

Chapter 6

필터링



학습목표



이번 장에서 다루게 되는 내용은 다음과 같습니다.

이번 강의 학습 목표

- 1) 필터의 종류에 대해 이해한다.
- 2) FIR과 IIR 필터의 특성에 대해 이해한다.
- 3) FIR 필터 설계에 대해 이해한다.
- 4) IIR 필터 설계에 대해 이해한다.
- 5) Matlab을 이용한 예제를 통해 Chebyshev 필터를 이해한다.
- 6) Matlab을 이용한 예제를 통해 창함수를 이용한 필터링에 대해 이해한다.
- 7) Matlab을 이용한 예제를 통해 주파수 분석에 대해 이해한다.

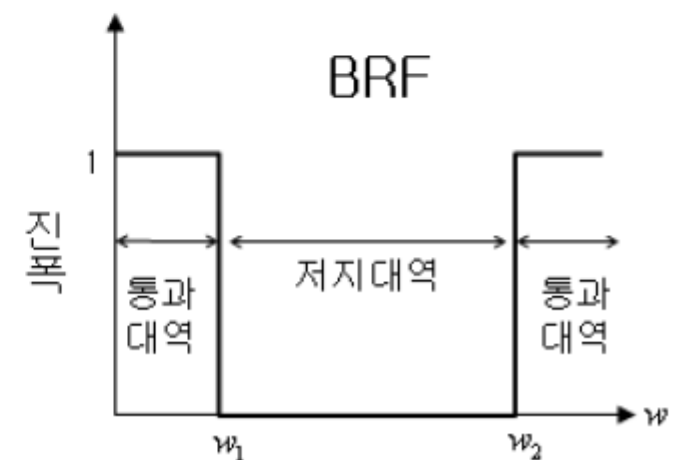
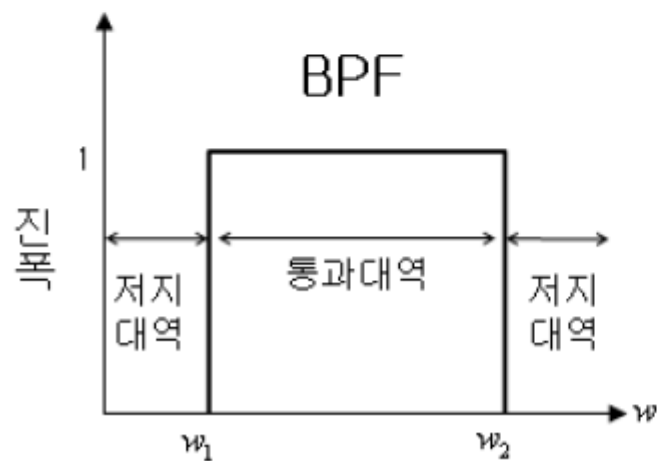
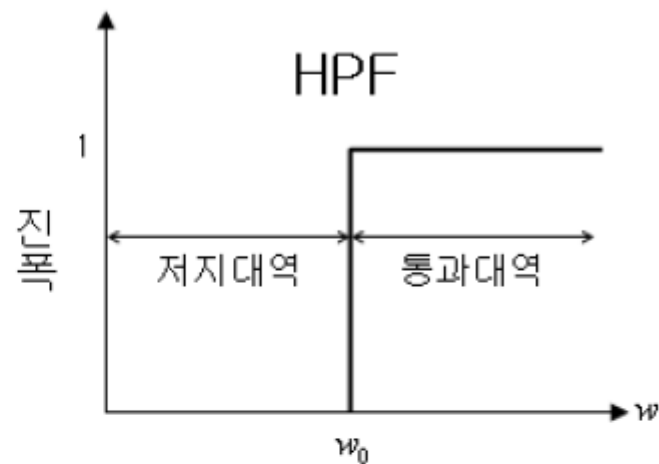
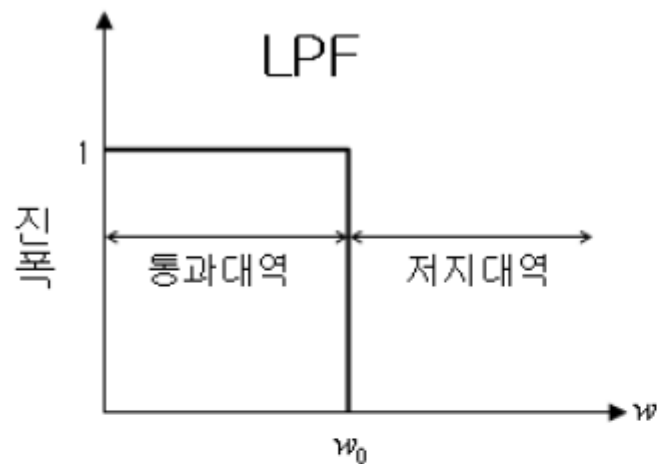


필터의 종류 (1)

- 주파수 선택 특성에 따라 구분.
 - 저역통과 필터(low pass filter : LPF)
 - 고역통과 필터(high pass filter : HPF)
 - 대역통과 필터(band pass filter : BPF)
 - 대역저지 필터(band rejection filter : BRF)



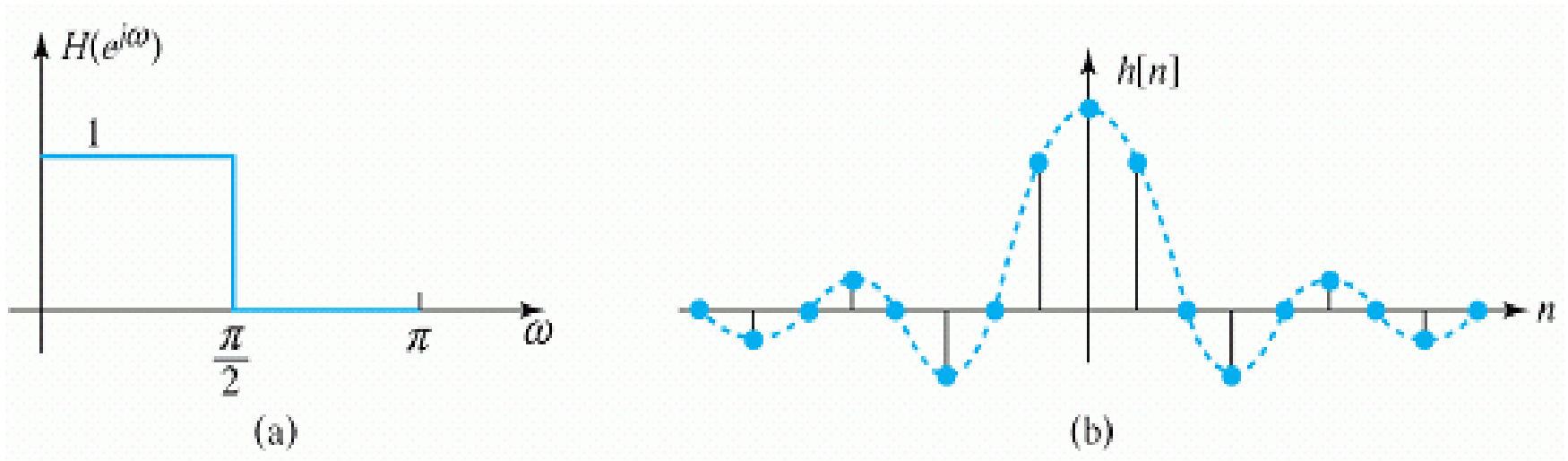
필터의 종류 (2)



필터의 형태



필터의 종류 (3)

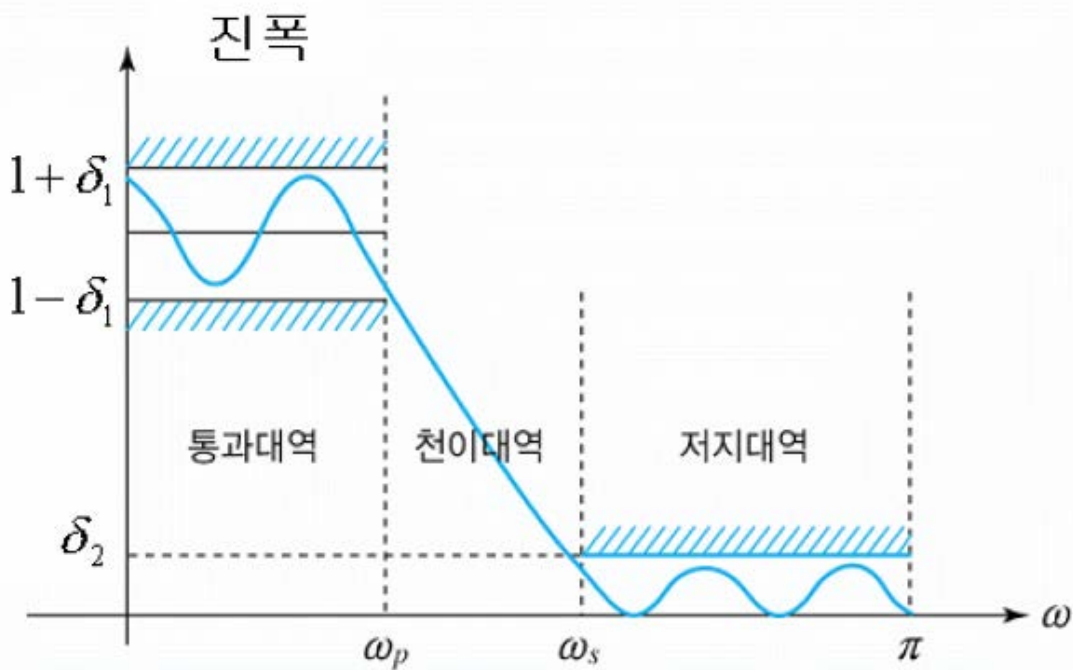


이상적인 저역 통과 필터

(a) 주파수 응답, (b) 임펄스 응답



필터의 종류 (4)



저역통과 필터의 진폭응답

- Ⓐ ω_p : 통과대역 주파수
- Ⓑ δ_1 : 통과대역 리플 크기
- Ⓒ ω_s : 저지대역 주파수
- Ⓓ δ_2 : 저지대역 리플 크기
- Ⓔ f_s : 샘플링 주파수
- Ⓕ N : 필터의 차수



디지털 필터의 설계

(a) 원하는 필터의 사양 결정

: 진폭응답(통과대역, 천이대역, 저지대역), 위상응답(위상, 군 지연), 샘플링주파수와 양자화 비트 수 등 결정

(b) 전달함수 결정 : 필터의 계수값을 결정

(c) 필터의 구조 결정

: FIR인 경우 : 직접형인 횡단 구조와 격자 구조가 있음.

IIR의 경우 : 직접형, 종속형, 병렬형 구조 있음.

(d) 유한어장(word length) 영향 분석

(e) 필터를 소프트웨어나 하드웨어로 구현

: 고속의 디지털 신호처리 전용 처리기를 이용한 디지털 필터를 구현.



