

DENEY 4: Sayısal Filtreler

I. AMAÇ

Bu deneyin amacı sonlu dürtü yanıtı (FIR) ve sonsuz dürtü yanıtı (IIR) sayısal filtrelerin tanıtılması ve incelenmesidir.

II. ÖN HAZIRLIK

- 1) FIR ve IIR filtreleri kısaca açıklayınız ve farklarını belirtiniz.
- 2) Gibb's fenomeni (olayı), pencereleme (windowing) kavramlarını açıklayınız.
- 4) Attenuation (bastırma), kesim frekansı, filtre derecesi kavramlarını kısaca açıklayınız.
- 5) `butter`, `buttord`, `freqz`, `filter`, `fdesign` komutlarını açıklayınız ve örnek veriniz.
- 6) MATLAB `fdatool` hakkında kısaca bilgi veriniz. Burada yer alan “*Full View Analysis*” penceresini açarak ekran görüntüsü alınız.

III. ÖN BİLGİ

Girişi $x(n)$ çıkışı $y(n)$ ile gösterilen sayısal bir sistem aşağıdaki gibi bir **fark denklemi** ile ifade edilmektedir.

$$y(n) = b_0x(n) + b_1x(n-1) + \dots + b_Mx(n-M) - a_1y(n-1) \dots a_Ny(n-N) \quad (1)$$

Yukarıda belirtilen sayısal sistem bir filtreyi de ifade edebilir. Herhangi bir $x(n)$ girişine karşılık gelen çıkış değerleri; başlangıç koşullarının ($y(-1)$, $y(-2)$, $x(-1)$, $x(-2)$ gibi) bilinmesi durumunda denklem (1) yardımı ile elde edilebilir. Matlab' deki *filter* komutu yardımı ile sayısal bir filtrenin çıkışı elde edilebilir.

$A = [1 \ a_1 \ a_2 \ \dots \ a_n]$ ve $B = [b_0 \ b_1 \ \dots \ b_m]$ olmak üzere;

$$\mathbf{Y} = \text{filter}(\mathbf{B}, \mathbf{A}, \mathbf{x}) \quad (2)$$

olur.

Başlangıç koşulları 0 kabul edilir ve denklem (1)'in Z dönüşümü alınır **transfer fonksiyonu** $H(z)$ aşağıdaki gibi elde edilir.

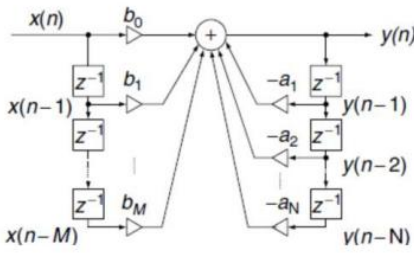
$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_M z^{-M}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_N z^{-N}} = \frac{B(z)}{A(z)} \quad (3)$$

Matlab 'ta bulunan *freqz()* komutu ile filtrenin frekans cevabı elde edilebilir. Aşağıda *freqz()* komutunun kullanılışı gösterilmektedir.

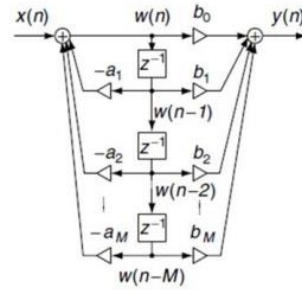
[h,w]=freqz(B,A,N)

Yukarıdaki denklemdeki N frekans cevabı elde edilirken kaç tane nokta kullanılacağını belirtmektedir yani değerler $\frac{\pi}{N}$ adımlarla elde edilmektedir.

Sayısal bir filtre b ve a katsayılarının bilinmesi durumunda gerçekleştirilebilir. Yani filtreyi sayısal olarak gerçekleştirebilmek için b ve a katsayılarını bulmak gerekir. Katsayıları elde edilen bir sayısal filtre Direct Form-I, Direct Form-II, seri ve paralel olmak üzere 4 farklı şekilde gerçekleştirilebilir. Aşağıda bu elde etme yöntemlerine ilişkin blok şemalar gösterilmektedir.

