임베디드 신호처리 실습
결과리포트
LAB-8

한국산업기술대학교
KOREA POLYTECHNIC UNIVERSITY

전자공학부 임베디드시스템
2016146048 한 대성
2014146012 박동훈
김수민 교수님

3.1 시스템 이산화

(1) 실습 다음은 cutoff 주파수가 1 rad/sec인 3차 아날로그 Butterworth LPF의 전달함수이다.

$$H(s) = \frac{1}{s^3 + 2s^2 + 2s + 1}$$

이 시스템에 bilinear transformation을 적용하여 디지털 필터로 변환하라. 이때 다음과 같은 표본화 주파수를 사용하라. (bilinear 함수 사용)

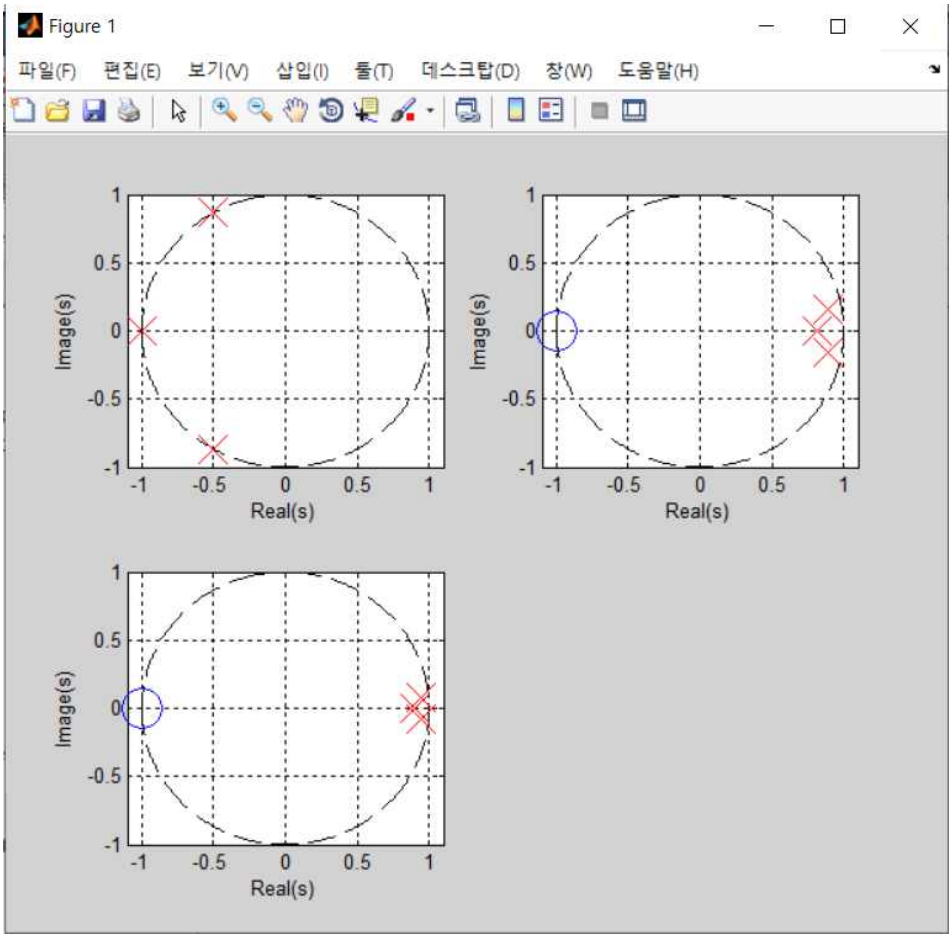
$$f_s \in \{5, 10\} \text{ Hz}$$

```
n = linspace(-pi, pi, 360);
w = linspace(0, 2, 100);
t = linspace(0,49,50);
t2 = linspace(0,99,100);
t3 = linspace(0,149,150);

den = [1 2 2 1];

[numd, dend] = bilinear(1, den, 5); %아날로그 시스템 이산화
[numd2, dend2] = bilinear(1, den, 10); %%아날로그 시스템 이산화

[zero, pole] = tf2zp(1,den);
[zero2, pole2] = tf2zp(numd,dend); %전달함수로부터 zero와 pole을 구함
[zero3, pole3] = tf2zp(numd2,dend2); %전달함수로부터 zero와 pole을 구함

Zero = [zero2 zero3];
Pole = [pole pole2 pole3];
```