FIR과 IIR 필터 특성

- 원하는 필터의 사양이 결정 ➔ FIR 필터 혹은 IIR 필터로 설계 할지 결정.

[표 6-1] FIR 필터와 IIR 필터의 비교

구 분	FIR	IIR
선형성	선형 위상 특성	비선형 위상 특성
안정성	높음	낮음
계산량	예리한 필터 특성을 위해서는 많은 계수가 요구됨	적은 수의 계수로도 FIR과 비슷한 특성을 가짐
계수 오차	적은 반올림 계수 오차	큰 반올림 계수 오차



FIR 필터의 설계

- FIR 필터 설계 방법.
 - 창(window) 함수 방법.
 - 주파수 샘플링 방법.



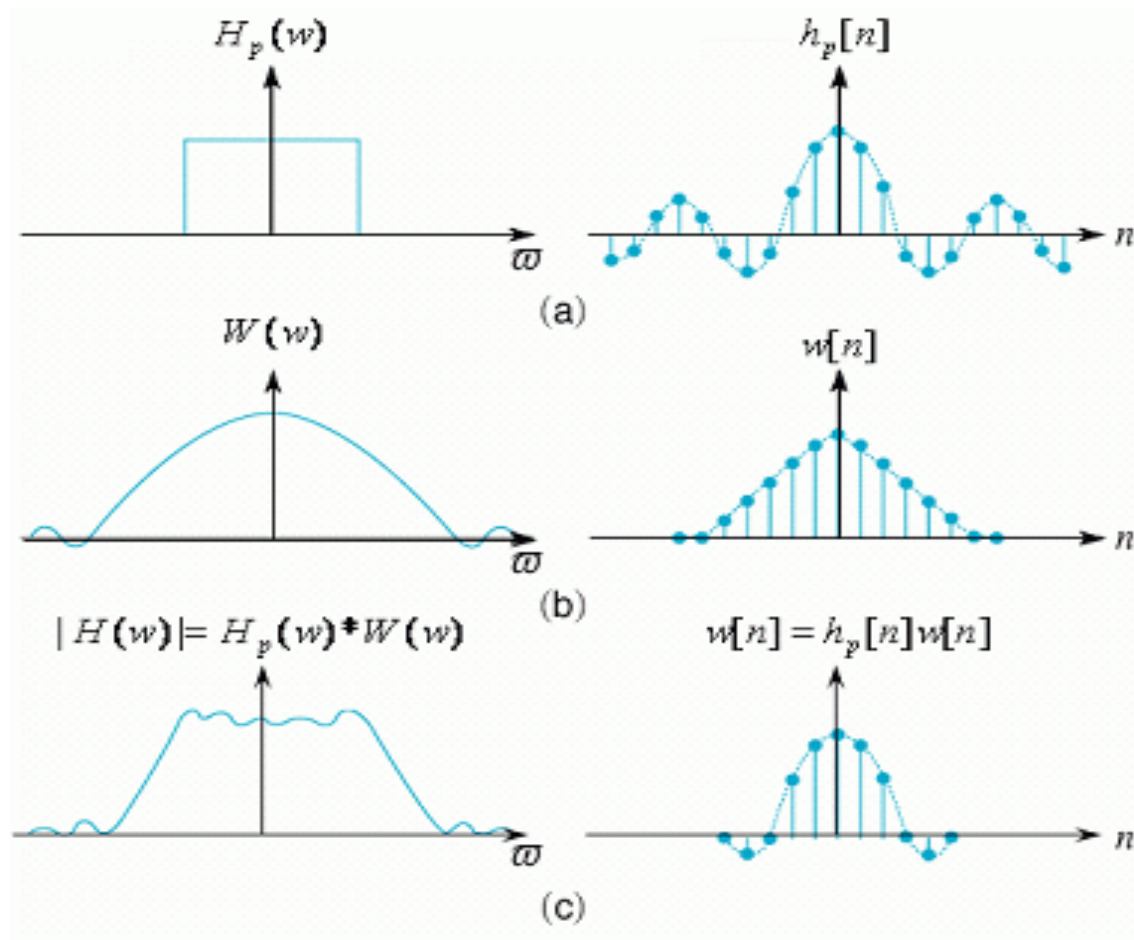
FIR 필터의 설계 - 창함수 방법 (1)

□ 창(window) 함수 방법

- 무한 임펄스 응답을 갖는 이상적인 LPF의 전달함수에 정해진 window 함수를 컨볼루션하여 원하는 특성의 전달함수를 얻음.
- 시간영역에서는 이상적 LPF의 임펄스 응답에 창함수를 곱하여 원하는 임펄스 응답을 얻고, 얻어진 임펄스 응답을 필터의 계수값으로 정함.
- 창함수의 종류에 따라 필터의 특성은 달라짐.



FIR 필터의 설계 - 창함수 방법 (2)



창함수를 이용하는 방법



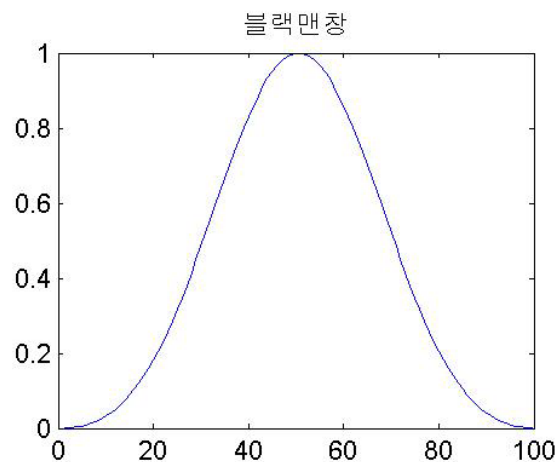
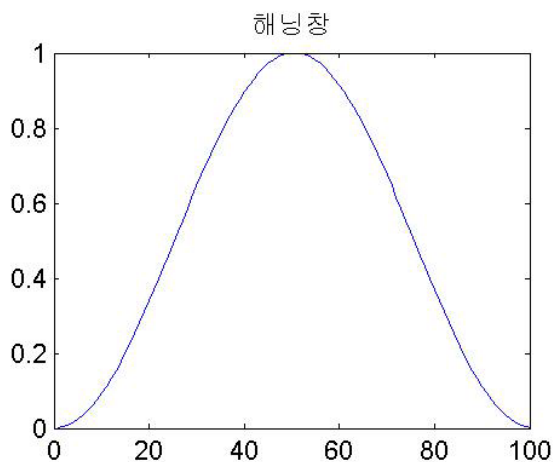
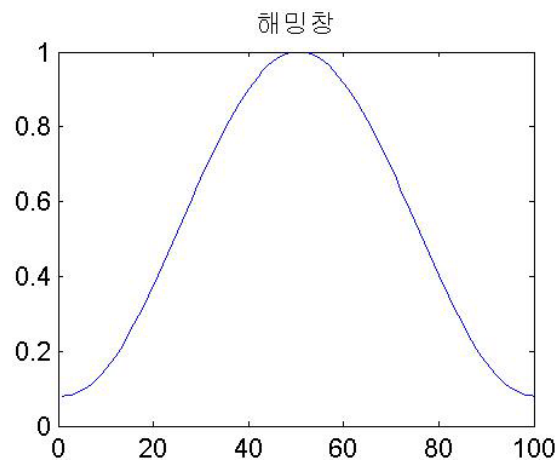
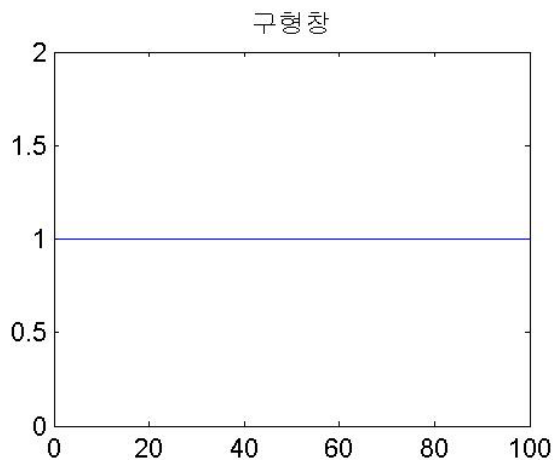
FIR 필터의 설계 - 창함수 방법 (3)

[표 6-2] 대표적인 창함수

창함수명	함수(구간 : $-\frac{N}{2} \leq n \leq \frac{N}{2}$)
구형창 (Rectangular window)	$w[n] = 1$
해밍창 (Hamming window)	$w[n] = 0.54 + 0.46 \cos\left(\frac{2\pi n}{N}\right)$
해닝창 (Hanning window)	$w[n] = 0.5 + 0.5 \cos\left(\frac{2\pi n}{N}\right)$
블랙맨창 (Blackman window)	$w[n] = 0.425 + 0.5 \cos\left(\frac{2\pi n}{N-1}\right) + 0.08 \cos\left(\frac{4\pi n}{N-1}\right)$



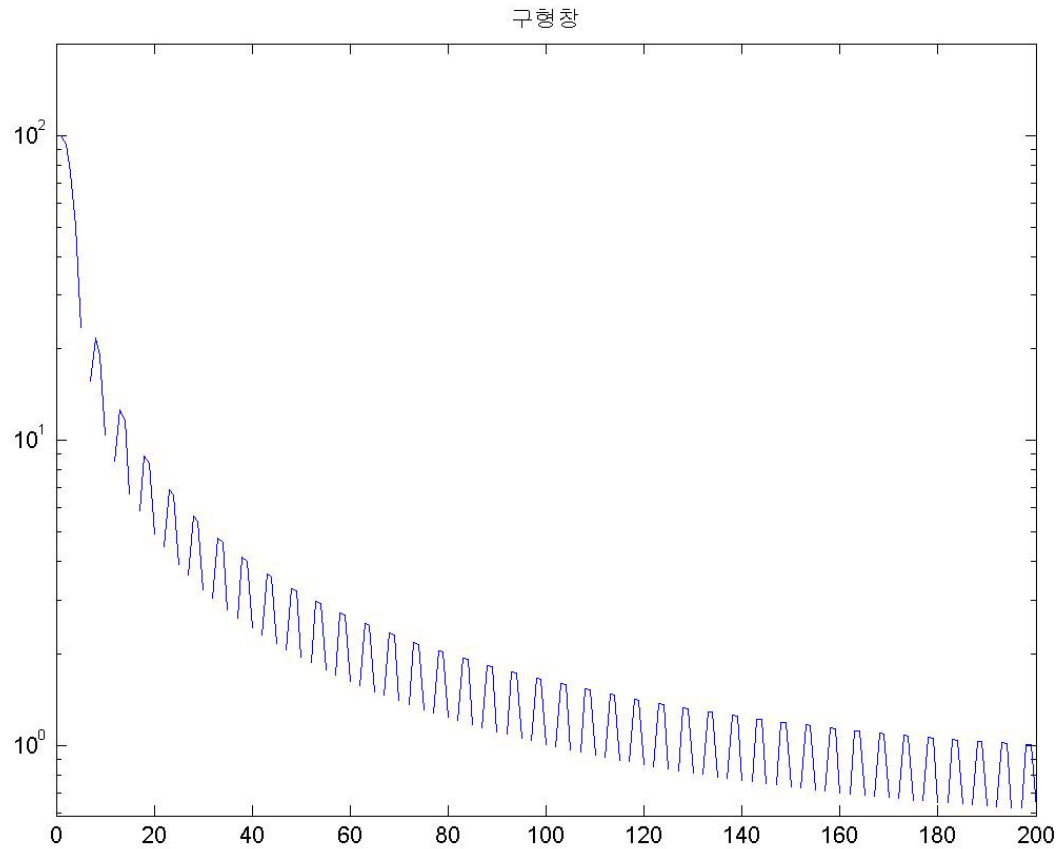
FIR 필터의 설계 - 창함수 방법 (4)



