

۱ پنجره کایزر

7.5) فیلتر خواسته شده باید دارای مشخصات زیر باشد:

$$\begin{aligned} |H(e^{j\omega})| &\leq 0.01 & 0 \leq |\omega| \leq 0.25\pi \\ 0.95 \leq |H(e^{j\omega})| &\leq 1.05 & 0.35\pi \leq |\omega| \leq 0.6\pi \\ |H(e^{j\omega})| &\leq 0.01 & 0.65\pi \leq |\omega| \leq \pi \end{aligned}$$

الف) برای طراحی فیلتر مورد نظر ابتدا باید مقدار δ را تعیین کرد. برای این کار ابتدا باید مقادیر δ_1 ، δ_2 و δ_3 را تعیین کرد. باید توجه داشت که مقدار δ در هر قسمت برابر ماکزیمم اختلاف مجاز از مقدار مجاز تقسیم بر مقدار ناپیوستگی در آن قسمت است. در نتیجه داریم:

$$\delta_1 = \frac{0.01}{1} = 0.01$$

$$\delta_2 = \frac{0.05}{1} = 0.05$$

$$\delta_3 = \frac{0.01}{1} = 0.01$$

حال باید مقدار δ را تعیین کنیم. مقدار δ برابر مینیمم مقادیر δ_1 ، δ_2 و δ_3 است. در نتیجه داریم:

$$\delta = \min(\delta_1, \delta_2, \delta_3) = \min(0.01, 0.05, 0.01) = 0.01$$

سپس باید مقدار A را محاسبه کنیم:

$$A = -20 * \log_{10}(\delta) = -20 * \log_{10}(0.01) = 40$$

سپس باید مقدار β را محاسبه کنیم:

$$\beta = \begin{cases} 0.1102(A - 8.7) & A > 50 \\ 0.5842(A - 21)^{0.4} + 0.07886(A - 21) & 21 \leq A \leq 50 \\ 0 & A < 21 \end{cases}$$

$$A = 40$$

$$\beta = 3.3953$$

سپس باید مقدار $\Delta\omega$ را محاسبه کنیم. برای این کار ابتدا باید مقادیر $\Delta\omega_1$ و $\Delta\omega_2$ را تعیین کرد.

$$\Delta\omega_1 = 0.1\pi$$

$$\Delta\omega_2 = 0.05\pi$$

حال باید مقدار $\Delta\omega$ را تعیین کنیم. مقدار $\Delta\omega$ برابر مینیمم مقادیر $\Delta\omega_1$ و $\Delta\omega_2$ است. در نتیجه داریم:

$$\Delta\omega = \min(\Delta\omega_1, \Delta\omega_2) = \min(0.1\pi, 0.05\pi) = 0.05\pi$$

سپس باید مقدار M را محاسبه کنیم:

$$M = \left\lceil \frac{A - 8}{2.285\Delta\omega} \right\rceil = \left\lceil \frac{40 - 8}{2.285 * 0.05\pi} \right\rceil = \lceil 89.154 \rceil = 90$$

(ب) میزان تاخیر این فیلتر برابر است با:

$$\frac{M}{2} = \frac{90}{2} = 45$$

(ج) برای بدست آوردن پاسخ ضربه ایده‌آل $h_d[n]$ ابتدا $H_d(e^{j\omega})$ را محاسبه می‌کنیم.

$$H_d(e^{j\omega}) = e^{-j45\omega} (u(\omega - 0.3\pi) - u(\omega - 0.625\pi)) \quad 0 \leq |\omega| \leq \pi$$

سپس با تبدیل فوریه معکوس گرفتن از $H_d(e^{j\omega})$ ، $h_d[n]$ را بدست می‌آوریم:

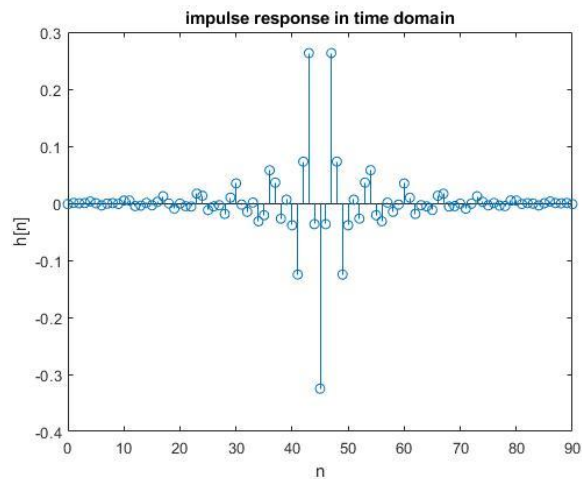
$$h_d[n] = -0.3 * \text{sinc}(0.3(n - 45)) + 0.625 * \text{sinc}(0.625(n - 45))$$

سپس با استفاده از دستور Kaiser ضرایب فیلتر Kaiser را بدست می‌آوریم.