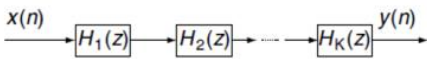


Şekil 1. Direct Form-I

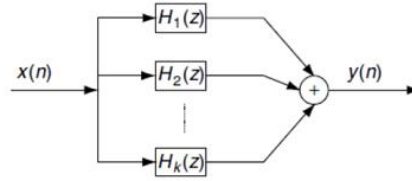


Şekil 2. Direct Form -II

Not: $w(n) = x(n) - a_1 w(n-1) - \dots - a_M w(n-M)$



Şekil 3. Seri



Şekil 4. Paralel

- FIR Filtre

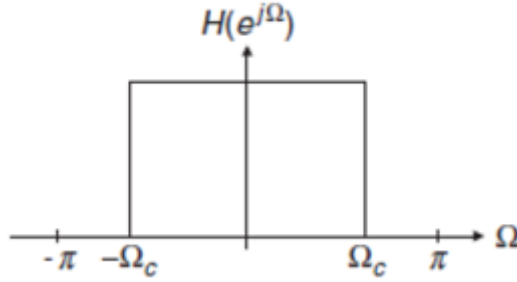
Bir FIR filtrenin birim darbe cevabı sonlu sayıda örnek içerir ve aşağıdaki **fark denklemi** ile tanımlanır.

$$y(n) = \sum_{i=0}^K b_i x(n-i) = b_0 x(n) + b_1 x(n-1) + \dots + b_K x(n-K) \quad (7)$$

Denklem (7) 'nin Z dönüşümü alındığında aşağıdaki ifade elde edilir.

$$Y(z) = b_0X(z) + b_1z^{-1}X(z) + \dots + b_Kz^{-K}X(z) \Rightarrow H(z) = b_0 + b_1z^{-1} + \dots + b_Kz^{-K} \quad !!!!$$

Aşağıda ideal alçak geçiren filtrenin frekans cevabı gösterilmektedir. Bilindiği gibi transfer fonksiyonunda $z = e^{j\Omega}$ yazılırsa $H(e^{j\Omega})$ elde edilir.



Şekil 5. İdeal alçak geçiren filtrenin frekans cevabı

İdeal alçak geçiren filtrenin frekans cevabı yukarıdaki şekilden de görüldüğü gibi $0 \leq \Omega \leq \Omega_c$ için 1 diğer yerlerde 0'dır ($\Omega_c \leq |\Omega| < \pi$).

$$H(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} c_n e^{-jw_0 n \Omega} \text{ ifadesinde } w_0 = \frac{2\pi}{T_0} = \frac{2\pi}{2\pi} = 1 \text{ 'dir.}$$

Fourier serisi katsayısı,

$$c_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\Omega}) e^{jn\Omega} d\Omega = \frac{1}{2\pi} \int_{-\Omega_c}^{\Omega_c} 1 d\Omega \text{ biçiminde elde edilir.}$$

İdeal alçak geçiren filtrenin frekans karakteristiği kullanılırsa;

$$c_0 = \frac{1}{2\pi} \int_{-\pi}^{\pi} H(e^{j\Omega}) d\Omega = \frac{1}{2\pi} \int_{-\Omega_c}^{\Omega_c} 1 d\Omega = \frac{\Omega_c}{\pi} \text{ ve } c_n = \frac{1}{2\pi} \int_{-\Omega_c}^{\Omega_c} (e^{jn\Omega}) d\Omega = \frac{\sin(\Omega_c n)}{\pi n}; n \neq 0$$

olur.

Sonuç olarak **ideal alçak geçiren filtre** için

$$H(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} c_n e^{-jw_0 n \Omega} \Rightarrow H(z) = \sum_{n=-\infty}^{\infty} c_n z^{-n}; c_0 = \frac{\Omega_c}{\pi}; c_n = \frac{\sin(\Omega_c n)}{\pi n} \quad (9)$$

biçiminde olur.

Daha önce bahsedildiği gibi FIR filtrenin darbe cevabı sonlu sayıda bileşen içerir. Dolayısı ile c_n katsayılarının baskın olan M tanesi tasarımda kullanılmalıdır. Nedensel olmayan FIR filtre için $c_n = h(n)$ alınırsa;

$$H(z) = h(M)z^M + \dots + h(1)z^1 + h(0) + h(1)z^{-1} + \dots + h(M)z^{-M} \quad (10)$$

elde edilir.