3차 Butterworth LPF를 차수를 입력하여 구현하고 표본화 주파수를 사용, 시스템에 bilinear transformation을 적용하여(bilinear 함수 사용) 디지털 필터로 변환하였다. 변환한 디지털 필터의 전달함수로부터 zero와 pole을 구했다.

(2) 실습 DEMO (1)의 아날로그 필터와 디지털 필터의 pole-zero plot을 표시하라. 각 시스템의 안정성을 판단하라. (반지름 1인 원과 함께 표시하라.) (tf2zp 함수 사용) (그림 6 참고)

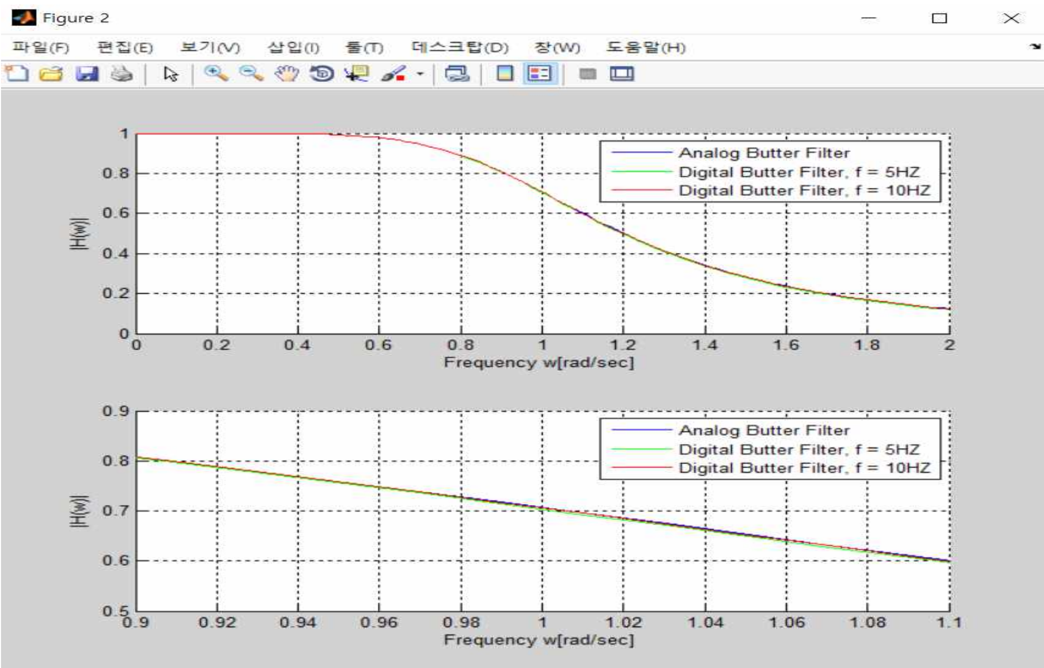
```
figure(1)
for i = 1 : length(Pole)
    subplot(2,2,i)
    plot(cos(n), sin(n), '--k');
    hold on; grid on;
    plot(real(Pole(:,i)), imag(Pole(:,i)) , 'rx', 'Markersize',20);

    if(i == 1)
        plot(real(zero), imag(zero) , 'o', 'Markersize',20);
    end

    if(i>1)
        plot(real(Zero(:,i-1)), imag(Zero(:,i-1)) , 'o', 'Markersize',20);
    end
    xlabel('Real(s)'); ylabel('Image(s)');
    xlim([-1.1 1.1]);
end

