결정하며, 필터를 조정할 수 있는 파라미터가 pole의 개수, 즉 시스템의 차수이다.

Figure (2)

subplot(221) : Butterworth 필터는 통과대역과 저지대역이 평평하며 천이구간이 큰 특징을 가지고 있다. 차수가 커질수록 천이 대역이 작아지므로 성능이 향상된다.

subplot(222) : Chevyshev Type-I는 stopband 사이의 천이구간을 줄이기 위해 passband에 리플을 허용하였다. 리플을 허용하므로 천이구간을 줄이는 효과는 있지만 신호의 왜곡이 발생한다는 단점이 있다. 따라서 시스템의 차수가 커질수록 천이대역이 작아지므로 성능이 좋아지지만 리플이 많아지므로 신호의 왜곡을 감수해야한다.

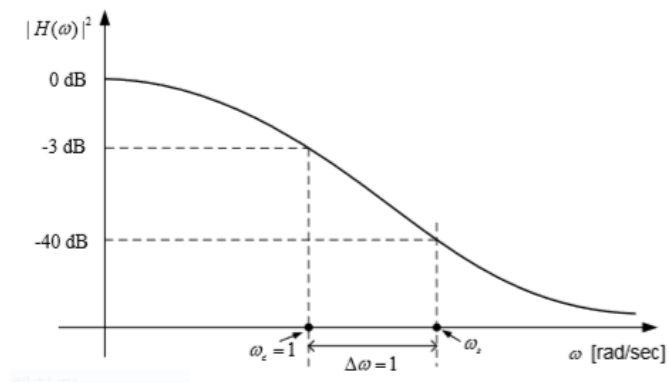
subplot(223) : Chevyshev Type-II는 passband 사이의 천이구간을 줄이기 위해 stopband에 리플을 허용하였다. 리플을 허용하므로 천이구간을 줄이는 효과는 있지만 신호의 왜곡이 발생한다는 단점이 있다. 따라서 시스템의 차수가 커질수록 천이대역이 작아지므로 성능이 좋아지지만 리플이 많아지므로 신호의 왜곡을 감수해야한다.

subplot(224) : Elliptic 필터는 stopband와 passband에 리플을 허용한 필터이다. 그렇기 때문에 Chevyshev Type-I/II와 같이 시스템의 차수가 높아질수록 천이대역이 줄어들지만 신호의 왜곡이 발생하며, stopband와 passband 전체에 리플을 허용하므로 Chevyshev Type-I/II보다 더 많이 왜곡된다.

⇒ 따라서 시스템의 차수가 높아질수록 Butterworth 필터의 성능이 가장 좋아진다는 특징을 가진다.

3.2 Butterworth 필터

(실습 3.2.1) 그림과 같은 사양의 LPF를 Butterworth 필터로 구현하고자 할 때 필요한 차수는 얼마인가?



(실습 3.2.2) 위에서 구한 차수의 Butterworth 필터 전달함수를 구하고 pole-zero plot을 반지름 1인 원과 함께 그래프에 표시하라. (buttap 함수 이용)

=>DEMO에서 보인다.

(실습 3.2.3) 실습 3.2.2에서 구한 필터의 주파수 응답의 크기를 그래프에 표시하고 설계 목표에 맞는지 확인하라.

(주파수 응답을 전력에 대해 dB-scale로 표시하라. 즉 $10 \cdot \log_{10}|H(w)|^2$ 으로 환산하여 표시 할 것)

=>DEMO에서 보인다.

(실습 3.2.4) 실습 3.2.3에서 설계한 LPF를 다음과 같은 다섯 종류의 필터로 변환하고 각 필터의 주파수 응답을 dB-scale로 그래프에 표시하라. (lp2bp, lp2hp, freqs 함수 사용)

1. LPF, cutoff 주파수 = 300 Hz
2. BPF, passband = [300, 1000] Hz
3. BPF, passband = [1000, 2000] Hz
4. BPF, passband = [2000, 4000] Hz
5. HPF, cutoff 주파수 = 400 Hz

=>DEMO에서 보인다.

(DEMO) 위 실습의 결과를 그래프에 함께 나타내라.

-① DEMO 코딩

```
1 - clear;
2 - clc;
3
4 - se= linspace(-2*pi,2*pi,360);
5 - f_w= linspace(0,2.5,100);
6 - f_w_High= linspace(0,5000,10000);
7
8 % 단위원
9 - x=exp(1j*se);
10
11 %%% 3_2_2 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
12
13 - N = ceil(log10((1/10^(-4))-1)/(2*log10(2)));
14 % pole, zero 구하기
15 - [buttap_z, buttap_p, buttap_k] = buttap(N);
16
17 %%% 3_2_3 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
18
19 % 주파수 응답의 크기를 그래프에 표시(dB-scale로 표시)
20 % pole과 zero로 부터 전달함수 구하기
21 - [buttap_zero_vector, buttap_pole_vector]=zp2tf(buttap_z,buttap_p,buttap_k);
```