Filtrenin **nedensel olması için** darbe cevabı M örnek kadar geciktirilir ve aşağıda transfer fonksiyonu verilen $2M+1$ dereceli filtre elde edilir.

$$H(z) = b_0 + b_1z^{-1} + \dots + b_{2M}z^{-2M} \text{ ve } b_n = h(n - M), n=0,1, \dots, 2M \quad (11)$$

Anlatılan yöntemle yüksek geçiren, bant durduran, bant geçiren gibi diğer FIR filtre çeşitleri de tasarlanabilir. Ancak bu işlemlerin her defasında tekrarlanmasına gerek yoktur. İzlenecek olan adımlar belli olduğundan Matlab gibi programlarla filtre tasarımı gerçekleştirilebilir.

Gibbs olayı : Fourier katsayılarının sonsuz tanesinin de alınamamasından dolayı işaretin Fourier dönüşümünde oluşan dalgalanmalar.

Pencereleme : Fourier dönüşümü alınacak olan işaretin başında ve sonundaki süreksizliklerin bastırılması için kullanılan yöntem.

Bu sorunun etkisini azaltmak için pencere (window) fonksiyonu tasarımlarda kullanılabilir ($h_w(n) = h(n)w(n)$). Öte yandan filtre tasarımı Fourier dönüşümü yönteminden farklı yöntemler de kullanılmaktadır.

Filtre tasarımı kullanılabilen yöntemlerden biri de frekansta örnekleme yöntemidir. Frekansta örnekleme yönteminde istenilen filtrenin frekans cevabından faydalanarak filtre tasarlanmaktadır. Bu sebeple tasarımı daha esnektir. Nedensel bir FIR filtrenin birim darbe cevabının $h(n)$, $n = 0,1,\dots,N-1$ olduğunu varsayalım. Birim darbe cevabı $h(n)$ 'e denk gelen ayrık Fourier dönüşümü katsayıları da $H(k)$, $k = 0,1,\dots,N-1$ gösteriliyor olsun. Tasarlanmak istenen bir filtrenin frekans cevabı örneklendiğinde $H(k)$ elde edilmiş olacaktır. Ayrık ters Fourier dönüşümü ile filtre katsayıları aşağıdaki gibi elde edilebilir.

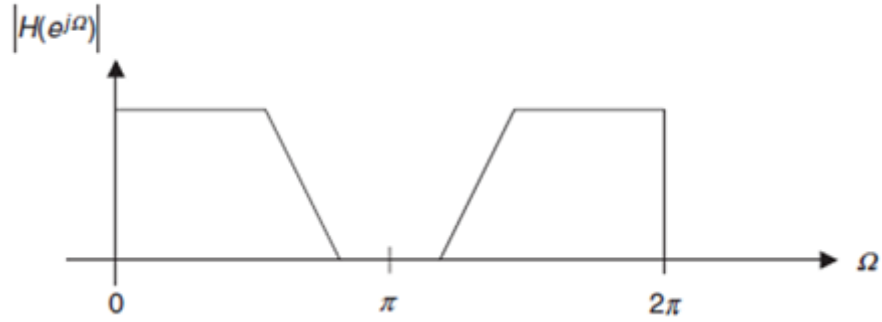
$$h(n) = \frac{1}{N} \sum_{k=0}^{N-1} H(k) W_N^{-nk}, n = 0,1, \dots, N-1 \quad (12)$$

$$W_N = e^{-\frac{j2\pi}{N}} = \cos\left(\frac{2\pi}{N}\right) - j\sin\left(\frac{2\pi}{N}\right) \quad (13)$$

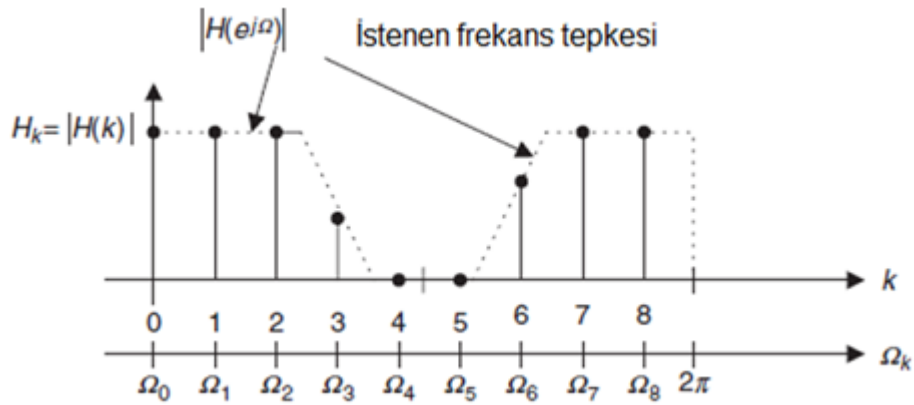
Filtrenin derecesi $N = 2M+1$ olarak alınırsa denklem (12) aşağıdaki gibi tekrar yazılabilir.