```



3차 아날로그 Butterworth LPF는 이전 실습과 같이 극점이 좌반 평면에 등 간격으로 위치해 있는 것을 볼 수 있으므로 이는 안정한 시스템이라는 것을 알 수 있다. 디지털 Butterworth LPF의 pole-zero plot은 위와 같다. (위쪽 두번째 그림이 fs = 5, 아래 그림이 fs = 10 으로 변환한 것) 디지털 LPF도 모든 극점이 단위 원 안에 위치 하고 있으므로 안정된 시스템이라고 볼 수 있다. fs = 10일 때 극점의 위치가 fs = 5일 때에 비해 더 모여 있는 것을 볼 수 있는데, 샘플링 주파수가 클수록 아날로그에서 더 많은 신호를 가져오고, 주기가 작아지기 때문에 극점이 모여 있는 것이다.

(3) 실습 DEMO (1)의 아날로그 필터와 디지털 필터의 주파수 응답의 크기를 그래프에 표시하라. (아날로그 주파수 ω에 대해 표시할 것.) (freqs, freqz 함수 사용) (그림 7 참고)

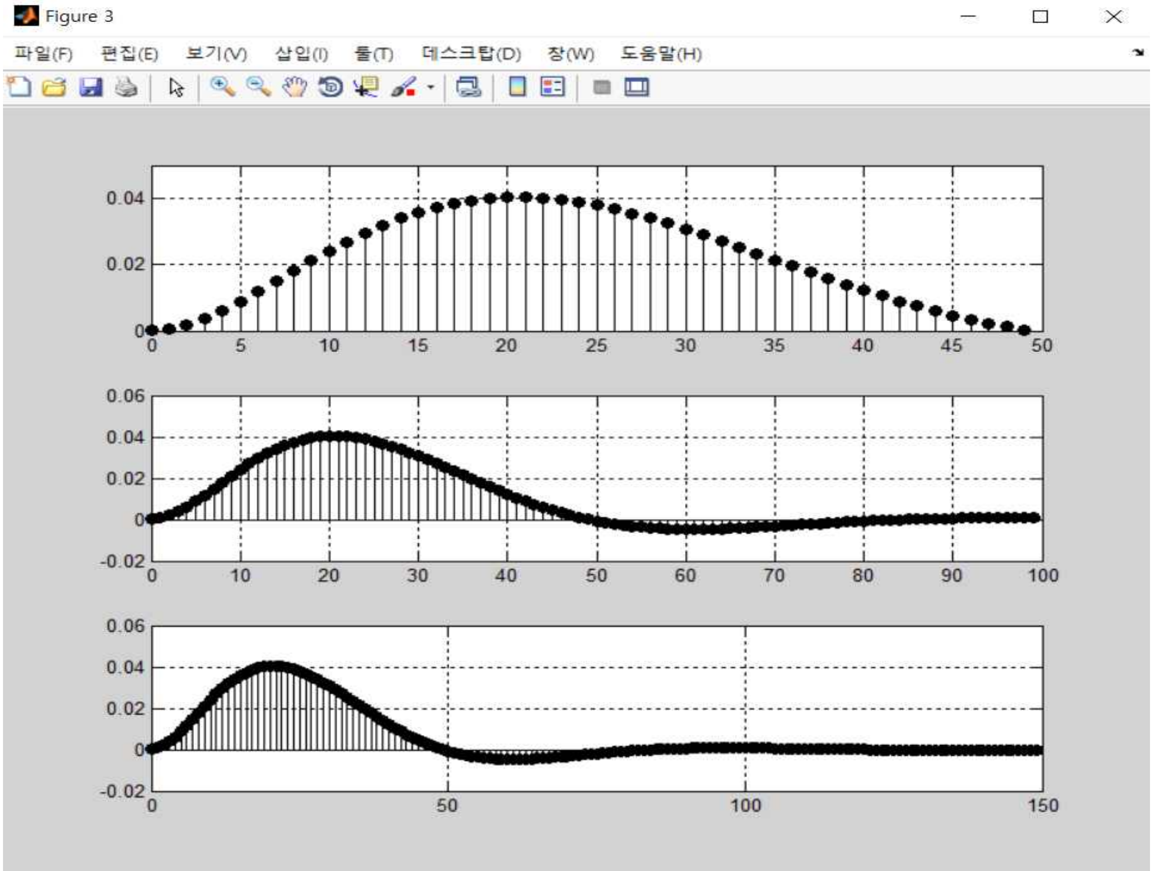
```
H = freqs(1, den, w); % 1,den에 대한 주파수 응답
H2 = freqz(numd, dend, w/5); %numd, dend에대한 주파수 응답
H3 = freqz(numd2, dend2, w/10); %num2, den2에대한 주파수 응답

figure(2)
for i = 1 : 2
    subplot(2,1,i);
    hold on; grid on;
    plot(w, abs(H),'b');
    plot(w, abs(H2),'g');
    plot(w, abs(H3),'r');

    if(i > 1)
        xlim([0.9 1.1]);
    end

    xlabel('Frequency w[rad/sec]'); ylabel('|H(w)|');
    legend('Analog Butter Filter','Digital Butter Filter, f = 5HZ','Digital Butter Filter, f = 10HZ');
end
```