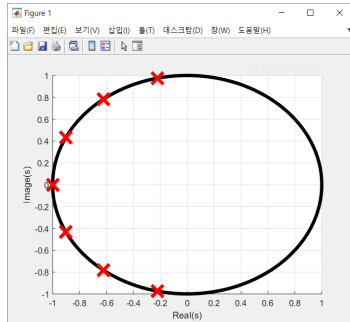
[illegible]

```

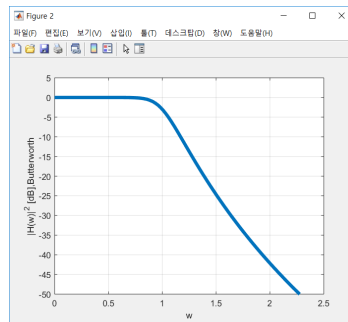
66 — figure(1);
67
68 — plot(real(buttap_z),imag(buttap_z),'o','MarkerSize',20,'LineWidth',4)
69 — hold on;
70 — plot(x,'k:','MarkerSize',20,'LineWidth',4)
71 — plot(real(buttap_p),imag(buttap_p),'xr','MarkerSize',20,'LineWidth',4)
72 — hold off;
73 — xlabel('Real(s)');
74 — ylabel('Image(s)');
75 — grid on;
76
77 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
78 — figure(2);
79
80 — plot(f_w,buttap_freq_response,'LineWidth',4);
81 — axis([0 2.5 -50 5])
82 — xlabel('w');
83 — ylabel('|H(w)|^2 [dB],Butterworth');
84 — grid on;
85
86 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
87 — figure(3);
88
89 — plot(f_w_High,lpf_freq_response300,'LineWidth',4);
90 — hold on;
91 — plot(f_w_High,bpf_freq_response300,'LineWidth',4);
92 — plot(f_w_High,bpf_freq_response1000,'LineWidth',4);
93 — plot(f_w_High,bpf_freq_response2000,'LineWidth',4);
94 — plot(f_w_High,hpf_freq_response4000,'LineWidth',4);
95 — hold off;
96 — ylim([-30 10])
97 — xlabel('Frequency [Hz]');
98 — ylabel('|H(w)|^2 [dB]');
99 — legend('Filter 1, LPF', 'Filter 2, BPF', 'Filter 3, BPF', 'Filter 4, BPF', 'Filter 5, HPF');
100 — grid on;

```

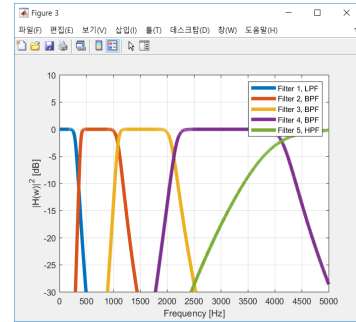
-② DEMO 코딩의 결과_ 그래프



▲ 실습 3.2.2의 결과



▲ 실습 3.2.3의 결과



▲ 실습 3.2.4의 결과

-③ DEMO 결과 해석

Figure (1) : 실습 3.2.1에서 구했듯이 $N=7$ 이므로 7차 Butterworth 필터이며, 시스템의 차수는 pole의 개수가 결정하므로 그래프에서 pole의 개수가 7개인 것을 확인 할 수 있다. 모든 pole이 반지름이 1인 좌측부분, 즉 시스템이 안정한 좌평면에 등간격으로 위치하는 것을 볼 수 있다. Butterworth 필터는 zero가 없고 pole만 있는 시스템이다.

Figure (2) : 이 그래프는 주어진 천이대역에 필요한 차수와 그에 따른 주파수 응답의 크기의 제곱을 dB-scale로 그린 것이다. $-40[\text{dB}]$ 를 만족하는 주파수가 stopband의 시작점이고 cutoff 주파수는 $1[\text{rad/sec}]$ 이다. DEMO에서는 시스템의 차수가 7차 이므로 천이구간 (Δw)은 1이다. 이 그래프는 통과대역과 저지대역이 평평한 Butterworth 필터의 특징을 보여주며 낮은 주파수에서 통과대역이 존재하므로 Low_Pass_Filter(LPF)이다. 따라서 7차 Butterworth LPF의 결과를 확인 할 수 있다.

Figure (3) : 아날로그 프로토 타입 필터들을 cutoff주파수가 $1[\text{rad/sec}]$ 인 LPF로 정의되어 있는데 이를 이용해 설계한 필터의 특성을 유지하면서 cutoff 주파수를 바꾸면 LPF를 BPF, HPF, BSF등으로 형태를 바꾸는 것을 주파수 변환 이라고 한다. 실습 3.2.4의 코딩을 살펴보면 lp2lp, lp2bp, lp2hp 함수를 사용 한 것을 볼 수 있다. 이 함수 중에서 lp2bp함수를 보면 예를들어 passband = $[300, 1000]$ Hz인 경우, 300과 1000의 중간지점, 1000에서 300을 뺀 천이구간을 계산해서 함수에 적용해야 한다. 그 결과 그래프를 보면, LPF가 원하는 형태에 따라 cutoff 주파수를 바꿔 LPF, HPF 그리고 BPF까지 확인 할 수 있다.

※주파수 변환

▶ LPF \rightarrow LPF : cutoff 주파수가 w_1 인 LPF $H(s)$ 를 cutoff 주파수가 w_2 인 LPF로 변환했을 때의 전달함수 $H_{new}(s)$ 는 다음과 같다.

▶ LPF \rightarrow HPF : cutoff주파수가 w_1 인 LPF $H(s)$ 를 주파수 w_2 이상의 모든 대역 신호를 통과시키는 HPF로 변환했을 때의 전달함수 $H_{new}(s)$ 는 다음과 같다.

▶ LPF \rightarrow BPF : cutoff주파수가 w_c 인 LPF $H(s)$ 를 주파수 $[w_1, w_2]$ 의 신호를 통과시키는 BPF로 변환했을 때의 전달함수 $H_{new}(s)$ 는 다음과 같다.

▶ LPF \rightarrow BSF : cutoff주파수가 w_c 인 LPF $H(s)$ 를 주파수 $[w_1, w_2]$ 의 신호를 저지(stop)시키는 BSF로 변환했을 때의 전달함수 $H_{new}(s)$ 는 다음과 같다.