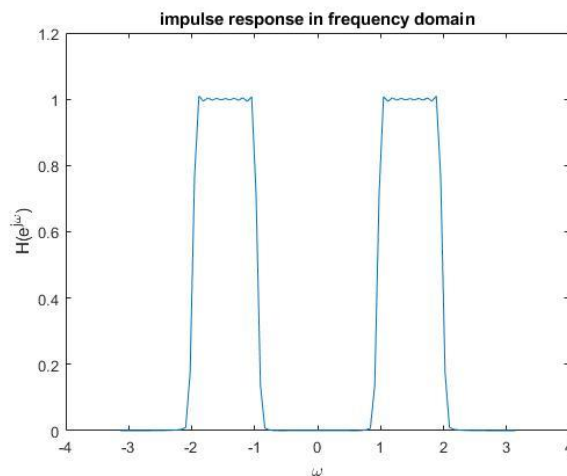
سپس با ضرب نقطه به نقطه پاسخ ضربه ایده‌آل شیفت‌یافته در فیلتر Kaiser، پاسخ ضربه فیلتر مورد نظر را محاسبه می‌کنیم.

سپس با تبدیل فوریه گرفتن از پاسخ ضربه فیلتر مورد نظر، پاسخ فرکانسی فیلتر را بدست می‌آوریم.

پاسخ ضربه فیلتر مورد نظر:



پاسخ فرکانسی فیلتر مورد نظر:



در انتها با استفاده از نمودار پاسخ فرکانسی فیلتر مورد نظر مقادیر Overshoot و Undershoot را در passband و stopband بدست می آوریم.

$$\text{Undershoot} = 0.0065$$

$$\text{Overshoot} = 0.009625$$

کد متلب :

```
delta1 = 0.01;
delta2 = 0.05;
delta3 = 0.01;
delta = min([delta1,delta2,delta3]);
A = -20*log10(delta);
if (A < 21)
    beta = 0;
elseif (A >= 21 && A <= 50)
    beta = 0.5842*(A-21)^0.4 + 0.07886*(A-21);
elseif (A > 50)
    beta = 0.1102*(A-8.7);
end
deltaw1 = 0.1*pi;
deltaw2 = 0.05*pi;
deltaw = min(deltaw1,deltaw2);
M = ceil((A-8)/(2.285*deltaw));
delay = M/2;
n = 0:M;
hd = sin(0.3*pi*(n-M/2))./(pi*(n-M/2)) - sin(0.625*pi*(n-M/2))./(pi*(n-M/2));
if M/2 == floor(M/2)
    hd(M/2+1) = 0.3 - 0.625;
end
KaiserFilter = kaiser(M+1,beta)';
h = hd.*KaiserFilter;
H = fftshift(fft(h,length(h)));
stem(n,h)
xlabel('n')
```

```

ylabel('h[n]')
title('impulse response in time domain')
figure
f = (0:length(H)-1)*2*pi/(length(H)-1) - pi;
plot(f,abs(H))
xlabel('\omega')
ylabel('H(e^{j\omega})')
title('impulse response in frequency domain')

```

7.6) فیلتر خواسته شده باید دارای مشخصات زیر باشد:

$$\begin{aligned}
 0.9 < |H(e^{j\omega})| < 1.1 & \quad 0 \leq |\omega| \leq 0.2\pi \\
 |H(e^{j\omega})| \leq 0.06 & \quad 0.3\pi \leq |\omega| \leq 0.475\pi \\
 1.9 < |H(e^{j\omega})| < 2.1 & \quad 0.525\pi \leq |\omega| \leq \pi
 \end{aligned}$$

الف) برای طراحی فیلتر مورد نظر ابتدا باید مقدار δ را تعیین کرد. برای این کار ابتدا باید مقادیر δ_1 ، δ_2 و δ_3 را تعیین کرد. باید توجه داشت که مقدار δ در هر قسمت برابر ماکزیمم اختلاف مجاز از مقدار مجاز تقسیم بر مقدار ناپیوستگی در آن قسمت است. در نتیجه داریم:

$$\begin{aligned}
 \delta_1 &= \frac{0.1}{1} = 0.1 \\
 \delta_2 &= \frac{0.06}{1} = 0.06 \\
 \delta_3 &= \frac{0.1}{2} = 0.05
 \end{aligned}$$