아날로그 주파수 w에 대하여 주파수를 각각 표시했다.
위의 그래프는 0부터 2까지 범위를 표현했고, 아래의 그래프는 구간을 확대하여(0.9부터 1.1까지) 나타내었다. 그래프 확인이 어려워 각 주파수 응답의 굵기를 다르게 지정하여 구분해 주었다. 아날로그 필터는 freqs 함수를, 디지털 필터는 freqz 함수를 사용하여 주파수 응답을 구해주었다.
굵기를 다르게 지정하여 확인해 보았는데, 아날로그 필터가 제일 위에 있고 그다음이 fs = 10일 때의 디지털 필터, 마지막으로 fs = 5일때의 디지털 필터가 있다. 이 것을 보았을 때 샘플링 주파수가 클수록 아날로그에서 더 많은 신호를 가져오기 때문에 fs가 커질수록 아날로그 필터의 응답과 흡사해지는 것을 확인할 수 있다.

(4) 실습 DEMO (1)에서 $f_s = 10$ Hz일때 구한 디지털 필터의 임펄스 응답을 구하고 $n = 0, 1, \cdots, L-1$에 대해 그래프에 표시하라. L값을 달리하며 임펄스 응답을 관찰하라. 이 시스템의 임펄스 응답이 유한한지 무한한지 판단하라. (filter 함수 사용) (그림 8 참고)

```
L = [ones(1,1), zeros(1,49)]; %배열의 첫요소를 1로 만듬
L2 = [ones(1,1), zeros(1,99)]; %배열의 첫요소를 1로 만듬
L3 = [ones(1,1), zeros(1,149)]; %배열의 첫요소를 1로 만듬

y = filter(numd2, dend2, L); %디지털 시스템의 출력을 구함
y2 = filter(numd2, dend2, L2); %디지털 시스템의 출력을 구함
y3 = filter(numd2, dend2, L3); %디지털 시스템의 출력을 구함

figure(3)
subplot(311);
stem(t, y, 'k', 'filled');
xlim([0 50]); ylim([0 0.05]); grid on;
subplot(312);
stem(t2, y2, 'k', 'filled');
xlim([0 100]); ylim([-0.02 0.06]); grid on;
subplot(313);
stem(t3, y3, 'k', 'filled');
xlim([0 150]); ylim([-0.02 0.06]); grid on;
```



y2

1x100 double

		83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100
1	e-04	-4.1351e-04	-2.3895e-04	-7.8463e-05	6.7731e-05	1.9958e-04	3.1719e-04	4.2076e-04	5.1064e-04	5.8726e-04	6.5113e-04	7.0286e-04	7.4309e-04	7.7252e-04	7.9189e-04	8.0197e-04	8.0354e-04	7.9738e-04	7.8429e-04

y3

1x150 double

	132	133	134	135	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
1	1.2732e-04	-1.2952e-04	-1.3027e-04	-1.2971e-04	-1.2797e-04	-1.2518e-04	-1.2147e-04	-1.1695e-04	-1.1175e-04	-1.0598e-04	-9.9750e-05	-9.3164e-05	-8.6319e-05	-7.9303e-05	-7.2199e-05	-6.5084e-05	-5.8026e-05	-5.1088e-05	-4.4322e-05

L에 대하여 배열을 만들어서 각 배열의 요소에 50, 100, 150을 넣어 길이를 구하고, 디지털 시스템의 출력을 구한 후 그래프로 나타내었다.