$$h(n) = \frac{1}{2M+1} \left\{ H_0 + 2 \sum_{k=0}^M H_k \cos\left(\frac{2\pi k(n-M)}{2M+1}\right) \right\}, n = 0,1, \dots, 2M \quad (14)$$

Yukarıdaki denklemde tasarlanmak istenen filtrenin frekans cevabının örneklenmesi sonucu elde edilen örnek değerlerini göstermektedir.



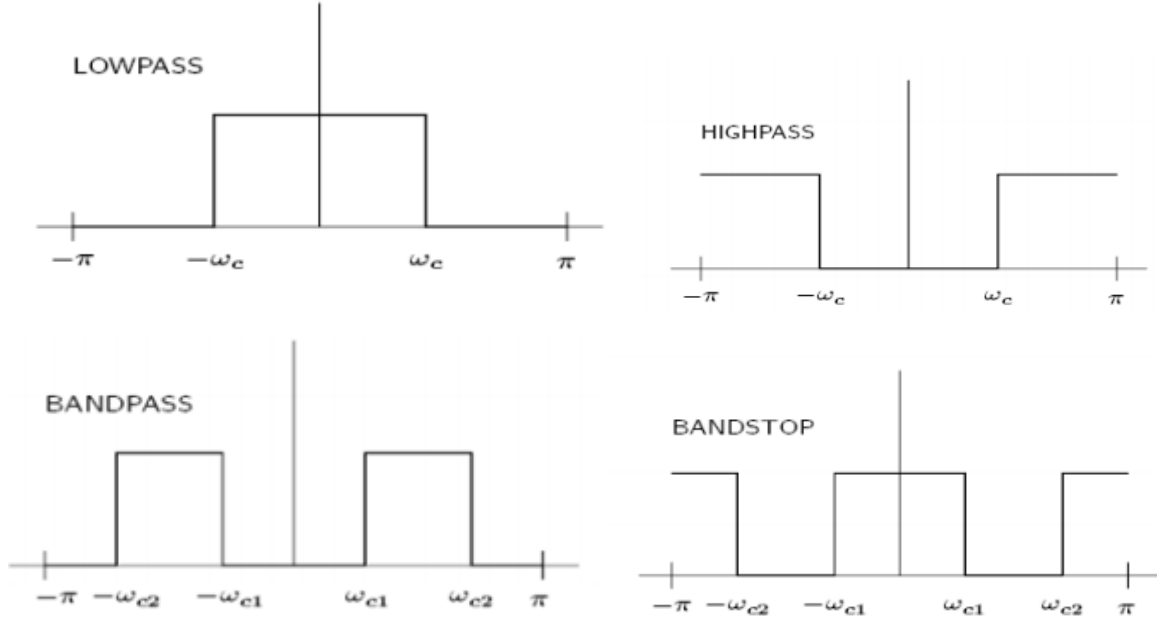
Şekil 7. Tasarlanmak istenmek istenen filtrenin frekans cevabı



Şekil 8. Örneklenmiş frekans cevabı

Şekil (8)'den de görüldüğü gibi tasarlanmak istenen filtrenin frekans cevabı $\Omega_k = \frac{2\pi k}{2M+1}$ aralıklarla alınan örneklerle elde edilmektedir. Filtre katsayılarının ilk yarısı elde edildikten sonrakiler $h(n) = h(2M - n), n = M + 1, \dots, 2M$ olduğundan hesaplama yapmadan elde edilebilir.

!!!! Aşağıdaki şekilde ideal filtre frekans cevapları gösterilmektedir. !!!!



FIR Filtrenin Avantajları :

- Her zaman doğrusal fazdadır.
- Kutupları olmadığı için her zaman kararlıdır.