حال باید مقدار δ را تعیین کنیم. مقدار δ برابر مینیمم مقادیر δ_1 ، δ_2 و δ_3 است. در نتیجه داریم:

$$\delta = \min(\delta_1, \delta_2, \delta_3) = \min(0.1, 0.06, 0.05) = 0.05$$

سپس باید مقدار A را محاسبه کنیم:

$$A = -20 * \log_{10}(\delta) = -20 * \log_{10}(0.05) = 26.021$$

سپس باید مقدار β را محاسبه کنیم:

$$\beta = \begin{cases} 0.1102(A - 8.7) & A > 50 \\ 0.5842(A - 21)^{0.4} + 0.07886(A - 21) & 21 \leq A \leq 50 \\ 0 & A < 21 \end{cases}$$

$$A = 26.021$$

$$\beta = 1.51$$

ب) باید مقدار $\Delta\omega$ را محاسبه کنیم. برای این کار ابتدا باید مقادیر $\Delta\omega_1$ و $\Delta\omega_2$ را تعیین کرد.

$$\Delta\omega_1 = 0.1\pi$$

$$\Delta\omega_2 = 0.05\pi$$

حال باید مقدار $\Delta\omega$ را تعیین کنیم. مقدار $\Delta\omega$ برابر مینیمم مقادیر $\Delta\omega_1$ و $\Delta\omega_2$ است. در نتیجه داریم:

$$\Delta\omega = \min(\Delta\omega_1, \Delta\omega_2) = \min(0.1\pi, 0.05\pi) = 0.05\pi$$

سپس باید مقدار M را محاسبه کنیم:

$$M = \left\lceil \frac{A - 8}{2.285\Delta\omega} \right\rceil = \left\lceil \frac{26.021 - 8}{2.285 * 0.05\pi} \right\rceil = \lceil 50.208 \rceil = 51$$

اگر M فرد باشد، فیلتر مورد نظر نوع دوم و پاسخ فرکانسی آن در $\omega = \pm\pi$ ، صفر خواهد بود؛ اما پاسخ فرکانسی فیلتر مورد نظر $\omega = \pm\pi$ مخالف صفر است. پس M باید زوج باشد و در صورت فرد بودن باید یکی به آن افزوده شود. در نتیجه داریم:

$$M = 52$$

برای بدست آوردن پاسخ ضربه ایده‌آل $h_d[n]$ ابتدا $H_d(e^{j\omega})$ را محاسبه می‌کنیم.

$$H_d(e^{j\omega}) = u(\omega) - u(\omega - 0.25\pi) + 2 * u(\omega - 0.5\pi) \quad 0 \leq |\omega| \leq \pi$$

سپس با تبدیل فوریه معکوس گرفتن از $H_d(e^{j\omega})$ ، $h_d[n]$ را بدست می‌آوریم:

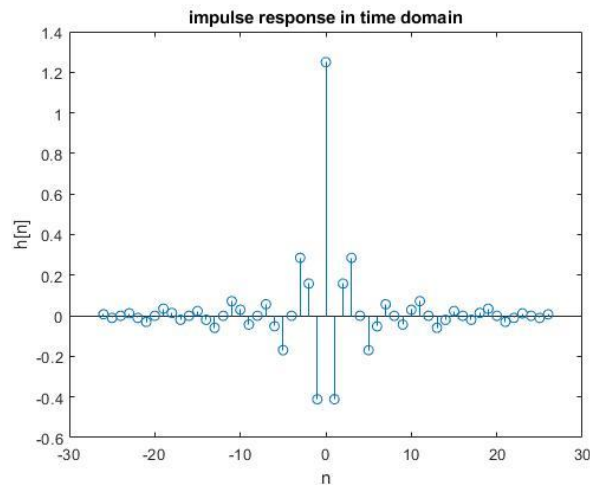
$$h_d[n] = 0.25 * \text{sinc}(0.25(n)) + 2 * \text{sinc}(n) - 2 * 0.5\text{sinc}(0.5(n))$$

سپس با استفاده از دستور Kaiser ضرایب فیلتر Kaiser را بدست می‌آوریم.

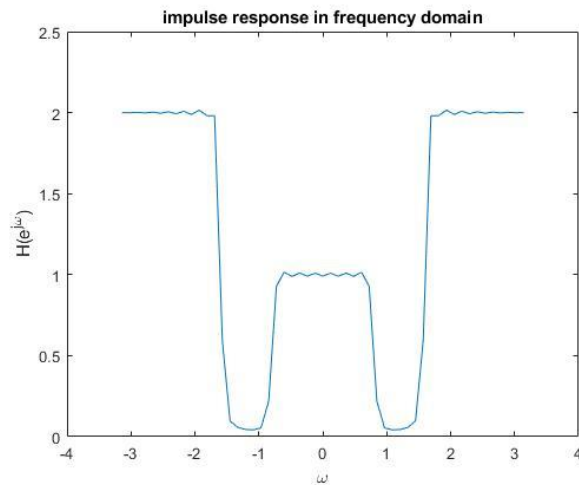
سپس با ضرب نقطه به نقطه پاسخ ضربه ایده‌آل شیفت یافته در فیلتر Kaiser، پاسخ ضربه فیلتر مورد نظر را محاسبه می‌کنیم.