그래프를 보면 점점 0에 수렴하는 것을 볼 수 있다. 하지만 작업 공간을 확인하면 0이 아닌 값을 가지고 있는 것을 볼 수 있다. 즉 이 응답은 무한하다.

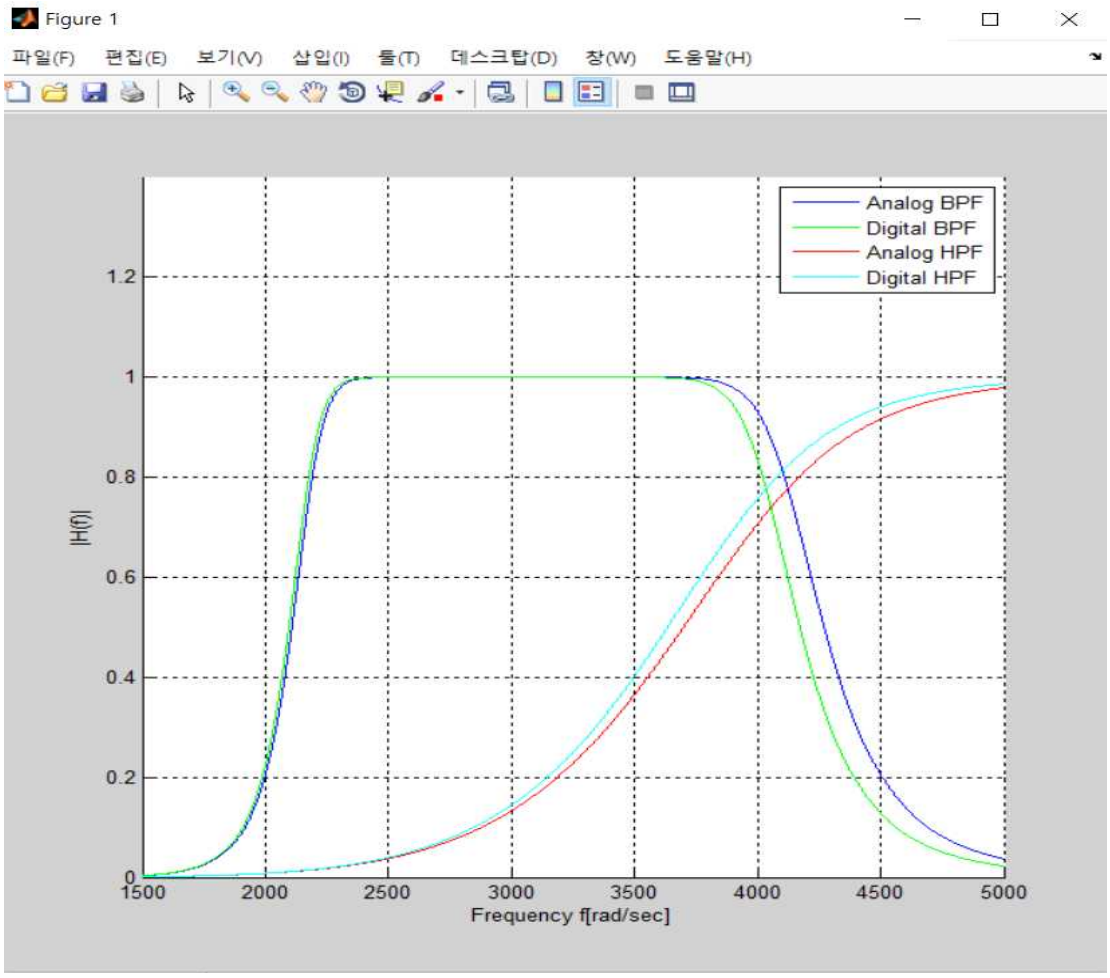
3.2 디지털 BPF, HPF 설계

(1) **실습** 7차 아날로그 Butterworth 프로토타입 필터를 이용해 다음과 같은 아날로그 필터를 설계하라. (아날로그 프로토타입 필터 실습의 실습 3.2 (4) 참조) (buttap, lp2bp, lp2hp 함수 사용)

1. BPF, passband $= [2000, 4000]$ Hz
2. HPF, cutoff 주파수 $f_c = 4000$ Hz

(2) **실습** **DEMO** 표본화 주파수 $f_s = 50,000$ Hz일때 bilinear transformation을 이용해 (1)의 BPF와 HPF를 이산화하여 IIR 필터를 설계한 뒤 주파수 응답을 아날로그 필터와 비교하라. (그림 9 참고)

```
w = linspace(1500, 5000*2*pi, 10000); %구간 설정
fs = 50000;
w2 = w/2/pi;
[zero, pole, k] = buttap(7); %7차 Butterworth 필터
[zero_h, pole_h] = zp2tf(zero, pole, k); %전달함수를 구하여 zero_h, pole_h에 저장

[BPF, BPF2] = lp2bp(zero_h, pole_h, 3000*2*pi, 2000*2*pi);
[HPF, HPF2] = lp2hp(zero_h, pole_h, 4000*2*pi);

H = freqs(BPF, BPF2, w);
H2 = freqs(HPF, HPF2, w);

[numd, dend]=bilinear(BPF, BPF2, fs);
[numd2, dend2]=bilinear(HPF, HPF2, fs);

H_1 = freqz(numd, dend, w/fs);