창함수의 시간 특성



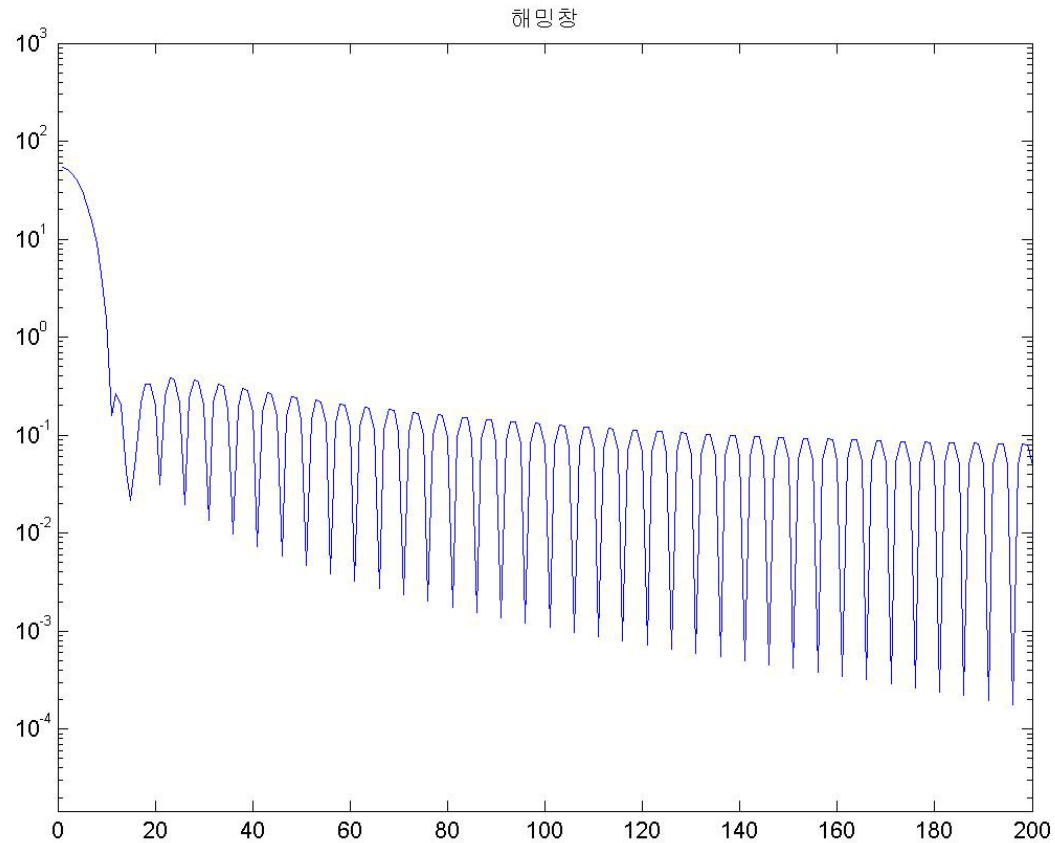
FIR 필터의 설계 - 창함수 방법 (5)



구형 창함수에 대한 감쇠 특성



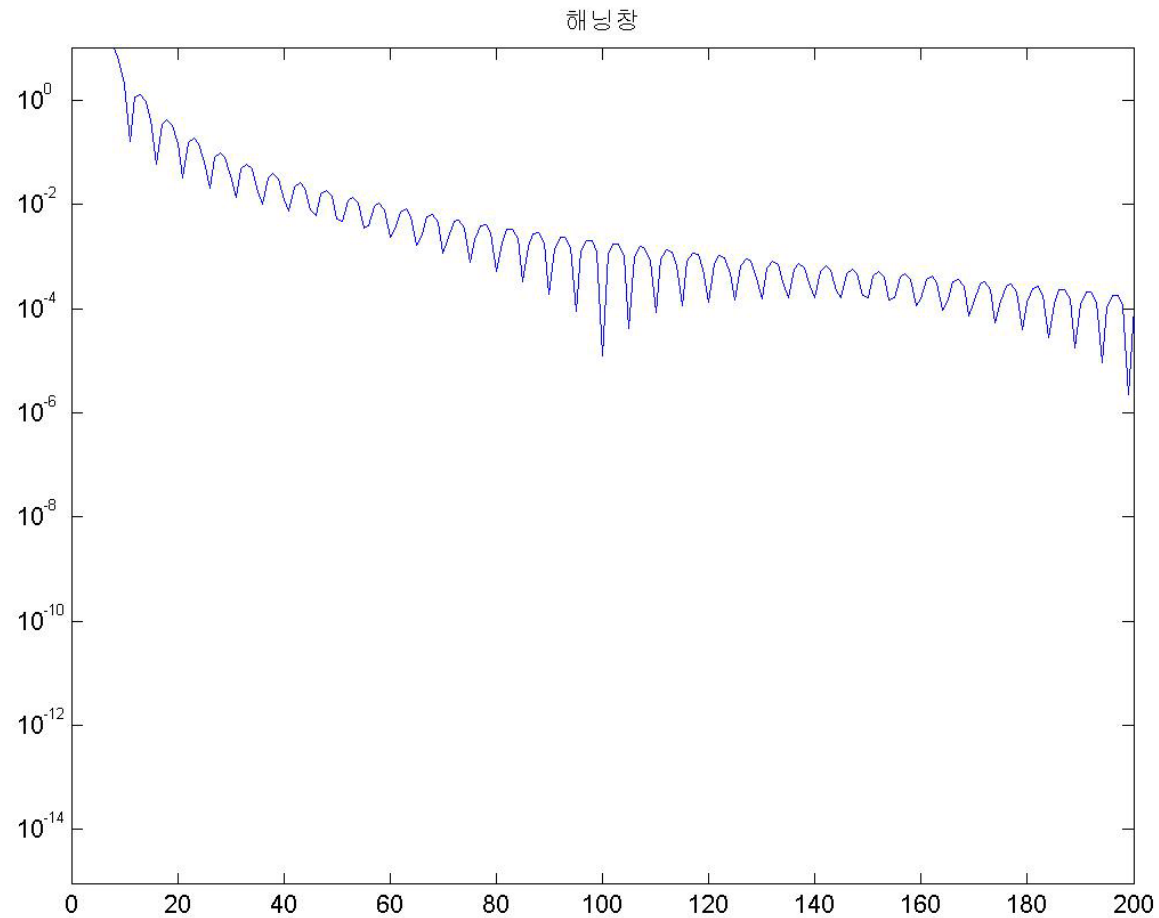
FIR 필터의 설계 - 창함수 방법 (6)



해밍 창함수에 대한 감쇠 특성



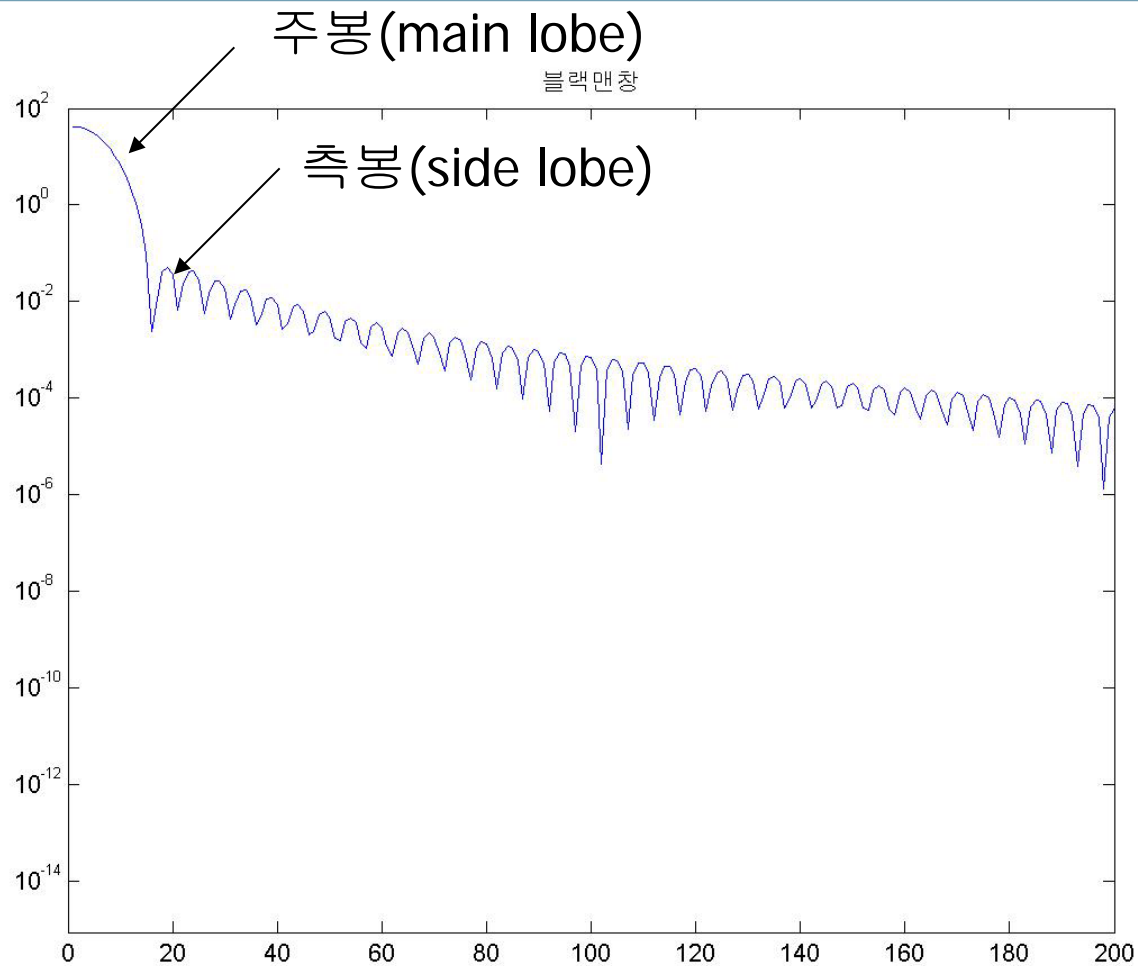
FIR 필터의 설계 - 창함수 방법 (7)



해닝 창함수에 대한 감쇠 특성



FIR 필터의 설계 - 창함수 방법 (8)



블랙맨 창함수에 대한 감쇠 특성



FIR 필터의 설계 - 창함수 방법 (8)

- ❑ 블랙맨 창 : 측봉의 감쇠 특성은 우수하지만 주봉의 폭이 커서 천이대역의 폭이 상대적으로 넓어짐, 그러나 통과대역의 맥동을 충분히 줄일 수 있음.
- ❑ 구형창, 해밍창 : 블랙맨 창에 비해 저지대역의 감쇠 특성이 우수하지는 않지만 상대적으로 측봉의 폭이 좁아 천이대역 폭이 좁음, 그러나 통과대역의 맥동이 커짐. ➔ 측봉의 크기를 충분히 감쇠시키면 저지대역의 주파수 특성을 좋게 할 수 있음.