- IIR Filtre

IIR filtrenin girişi ve çıkışı arasındaki ilişki denklem (1)'deki gibi, transfer fonksiyonu da denklem (3)'teki gibidir. IIR filtrenin birim darbe cevabı sonsuz sayıda bileşen içerir. Öte yandan IIR filtrenin transfer fonksiyonu, $H(z)$ 'nin paydası 1 olmadığından tasarımında **kutuplar göz önüne alınmalı ve kararlı olabilmesi için kutupların birim çember içinde olması sağlanmalıdır**. FIR filtreye göre daha az sayıda hesaplama gerektirir ancak FIR filtrenin kolay sağlayabildiği doğrusal fazı elde etmesi zordur. Bu sebeple IIR filtre daha az işlem ile gerçekleştirilebilecek bir filtreye ihtiyaç duyulduğunda ancak, doğrusal fazın önemli olmadığı durumlarda kullanılır. IIR filtre tasarlamak için analog filtrelerin transfer fonksiyonlarına dönüşüm uygulanabilir. IIR filtrelere örnek olarak Butterworth, Chebyshev Tip 1, Chebyshev Tip 2, Eliptik ve Bessel filtreler verilebilir.

IIR Filtrenin Avantajları :

- FIR filtrenin çok fazla sayıda yaptığı filtrelemeyi daha az sayıda katsayı ile yapar.
- İşlem yükü daha azdır.
- Daha kullanışlı filtreler tasarlanabilir.

- MATLAB Filtre Kodları

fir1 MATLAB komutu, pencere işlevi kullanım yöntemi ile lineer fazlı sayısal süzgeç tasarımı sağlar. Bu komut ile alçak geçiren, yüksek geçiren, band geçiren ve band söndüren süzgeç tasarlanabilir. En basit halinde süzgecin geçirme bandının merkez frekansındaki genlik yanıtı 0 dB'dir.

Komutun kullanım örnekleri:

→ $b = \text{fir1}(n, \omega_n)$; n-inci dereceden alçak geçiren sonlu dürtü yanıtı süzgecin $n + 1$ adet katsayısını döndürür. *fir1* komutu bu örnekte Hamming penceresi kullanır ve $0 - \pi$ arasında değerler alan ω_n kesim frekansına sahiptir.

→ $b = \text{fir1}(n, \omega_n)$; komutundaki ω_n iki elemanlı bir vektör ise, *fir1* komutu $\omega_1 < \omega < \omega_2$

Arası geçirme bandı olan band geçiren süzgeç katsayılarını döndürür.

→ $b = \text{fir1}(n, \omega_n, 'ftype')$; komutu süzgeç tipini tanımlar:

- 'high'; kesim frekansı ω_n olan yüksek geçiren süzgeç için

- 'stop'; band söndüren süzgeç için ($\omega_n = [\omega_1, \omega_2]$ ise band söndürme frekans aralığıdır).

→ $b = \text{fir1}(n, \omega_n, window)$; komutu istenen pencere fonksiyonunu kullanarak alçak geçiren süzgeç katsayılarını döndürür.

→ $b = \text{fir1}(n, \omega_n, 'ftype', window)$; komutu istenen pencere fonksiyonunu ve süzgeç türünü kabul eder.

$[Z, P, K] = \text{buttap}(N)$

buttap(N) işlevi N. dereceden Butterworth analog örnek süzgecin sıfırlarını (Z), kutuplarını (P) ve K kazanç değerini hesaplar. Sonuçta elde edilen süzgecin birim çemberin sol yarı düzleminde N tane kutbu olup, sıfırı yoktur ve K kazanç değeri de 1'dir. $[b, a] = \text{zp2tf}(z, p, k)$ komutu sıfır-kutup değerlerini aktarım işlevine dönüştürür. *b* ve *a* pay ve paydadaki aktarım işlevi katsayılarıdır. Butterworth Süzgecinde *b* değeri 1'dir.

$[N, Wn] = \text{buttord}(\omega_p, \omega_s, Rp, Rs)$

buttord komutu geçirme bandında R_p dB'den az ve söndürme bandında R_s dB'den fazla kadar zayıflatma yapan en düşük sayısal Butterworth süzgeci derecesini verir. ω_p ve ω_s değeri 0 ile 1 arasında olan normalize edilmiş geçirme ve söndürme bantlarının sınır frekans değerleridir. Burada 1 değeri π radyan örnek sonucuna karşılık gelmektedir.