H_2 = freqz(numd2, dend2, w/fs);

hold on; grid on;
plot(w2, abs(H),'b');
plot(w2, abs(H_1),'g');
plot(w2, abs(H2),'r');
plot(w2, abs(H_2),'c');
ylim([0 1.4]);xlim([1500 5000]);
legend('Analog BPF','Digital BPF','Analog HPF','Digital HPF');
xlabel('Frequency f[rad/sec]'); ylabel('|H(f)|');
```



디지털 라디안 주파수 w(Omega, 2*pi*f)로 아날로그와 디지털 필터의 규격을 맞춰주었다. 7차 아날로그 Butterworth 필터를 이용하여 필터를 설계한 후에 lp2bp, lp2hp 함수를 이용하여 freqs 함수로 아날로그 주파수 응답을 구했다. BPF는 각각의 주파수 대역의 중심주파수와 대역폭을 함수에 대입하여 구간을 정했고 HPF는 cutoff를 4000Hz으로 정하여 설계했다. 설계한 두 필터를 fs = 50000Hz 으로 주고 bilinear 함수를 이용하여 이산화시킨 다음, freqz 함수를 이용해서 디지털 시스템의 주파수 응답을 구해주었다. 디지털 주파수는 아날로그 주파수를 샘플링 주파수로 나눈 값이기 때문에 w을 fs로 나누어서 디지털 주파수의 응답을 구했다. 상당히 유사하게 디지털 필터가 설계됨을 확인할 수 있다.

3.3 신호의 필터링

(1) **실습** 7차 아날로그 Butterworth 프로토타입 필터를 이용해 다음과 같은 아날로그 필터를 설계하라. (아날로그 프로토타입 필터 실습의 실습 3.2 (4) 참조) (buttap, lp2lp 함수 사용)

- LPF, cutoff 주파수 $f_c = 300$ Hz

(2) **실습** 표본화 주파수 $f_s = 3000$ Hz일때 bilinear transformation을 이용해 (1)의 LPF를 이산화하여 IIR 필터를 설계하라.