FIR 필터의 설계 - 주파수 샘플링 방법 (1)

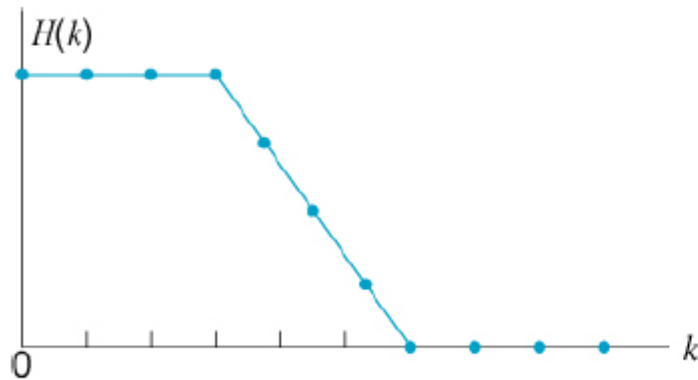
- 필터의 구성이 주파수 영역에서 주어진 경우 주파수 축 상에서 N개의 샘플치를 선택하여 역 이산 푸리에 변환을 수행함으로써 필터의 임펄스 응답을 구하는 방법
- DFT의 정의식

$$H[k] = \sum_{n=0}^{N-1} h[n] W_N^{nk}, \quad 0 \leq k \leq N-1$$

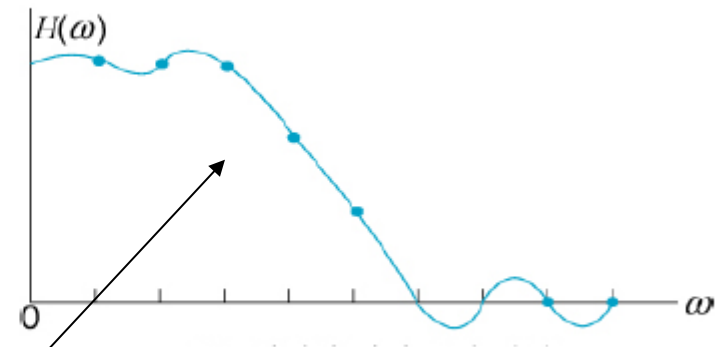
$$h[n] = \frac{1}{N} \sum_{k=0}^{N-1} H[k] W_N^{-nk}, \quad 0 \leq n \leq N-1$$



FIR 필터의 설계 - 주파수 샘플링 방법 (2)



(a) 근사화된 저역통과 필터



(b) 설계된 저역통과 필터

샘플치가 충분하지 않으면 통과대역과 저지대역의 맥동이 심하게 됨을 알 수 있음



IIR 필터의 설계

□ IIR 필터 설계 방법

: 원하는 특성의 아날로그 필터를 먼저 설계한 후 디지털 필터로 변환시키는 방법.

□ 임펄스 불변법(Impulse Invariance)

□ 쌍1차 Z변환(Bilinear Z transformation : BZT)



IIR 필터의 설계 – 임펄스 불변법 (1)

- 아날로그 필터의 임펄스 응답에 대응하는 디지털 필터의 임펄스 응답을 구하여 Z 영역에서 필터의 전달함수를 구함.
- 아날로그 필터의 전달함수가 ①식과 같이 주어진 경우 임펄스 응답은 ②식과 같다.

$$H(s) = \sum_{k=1}^N \frac{C_k}{s - p_k} \quad \dots \textcircled{1}$$

$$h(t) = \sum_{k=1}^N C_k e^{p_k t} \quad \dots \textcircled{2}$$

여기서 p_k : 아날로그 필터의 극점.



IIR 필터의 설계 – 임펄스 불변법 (2)

$$h(t) = \sum_{k=1}^N c_k e^{p_k t} \quad \dots \quad \textcircled{1}$$

- ①식에 대응하는 디지털 필터의 임펄스 응답을 구하면 ②식과 같다.

$$h[n] = \sum_{k=1}^N c_k e^{p_k T n} \quad \dots \quad \textcircled{2}$$

- ②식을 Z 변환을 이용하여 디지털 전달함수를 구하면

$$H[z] = \sum_{k=1}^N \frac{c_k}{1 - e^{p_k T} z^{-1}}$$

- 샘플링 시점에서 아날로그 필터와 디지털 필터의 임펄스 특성 동일

∴ 샘플링 주파수를 충분히 크게 할 필요가 있음.



IIR 필터의 설계 - 쌍 1차 Z 변환법

- 쌍 1차 Z 변환법은 아날로그 필터의 전달함수 $H(s)$ 를 쌍1차 변환을 사용하여 디지털 필터의 전달함수 $H[z]$ 로 변환하는 방법.
- S 평면에서 Z 평면으로의 매핑은 ①에 의해 이루어짐.

$$s = -\frac{2}{T} \left(\frac{1-z^{-1}}{1+z^{-1}} \right) \quad \dots \quad \textcircled{1}$$

- ①식을 쌍1차 변환이라함.
 - T : 샘플링 주파수
- 아날로그 필터의 전달함수 s 대신 ①식을 대입하여 디지털 필터의 전달함수를 구할 수 있음.



Matlab을 이용한 필터링 I

□ IIR필터를 이용한 필터링을 실습해보도록 하자.

다음과 같은 조건을 갖는 디지털 필터를 이용하여 1kHz, 2kHz, 4kHz의 sin신호가 합성된 신호를 대상으로 필터링을 수행해보자.

□ 샘플링 주파수 : 10kHz

□ 필터 종류 : 제 1형 Chebyshev 고역 통과 필터

□ 저지 대역 주파수 : 2.5kHz

□ 통과 대역 주파수 : 3kHz

□ 저지대역 리플 : 60dB

□ 통과대역 리플 : 0.1dB



Matlab을 이용한 필터링 I

```
Fs=10000;    % 샘플링 주파수
N=200;       % 샘플 개수
t=[1:N]/Fs;  % 시간 축 값
w=[0:N-1]/N*Fs; % 주파수 축 값
x=sin(2*pi*1000*t)+sin(2*pi*2000*t)+sin(2*pi*4000*t); % 입력 신호
s=abs(fft(x)); % 입력 신호에 대한 주파수 특성
%----- 필터 설계 -----
ws=2500/(Fs/2);
wp=3000/(Fs/2);
Rp=0.1;
Rs=60;
[Nth, Wn]=cheb1ord(wp,ws,Rp,Rs); % chebyshev 필터의 차수와 차단 주파수 설정
B, A]=cheby1(Nth,Rp,Wn,'high'); % 필터 설계
H=freqz(B,A,w,Fs); % 주파수 특성
x_filtered=filter(B,A,x); % 필터링 하기
```



Matlab을 이용한 필터링 I

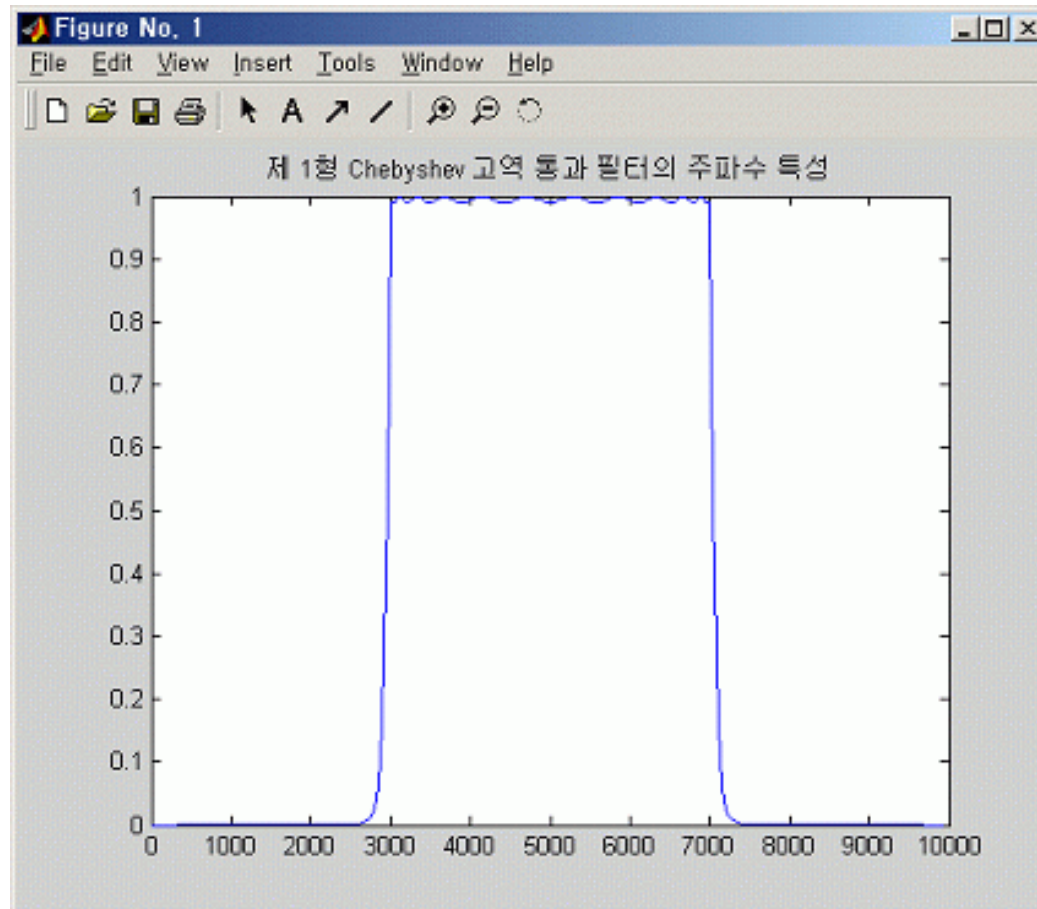
```
% 그림 출력
figure
plot(w,abs(H)); % 필터의 주파수 특성
title('제 1형 Chebyshev 고역 통과 필터의 주파수 특성');

figure
subplot(211);
plot(x);          %입력 신호 출력
title('입력 신호 ');
subplot(212);
plot(x_filtered); % 필터링 된 신호
title('필터링된 신호 ');

figure
subplot(211);
plot(s);          %입력 신호 출력
title('입력 신호의 주파수 특성');
subplot(212);
plot(w,abs(fft(x_filtered))); % 필터링 된 신호
title('필터링된 신호의 주파수 특성 ');
```