$[N, Wn] = \text{buttord}(\omega_p, \omega_s, Rp, Rs, 's')$ komutu ω_p ve ω_s [radyan/saniye] biriminde olmak üzere, analog bir süzgeç için hesaplama yapar.

$[b, a] = \text{butter}(N, Wn, 's')$ komutu N. derece alçak geçiren W_n [rad/sn] açısal kesim frekanslı bir analog Butterworth süzgeç tasarlar. $W_n = [\omega_{nl} \ \omega_{nu}]$ ise *butter* komutu 2n. derece band geçiren ve geçirme bandı $\omega_{nl} \ \omega_{nu}$ sınır frekansları olan süzgeç tasarlar. Benzer şekilde *cheby1*, *cheby2* ve *ellip* komutları da vardır.

$[b, a] = \text{butter}(n, W_n)$ *butter* komutu ile ($0 < W_n < 1.0$) kesim frekanslı N. dereceden N+1 uzunluklu sayısal Butterworth süzgeç katsayıları elde edilir.

IV. DENEYİN YAPILIŞI

!!! Deneyde yazılacak tüm kodların MATLAB'ın editör penceresinde düzenli bir şekilde yazılması ve kayıt edilmesi beklenmektedir. Oluşturulan tüm grafikleri için **açıklayıcı başlıklar** yazılması gerekmektedir.

1) a. $x[n] = u[n]$, başlangıç koşulları $y[-1] = 2$, $y[-2] = 1$, ve $x[-1] = x[-2] = 0$ olarak verilen, $y[n] - 0.6y[n-1] + 0.08y[n-2] = x[n-1]$ sistemin. $n = [0, 1, \dots, 10]$ için $y[n]$ yanıtını hesaplayınız ve çizdiriniz.

b. $y[-1] = y[-2] = 0$ yaparak sistem çıkışını (y_{out2}) *filter* komutunu kullanarak hesaplayınız ve **sonuçları subplot ile çizdiriniz**. (for ile yapılan filtrelemenin ters fazlısı çıkabilir (Negatif sonuç))

```
close all,clear all; clc;
a = [-0.6 0.08]; b = [0 1];
y0 = [1 2]; x0 = [0 0];

n = 0:10;
x = [x0 ones(1,11)];
y = [y0 zeros(1,11)];

M = length(b);
N = length(a);

for i=3:13
    y(i) = sum(b.*x(i-1:i-(M-1))) - sum(a.*y(i-1:-1:i-N));
end
y_out1 = y(3:13);
y_out2 = ..... %(filter komutu kullanılacak)
```

2)

$$a) H(z) = \frac{z}{z-0.5}$$

$$b) H(z) = 1 - 0.5z^{-1}$$

$$c) H(z) = \frac{0.5z^2 - 0.32}{z^2 - 0.5z + 0.25}$$

Yukarıda verilen transfer fonksiyonlarının b ve a katsayılarını belirleyiniz. Transfer fonksiyonları ile verilen filtrelerin genlik (dB) ve faz (derece) tepkilerini çizdiriniz.

Şıklarda verilen filtrelerin türünü belirleyiniz. (title(...))

```
% a sıklık için kod
b = [1]; a = [1 -0.5];
[h w] = freqz(b,a,1024);
figure,
subplot(211),plot(w/pi,20*log10(abs(h))),grid on, title(...) %filtre türü?
subplot(212),plot(w/pi,180/pi*angle(h)),grid on, title(...)
```


3) Aşağıda verilen frekans cevabına sahip alçak geçiren filtre tasarlanmıştır ($f_s = 1\text{kHz}$, $f_c = 62.5\text{ Hz}$, $w_c = 2\pi f_c/f_s = \pi/4$). Kodun ne iş yaptığını anlayarak **istenen satırlara kısa yorumlar yazınız !!!** (yorumların açıklayıcı olmaları gerekmektedir). Yazdığınız yorumları ve oluşan grafikleri ilgili araştırma görevlisine adım adım açıklayınız ve kodun genel olarak ne iş yaptığını anlatınız.