(3) **실습** 다음과 같은 신호 $x[n]$을 발생하라.

$$x[n] = \cos\left(2\pi \frac{f_1}{f_s} n\right) + \cos\left(2\pi \frac{f_2}{f_s} n\right)$$

- $f_1 = 100$ Hz, $f_2 = 500$ Hz, $f_s = 3000$ Hz
- $n = 0, 1, 2, \cdots, N$, $N = 500$

(4) **실습** (3)의 신호 $x[n]$을 (2)에서 설계한 디지털 필터에 입력하여 출력 $y[n]$을 구하라. (filter 함수 사용)

(5) **실습** $x[n]$과 $y[n]$의 크기 스펙트럼을 구하라. (myfun_SA 함수 사용, 표본화 실습 참고)

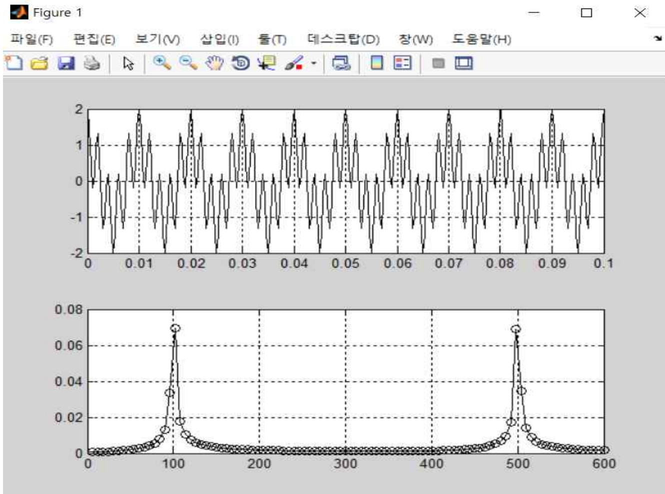
- $t = nT_s = \frac{n}{f_s}$ 관계를 이용하라.

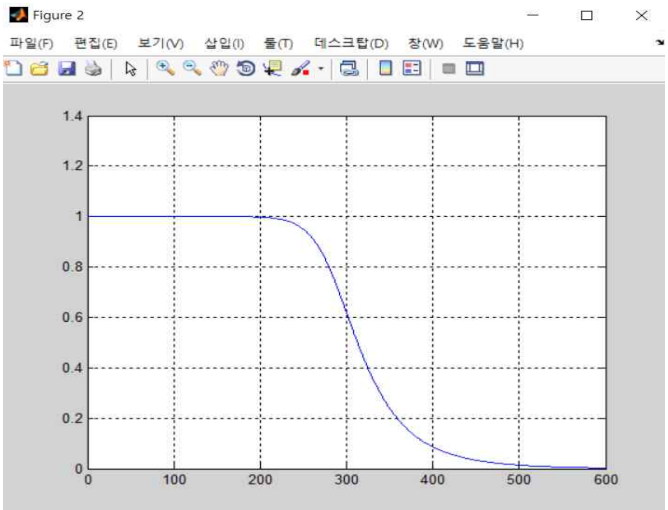
```
n = linspace(0, 499, 500);
w = linspace(0, 600*2*pi, 1000);
f1 = 100; f2 = 500; fs = 3000;
T = n/fs;
x = cos(2*pi*(f1/fs)*n)+cos(2*pi*(f2/fs)*n); %이산신호 x[n]

[zero, pole, k] = buttap(7); %7차 Butterworth 필터
[zero_h, pole_h] = zp2tf(zero, pole, k); %전달함수를 구하여 zero_h, pole_h에 저장

[F, X] = myfun_SA(T,x); %x[n]의 크기 스펙트럼

[LPF, LPF2] = lp2lp(zero_h, pole_h, 300*2*pi);
[numd, dend]=bilinear(LPF, LPF2, fs); %아날로그 시스템 이산화
H = freqz(numd, dend, w/fs); %디지털 주파수 응답


y = filter(numd, dend, x); %디지털 시스템의 출력
[F2, X2] = myfun_SA(T,y); %y[n]의 크기 스펙트럼
```