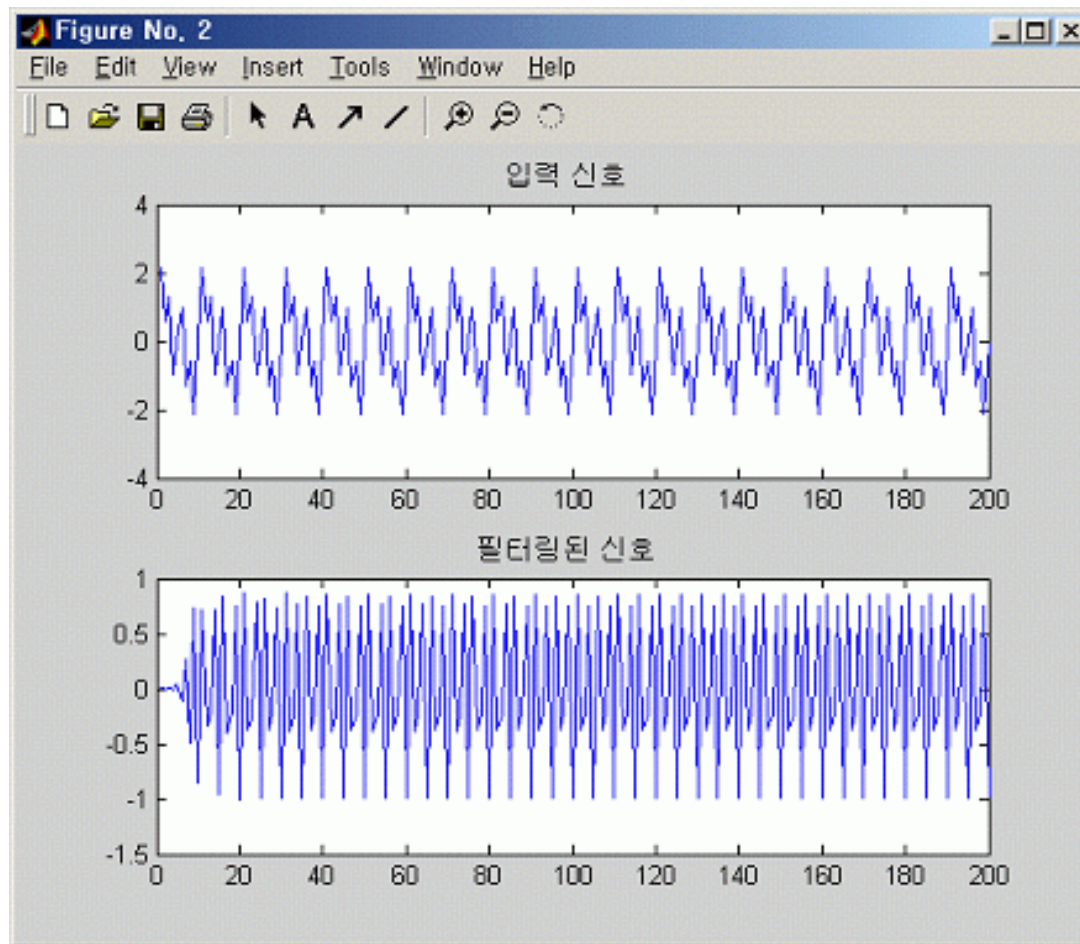
Matlab을 이용한 필터링 I



제 1형 Chebyshev 고역통과 필터의 주파수 특성



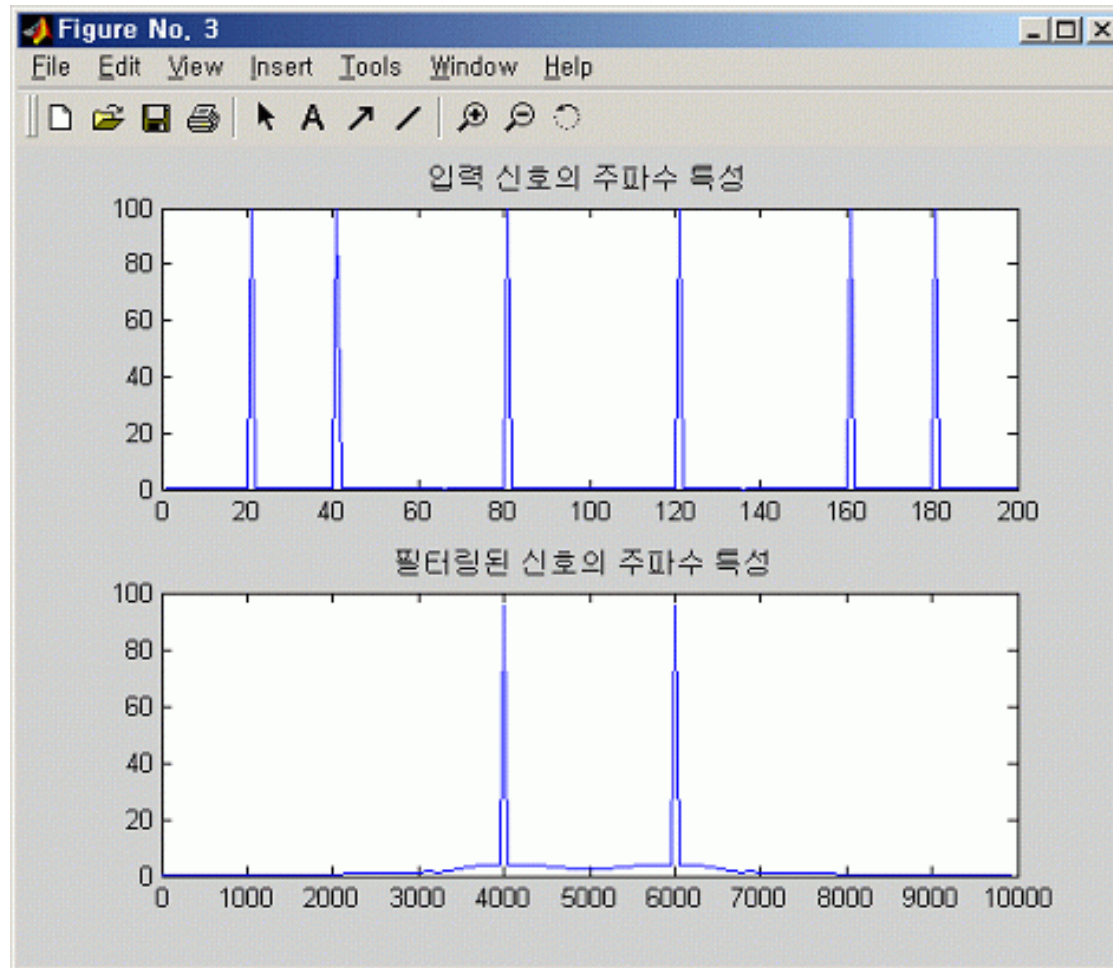
Matlab을 이용한 필터링 I



입력신호와 필터링된 신호



Matlab을 이용한 필터링 I



필터링 전과 후 신호의 주파수 스펙트럼 비교



Matlab을 이용한 필터링 II

- 창함수법을 이용한 FIR 필터를 설계하고 이를 적용해 필터링을 수행해보자.
- 입력 신호 : 1kHz, 2kHz, 4kHz 신호가 섞인 신호(신호 길이 : 500)
- 샘플링 주파수 : 10kHz
- 필터 조건 : 3kHz 차단 주파수의 저역 통과 필터(10차)



Matlab을 이용한 필터링 II

```
Fs=10000; % 샘플링 주파수
N=500;    % 샘플 개수
t=[1:N]/Fs; % 시간 축 값
w=[0:N-1]/N*Fs; % 주파수 축 값
x=sin(2*pi*1000*t)+sin(2*pi*2000*t)+sin(2*pi*4000*t); % 입력 신호
s=abs(fft(x)); % 입력 신호에 대한 주파수 특성
wn=3000/(Fs/2); % 차단 주파수 설정
```

```
fcoeff=fir1(10,wn); % 필터의 계수 구하기
[H W]=freqz(fcoeff,1,100); % 필터에 대한 주파수 특성
w=W/pi*(Fs/2);
x_filtered=filter(fcoeff,1,x); % 필터링 하기
```

```
% 그림 출력
figure
plot(w,abs(H)); % 필터의 주파수 특성
title('FIR 저역 통과 필터 주파수 특성');
```



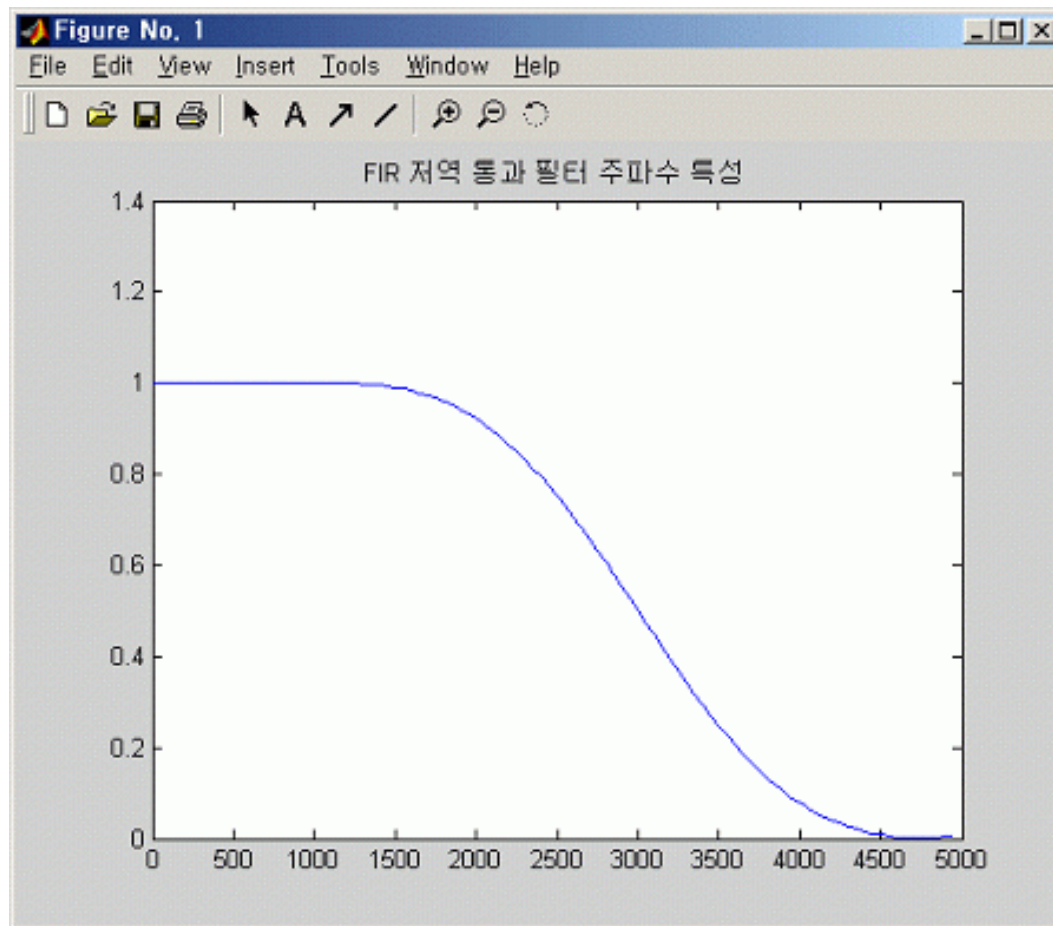

Matlab을 이용한 필터링 II

```
figure
subplot(211);
plot(x);           %입력 신호 출력
title('입력 신호 ');
subplot(212);
plot(x_filtered);  % 필터링 된 신호
title('필터링된 신호 ');
ww=[1:size(x,2)]/size(x,2)*Fs;
```

```
figure
subplot(211);
plot(ww,s);        %입력 신호 출력
title('입력 신호의 주파수 특성');
subplot(212);
plot(ww,abs(fft(x_filtered))); % 필터링 된 신호
title('필터링된 신호의 주파수 특성');
```