$$H_d(e^{jw}) = \begin{cases} 1 & -w_c \leq w \leq w_c \\ 0 & \text{diğer} \end{cases}$$

```
clear all,close all;clc
N = 50;
n=-N:N;

fs= 1e3; fc = 62.5; wc = 2*pi*fc/fs;
% wc=pi/8;

h = sin(wc*n)./(pi*n); % .....
h(N+1) = wc/pi;
figure
stem(n,h)
title(...)
xlabel('n'),axis([-2*N 2*N -1.5*wc/pi 1.5*wc/pi])
%%
w=linspace(-pi,pi,1024);
H = fftshift(fft(h,1024)); % .....
abs_H = abs(H); % ...
ang_H = unwrap(angle(H)); % ...

figure, plot(w,abs_H),grid on, title(...),
figure, plot(w,ang_H),grid on, title(...),
%%
top = 0;
for k=1:5
    top = top + 2*k*cos(k*pi/10*n); % ....
end

figure,
stem(n,top) , title(...),
%%
w=linspace(-pi,pi,1024);
T = fftshift(fft(top,1024));
abs_T = abs(T);

figure,
plot(w,abs_T),grid on,hold on, title(...), % ...
plot(w,max(abs_T)*abs_H,'r'),grid on, title(...),
%%
y = ifft(fft(top).*fft(h)); % ....
% y = conv(top,h);
```

```

Y = fftshift(fft(y,1024)); % .....
abs_Y = abs(Y);
figure, plot(w,abs_Y),grid on,hold on,
plot(w,max(abs_Y)*abs_H,'r'),grid on, title(...),

figure, plot(n,y) , title(...)

```

===== DENEY SONU =====

4) Aşağıda verilen kodda boş bırakılan kısımda *butter* komutu ile filtre katsayıları hesaplatarak gürültülü işareti *filter* komutu ile filtreleyiniz.

```

close all, clear all;clc,
fs = 1e4;
t = 0:1/fs:5;
sw = sin(2*pi*250*t);
v = 0.1*randn(size(sw)); % gürültü işareti
swn = sw + v;
%soundsc(swn,1e4)
figure,plot(t,sw), axis([0 0.04 -1.1 1.1])
figure,plot(t,swn), axis([0 0.04 -1.1 1.1])
figure,plot(linspace(-fs/2,fs/2,length(t)),fftshift(abs(fft(sw))))
figure,plot(linspace(-fs/2,fs/2,length(t)),fftshift(abs(fft(swn))))

[b a] = butter(2, 400/(fs/2));
figure, freqz(b,a)

y = filter ..... ??
figure, plot(t,y), axis([0 0.04 -1.1 1.1]),
figure,plot(linspace(-fs/2,fs/2,length(y)),.....)
% filtrelenmiş isaretin genlik cevabı çizdirilecek hint: fft, abs

```

5)

```

close all,clear all;clc,

wg=[0.25]; % Alt geciren
wd=[0.5];
%wg=[0.5] % ust geciren
%wd=[0.1]
%wg=[0.25 0.5]; % Band geciren
%wd=[0.1 0.7];
%wg=[0.25 0.5]; % Band Durduran
%wd=[0.1 0.7];
gddb=1;
sddb=40;
[N,Wn]=buttord(wg,wd,gddb,sddb);

```

```

...           % butterworth filtre katsayıları elde edilmesi(B,A) (butter
komutu ile)
fs=1000;
...           % katsayılar ile Hg frekans cevabı elde edilmesi (freqz
komutu ile )
Hg=20*log10(abs(H));
plot(W/pi,Hg)
grid on
AXIS([0 1 -40 5])
xlabel('W/pi');
ylabel('Kazanc(dB)');
title('IIR Butterworth Alt Geciren Suzgec')
figure, plot(abs(H));
grid on
xlabel('Hz');
ylabel('Kazanc');
AXIS([0 1000 0 1.2])
for i=1:(length(H));
    if abs(H(i))<0.01;
        H(i)=0;
    end
end
Ha=unwrap(angle(H));
figure, plot((Ha/pi)*180);
xlabel('Hz');
ylabel('Faz');
grid on

```

- Yukarıda verilen kodda “...” bırakılan yerleri *butter*, *freqz* komutları ile doldurarak matlab ortamında alçak geçiren filtre türü için çalıştırınız.
- Diğer kesim frekansları için deneyiniz.