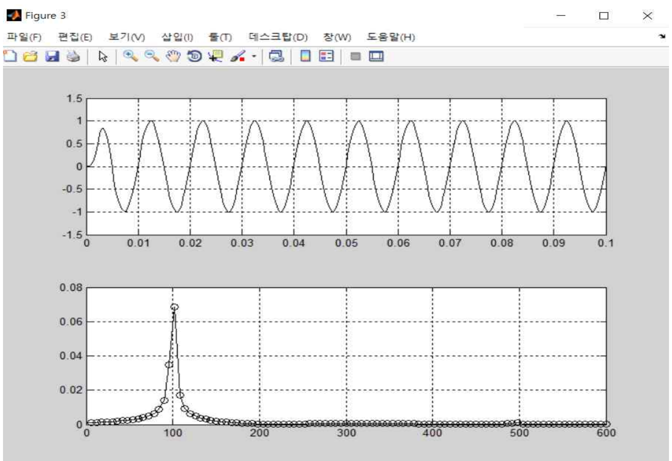
7차 아날로그 Butterworth LPF를 구한 다음, lp2lp 함수를 이용하여 cufoff 주파수가 300인 필터를 설계했다. 그 이후 bilinear 함수를 이용하여 이산화시킨 다음 freqz 함수를 이용해서 디지털 시스템의 주파수 응답을 구해주었다. 디지털 주파수는 아날로그 주파수를 샘플링 주파수로 나눈 값이기 때문에 w을 fs로 나누어서 디지털 주파수의 응답을 구했다. 신호의 길이 n을 linspace로 지정했고, x[n]과 y[n]의 크기 스펙트럼을 구하기 위하여 시간 변수 t를 주어진 관계를 통하여 정하였다. y[n]은 이산신호 x[n]을 filter 함수를 이용하여 디지털 시스템으로 출력한 것이고, 각 신호의 크기 스펙트럼을 myfun_SA 함수로 구하였다.

(6) 실습 DEMO 입력신호 $x[n]$과 크기 스펙트럼, 출력신호 $y[n]$과 크기 스펙트럼, 디지털 필터 $H_d(z)$의 주파수 응답의 크기를 그래프에 표시하라. (그림 10 참고)

```
figure(1);
subplot(211);
plot(T, x, '-k');
xlim([0 0.1]); grid on;

subplot(212);
plot(F, abs(X), '-ok');
xlim([0 600]); grid on;

figure(2);
plot(w/2/pi, abs(H));
grid on;

figure(3);
subplot(211);
plot(T, y, 'k');
xlim([0 0.1]); grid on;
ylim([-1.5 1.5]);

subplot(212);
plot(F2, abs(X2), '-ok');
xlim([0 600]); grid on;

```







figure(1) subplot(2,1,1) : 입력신호 x[n] subplot(2,1,2) : 입력신호 x[n]의 크기 스펙트럼

figure(2) : H_d(z)의 주파수 응답의 크기

figure(3) subplot(2,1,1) : 출력신호 y[n] subplot(2,1,2) : 출력신호 y[n]의 크기 스펙트럼

시간 축 입 출력 신호를 분석해보면, x[n]은 두 개의 주파수 성분으로 합쳐진 주기 신호 인데, 출력 y[n]을 보면 마치 한 개의 주파수 성분만 가지고 있는 주기 신호처럼 보인다. 왜냐하면, 100Hz와 500Hz 성분을 가지고 있는 x[n]의 스펙트럼이 LPF를 만나 고주파는 저지되고 저주파만 통과가 되어 y[n]의 스펙트럼과 같은 결과가 나왔기 때문이다. 이상적인 LPF가 아니기 때문에 저지 대역에 찌꺼기가 남게 된다.
figure(3)의 subplot(2,1,2)의 그래프, y[n]의 스펙트럼 500Hz 부분을 자세히 보면 아주 미세하게 아직 500Hz 성분이 남아있는 것을 확인할 수 있다.