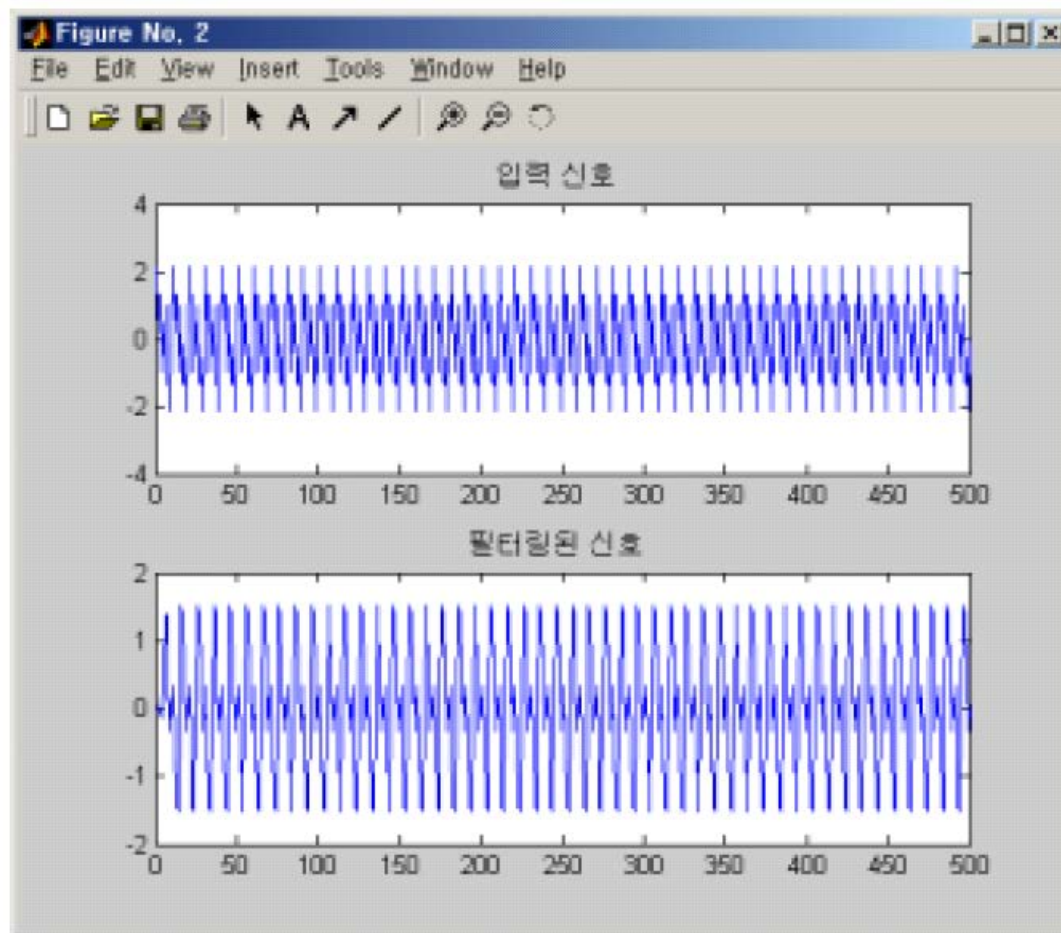
Matlab을 이용한 필터링 II



FIR 저역 통과 필터 주파수 특성



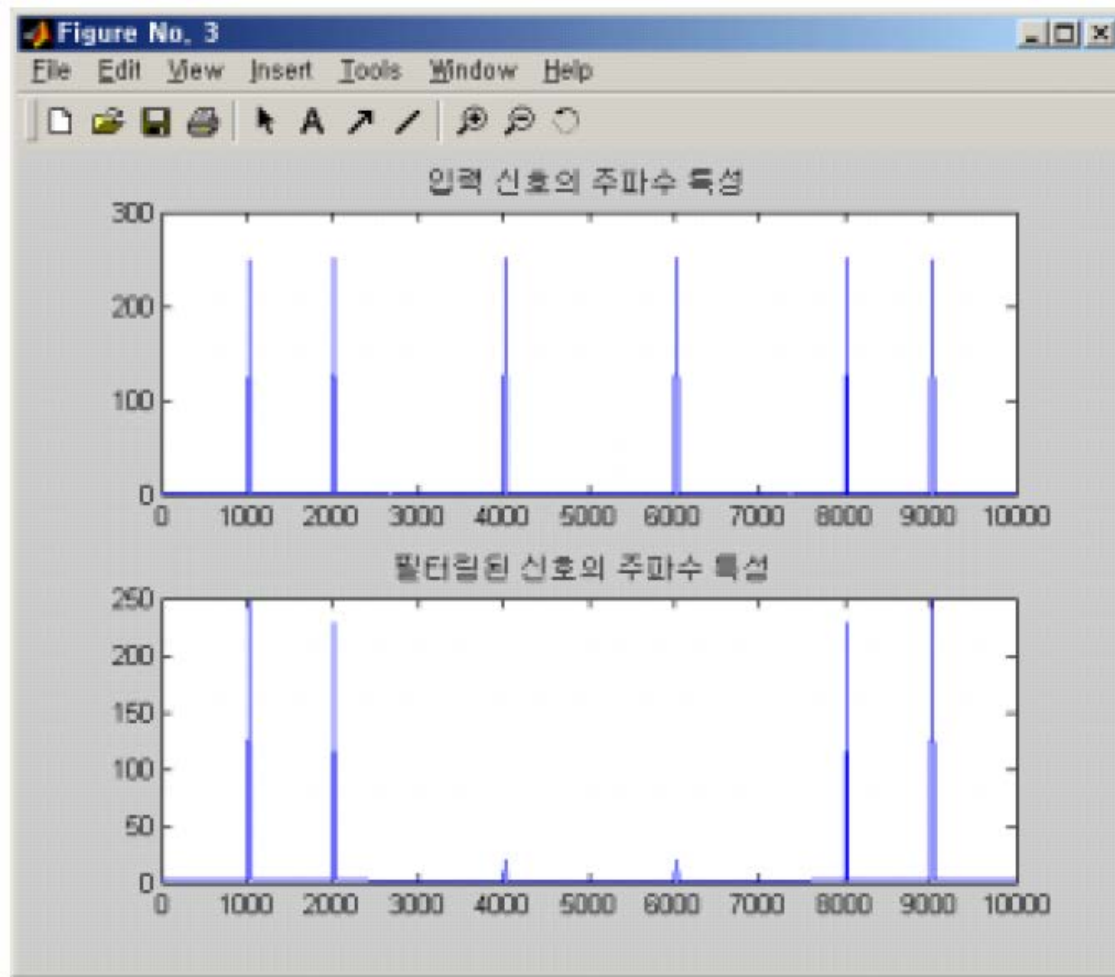
Matlab을 이용한 필터링 II



입력신호와 필터링된 신호



Matlab을 이용한 필터링 II



필터링 전과 후 신호의 주파수 스펙트럼 비교



Matlab을 이용한 필터링 III

- 실습 9의 나머지 세 Note에 대해 이름을 알아보도록 하자.

그림에서 주파수는 250Hz이상이므로 250Hz 차단 주파수를 갖는 고역 통과필터를 원신호에 적용하여 필터링을 수행하고 주파수 분석을 통하여 나머지 Note들에 대해 어떤 건반이 눌렸는지 확인하도록 하자.

1244.5	1318.5	
1108.7	1174.7	
	1046.5	
932.33	987.77	
830.61	880.00	
739.99	783.99	
	698.46	
622.25	659.26	
554.37	587.33	
	523.25	
466.16	493.88	
415.30	440.00	
369.99	392.00	
	349.23	
311.13	329.63	
277.18	293.66	
	261.63	

c#(도#)

bb(시b)

f(파)

기본 a 음



Matlab을 이용한 필터링 III

```
fid=fopen('sound.raw','r');  
x=fread(fid,'int16');  
N=2048;  
Fs=8000;  
deltaf=1/N*Fs;  
freq=[0:N-1]/N*Fs; % 주파수 좌표값
```

```
t=[1:N]/Fs; % 시간 축 값  
w=[0:N-1]/N*Fs; % 주파수 축 값
```

```
s=abs(fft(x)); % 입력 신호에 대한 주파수 특성
```

```
%----- 필터 설계 -----
```

```
ws=200/(Fs/2);  
wp=250/(Fs/2);  
Rp=0.1;  
Rs=60;  
[Nth, Wn]=cheb1ord(wp,ws,Rp,Rs); % chebyshev 필터의 차수와 차단 주파수 설정
```



Matlab을 이용한 필터링 III

```
[B, A]=cheby1(Nth,Rp,Wn,'high'); % 필터 설계  
H=freqz(B,A,w,Fs); % 주파수 특성
```

```
x_filtered=filter(B,A,x);% 필터링 하기
```

```
% 그림 출력
```

```
figure
```

```
plot(w,abs(H)); % 필터의 주파수 특성
```

```
title('제 1형 Chebyshev 고역 통과 필터의 주파수 특성');
```

```
figure
```

```
subplot(211);
```

```
plot(x); %입력 신호 출력
```

```
title('입력 신호 ');
```

```
subplot(212);
```

```
plot(x_filtered); % 필터링 된 신호
```

```
title('필터링된 신호 ');
```



Matlab을 이용한 필터링 III

```
figure
subplot(211);
plot(s);          %입력 신호 출력
title('입력 신호의 주파수 특성');
subplot(212);
plot(abs(fft(x_filtered))); % 필터링 된 신호
title('필터링된 신호의 주파수 특성');
```

```
note1=x_filtered(2760+500:2760+500+N-1); % Note1의 데이터
note2=x_filtered(6070+500:6070+500+N-1); % Note2의 데이터
note3=x_filtered(11001+500:11001+500+N-1); % Note3의 데이터
note4=x_filtered(14392+500:14392+500+N-1); % Note3의 데이터
note5=x_filtered(21152+500:21152+500+N-1); % Note3의 데이터
note6=x_filtered(24592+500:24592+500+N-1); % Note3의 데이터
```




Matlab을 이용한 필터링 III

```
X1=abs(fft(note1));  
X2=abs(fft(note2));  
X3=abs(fft(note3));  
X4=abs(fft(note4));  
X5=abs(fft(note5));  
X6=abs(fft(note6));
```

```
[Y1, I1]=max(X1(1:N/2));  
[Y2, I2]=max(X2(1:N/2));  
[Y3, I3]=max(X3(1:N/2));  
[Y4, I4]=max(X4(1:N/2));  
[Y5, I5]=max(X5(1:N/2));  
[Y6, I6]=max(X6(1:N/2));
```

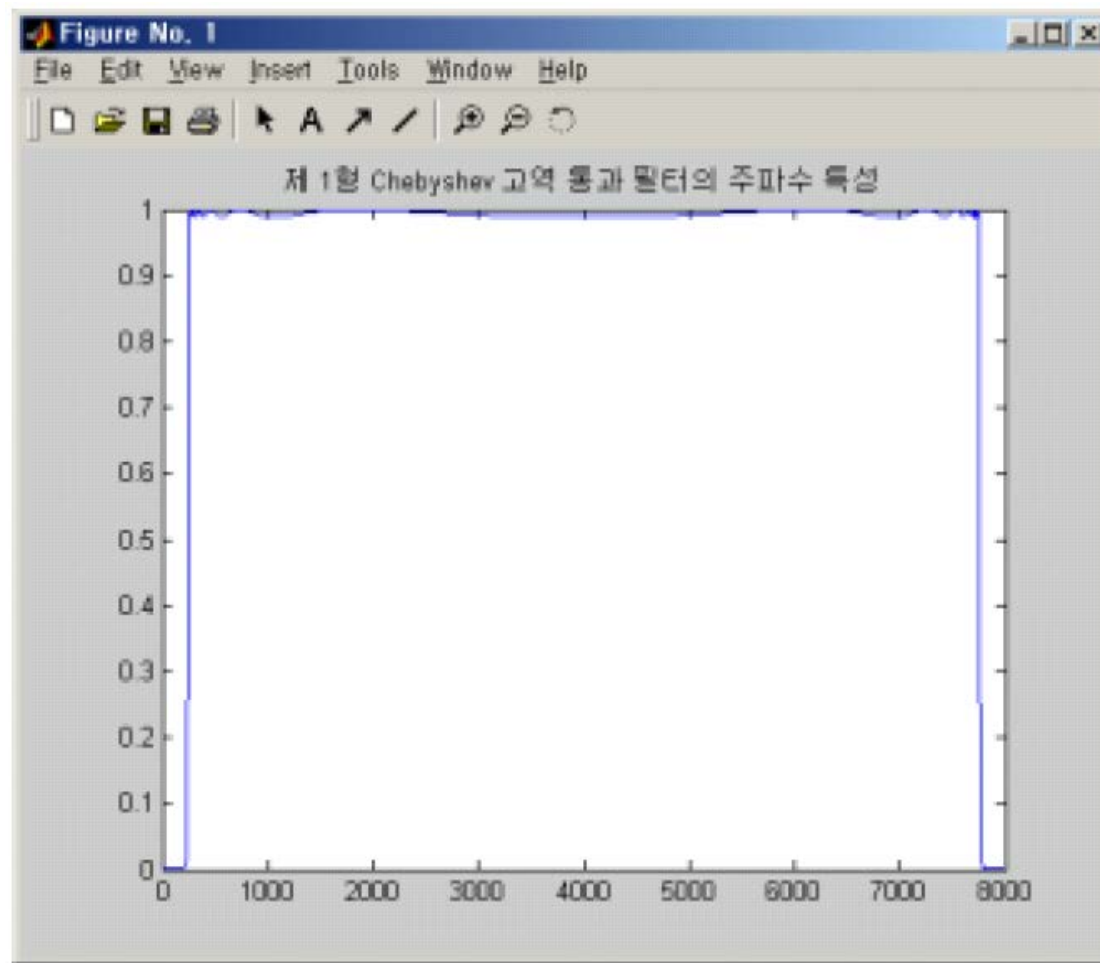


Matlab을 이용한 필터링 III

```
fprintf(1,'\\n첫번째 Note : %.2f',(l1-1)*deltaf);  
fprintf(1,'\\n두번째 Note : %.2f',(l2-1)*deltaf);  
fprintf(1,'\\n세번째 Note : %.2f',(l3-1)*deltaf);  
fprintf(1,'\\n네번째 Note : %.2f',(l4-1)*deltaf);  
fprintf(1,'\\n다섯번째 Note : %.2f',(l5-1)*deltaf);  
fprintf(1,'\\n여섯번째 Note : %.2f',(l6-1)*deltaf);
```



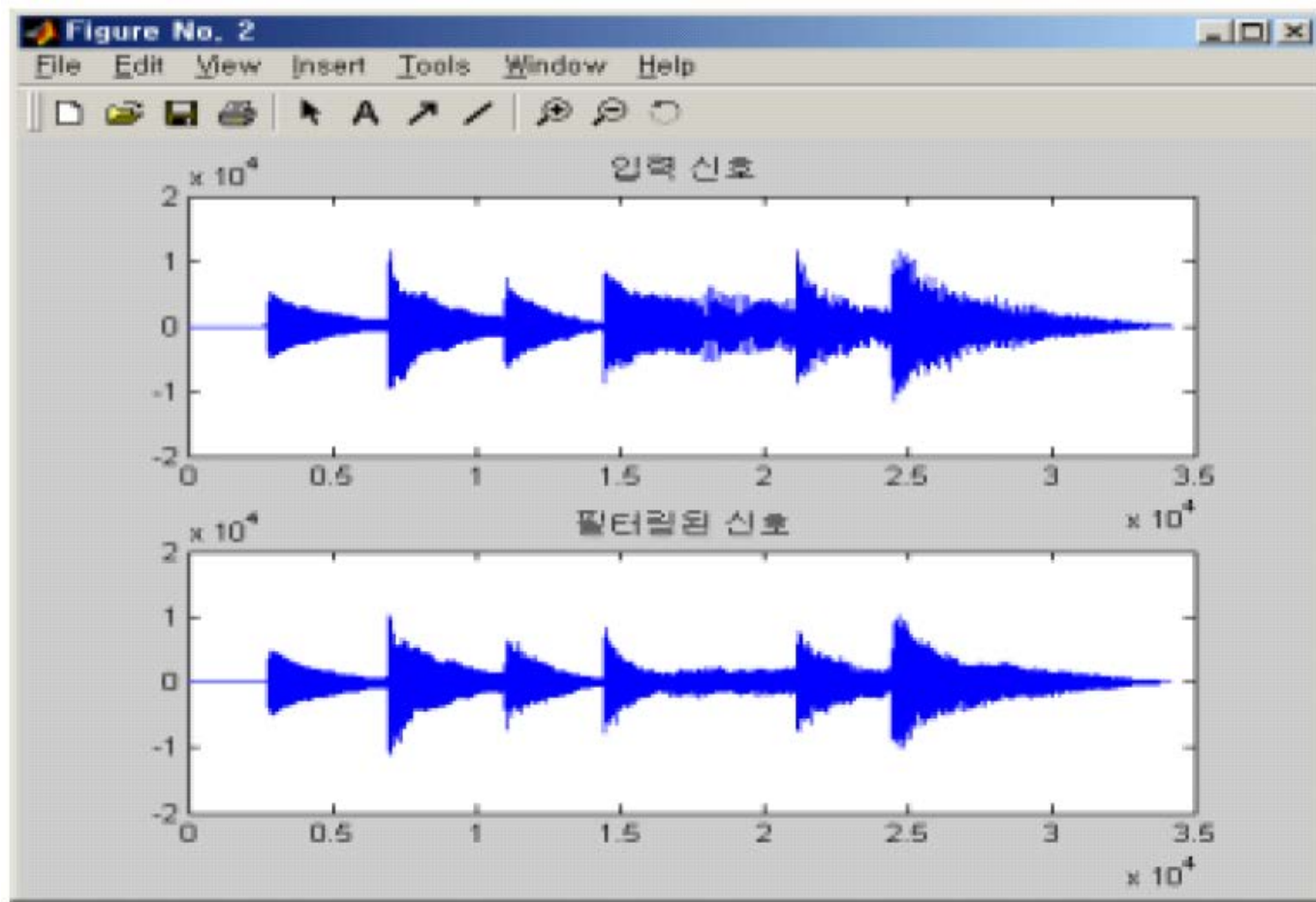
Matlab을 이용한 필터링 III



필터의 주파수 특성



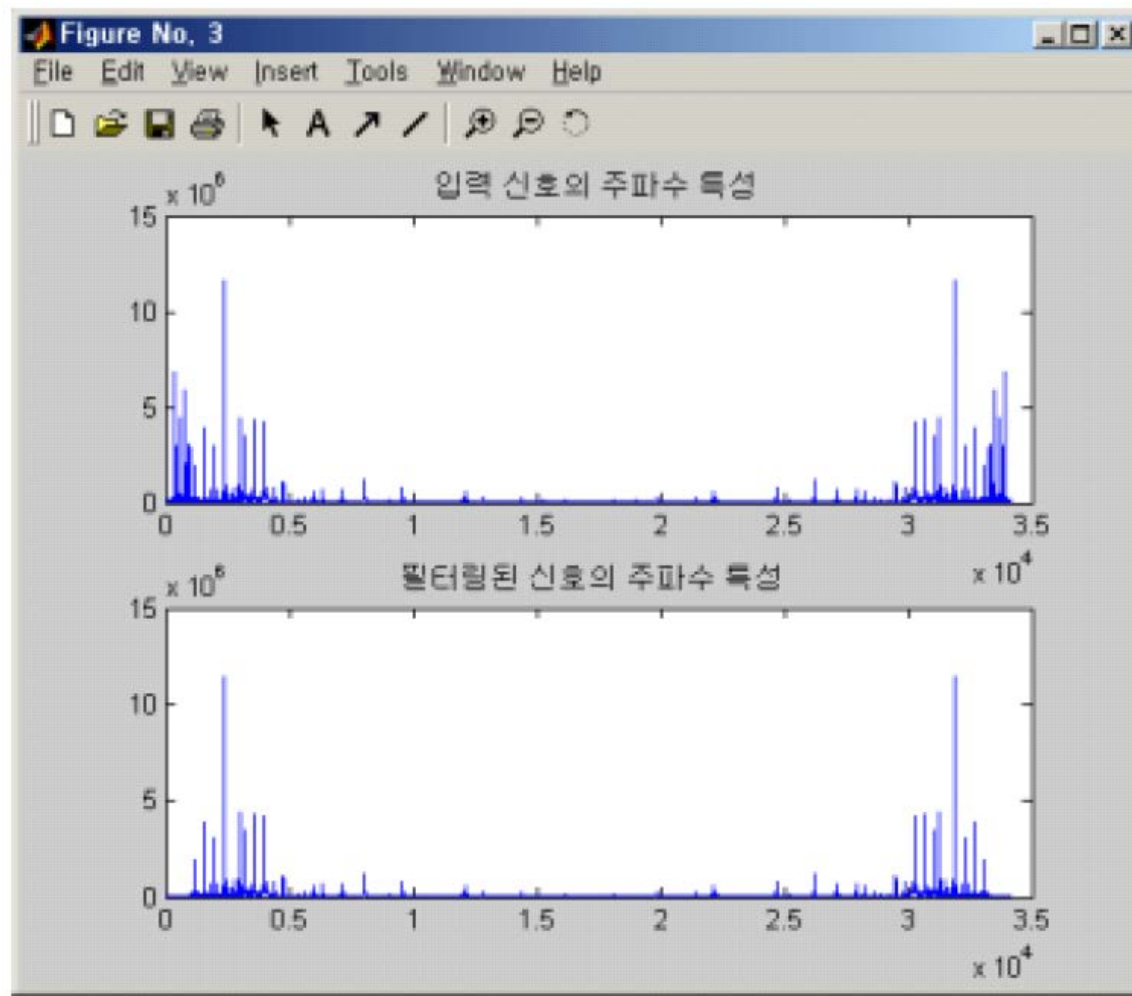
Matlab을 이용한 필터링 III



입력 신호와 필터링된 신호



Matlab을 이용한 필터링 III



입력신호와 필터링된 신호의 주파수 비교



Matlab을 이용한 필터링 III

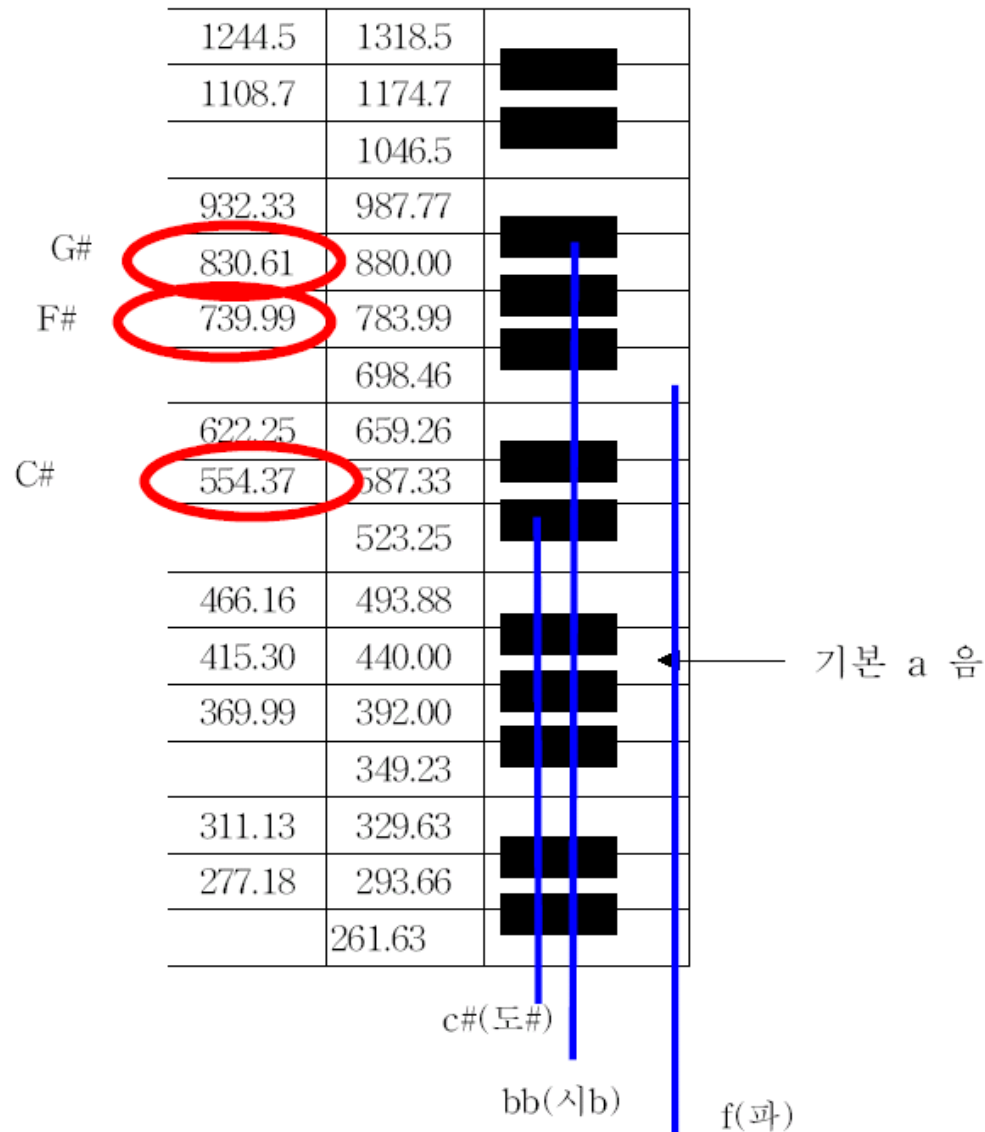
```
>>  
첫번째 Note : 554.69  
두번째 Note : 933.59  
세번째 Note : 699.22  
네번째 Note : 832.03  
다섯번째 Note : 742.19  
여섯번째 Note : 554.69  
>>
```

출력 결과



Matlab을 이용한 필터링 III

출력 결과에 대한
각 Note





Matlab을 이용한 필터링 III

각 Note 별 이름

Note	주파수 값	Note 이름
첫 번째 Note	554.69	C#
두 번째 Note	933.59	A# 또는 Bb
세 번째 Note	699.22	F
네 번째 Note	832.03	G#
다섯 번째 Note	742.19	F#
여섯 번째 Note	554.69	C#



평가하기

1. 아래와 같은 Matlab을 이용한 필터링 II 에서 사용한 창은 무엇인가? 또한 다른 창을 사용하기 위해서는 어떤 부분을 수정하여야 하는가?

```
Fs=10000; % 샘플링 주파수
N=500;    % 샘플 개수
t=[1:N]/Fs; % 시간 축 값
w=[0:N-1]/N*Fs; % 주파수 축 값
x=sin(2*pi*1000*t)+sin(2*pi*2000*t)+sin(2*pi*4000*t); % 입력 신호
s=abs(fft(x)); % 입력 신호에 대한 주파수 특성
wn=3000/(Fs/2); % 차단 주파수 설정

fcoeff=fir1(10,wn); % 필터의 계수 구하기
[H W]=freqz(fcoeff,1,100); % 필터에 대한 주파수 특성
w=W/pi*(Fs/2);
x_filtered=filter(fcoeff,1,x);% 필터링 하기

% 그림 출력
figure
plot(w,abs(H)); % 필터의 주파수 특성
title('FIR 저역 통과 필터 주파수 특성');
```



평가하기

1. 해밍 창

```
Fs=10000; % 샘플링 주파수
N=500;    % 샘플 개수
t=[1:N]/Fs; % 시간 축 값
w=[0:N-1]/N*Fs; % 주파수 축 값
x=sin(2*pi*1000*t)+sin(2*pi*2000*t)+sin(2*pi*4000*t); % 입력 신호
s=abs(fft(x)); % 입력 신호에 대한 주파수 특성
wn=3000/(Fs/2); % 차단 주파수 설정
```

```
fcoeff=fir1(10,wn); % 필터의 계수 구하기
```

```
[H W]=freqz(fcoeff,1,100); % 필터에 대한 주파수 특성
w=W/pi*(Fs/2);
x_filtered=filter(fcoeff,1,x); % 필터링 하기
```

```
% 그림 출력
figure
plot(w,abs(H)); % 필터의 주파수 특성
title('FIR 저역 통과 필터 주파수 특성');
```



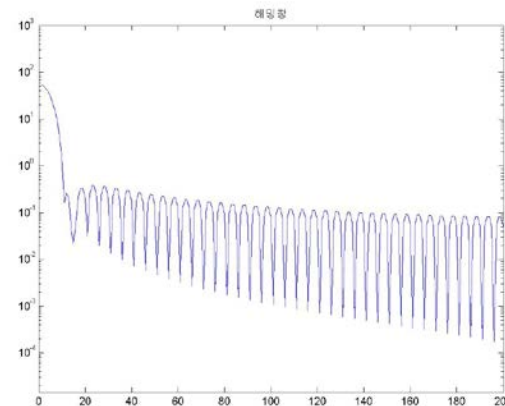
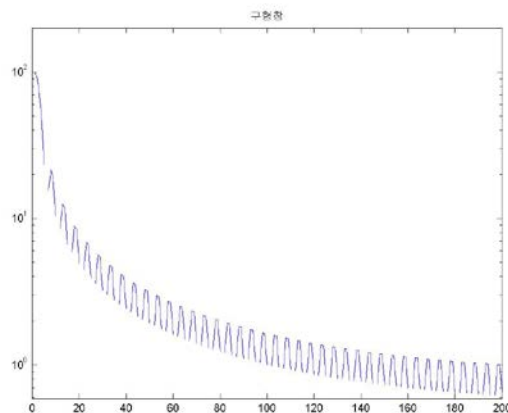
평가하기

1. 입출력 관계가 다음과 같을 때 시스템 각각에 대해 FIR 시스템인지 IIR 시스템인지 판별하시오.

System 1 : $y[n] = 0.5x[n] + 0.27x[n-1] + 0.77x[n-2]$

System 2 : $y[n] = 0.45x[n] + 0.5x[n-1] + 0.45x[n-2] + 0.53y[n-1] - 0.46y[n-2]$

2. 다음 그림은 구형창과 해밍창의 주파수 응답을 보여주고 있다. 각각을 비교 설명하시오.





평가하기

1. System 1 : FIR 시스템, System 2 : IIR 시스템
2. 구형차과 해밍창은 아래 그림의 A값은 주엽의 대역폭이며 대역폭이 크면 완만한 특성을 갖게 된다. 즉, 원 신호를 더 잘 표시할 수 있는 측면에서는 구형창이 더 좋다고 볼 수 있고 B는 주파수 누설에 대한 특성으로 작은 값을 갖는 해밍창이 성능이 우수함을 알 수 있다.