سپس با تبدیل فوریه گرفتن از پاسخ ضربه فیلتر مورد نظر، پاسخ فرکانسی فیلتر را بدست می‌آوریم.

پاسخ ضربه فیلتر مورد نظر:



پاسخ فرکانسی فیلتر مورد نظر:



در انتها با استفاده از نمودار پاسخ فرکانسی فیلتر مورد نظر مقادیر ماکسیمم Overshoot و Undershoot را در passband و stopband بدست می آوریم.

Undershoot in $0.25\pi = 0.0375$

Overshoot in $0.25\pi = 0.0395$

Undershoot in $0.5\pi = 0.0258$

Overshoot in $0.5\pi = 0.0882$

کد متلب:

```
delta1 = 0.1;
delta2 = 0.06;
delta3 = 0.05;
delta = min([delta1,delta2,delta3]);
A = -20*log10(delta);
if (A < 21)
    beta = 0;
elseif (A >= 21 && A <= 50)
    beta = 0.5842*(A-21)^0.4 + 0.07886*(A-21);
elseif (A > 50)
    beta = 0.1102*(A-8.7);
end
deltaw1 = 0.1*pi;
deltaw2 = 0.05*pi;
deltaw = min(deltaw1,deltaw2);
M = ceil((A-8)/(2.285*deltaw));
```

```

if M/2 ~= floor(M/2)
    M = M+1;
end
KaiserFilter = kaiser(M+1,beta)';
n = -M/2:M/2;
hd = sin(0.25*pi*(n))./(pi*(n)) + 2*sin(pi*(n))./(pi*(n)) - 2*sin(0.5*pi*(n))./(pi*(n));
if M/2 == floor(M/2)
    hd(M/2+1) = 0.25 - 2*0.5 + 2*1;
end
h = hd.*KaiserFilter;
H = fftshift(fft(h,length(h)));
stem(n,h)
xlabel('n')
ylabel('h[n]')
title('impulse response in time domain')
figure
f = (0:length(H)-1)*2*pi/(length(H)-1) - pi;
plot(f,abs(H))
xlabel('\omega')
ylabel('H(e^{j\omega})')
title('impulse response in frequency domain')

```

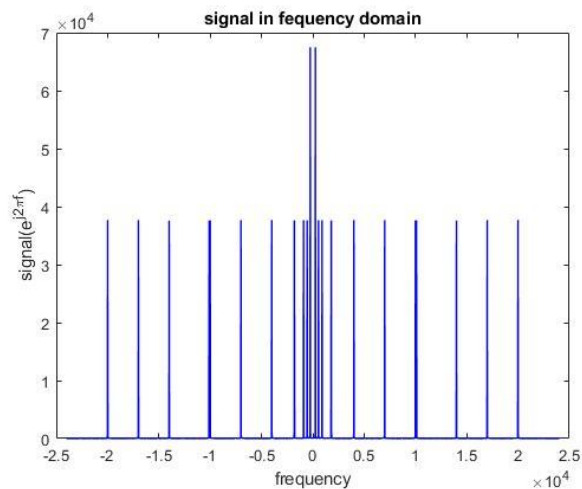
۲ طراحی فیلتر

این قسمت توسط استاد تدریس نشده است.

۳ طراحی فیلتر برای کار با داده صوتی

الف) ابتدا پاسخ فرکانسی سیگنال صوتی را رسم می‌کنیم.

پاسخ فرکانسی سیگنال صوتی:



کد متلب:

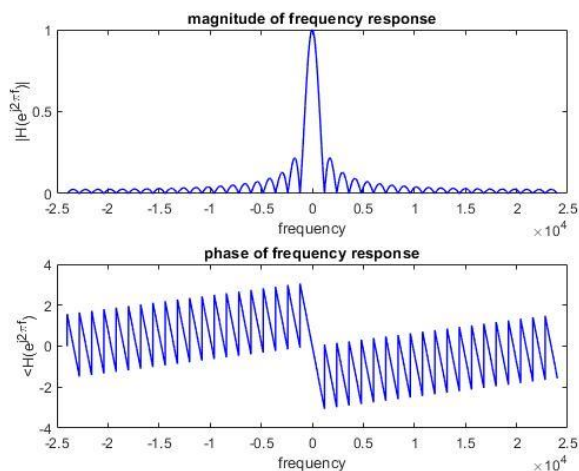
```
[signal,fs] = audioread('MultiFreq_Sig.wav');
signal = (signal(:,1)+signal(:,2))'/2;
frequency = -fs/2:fs/length(signal):fs/2-fs/length(signal);
signalF = fftshift(fft(signal));
figure();
plot(frequency,abs(signalF),'b','LineWidth',1)
xlabel('frequency')
ylabel('signal(e^{j2\pi f})')
title('signal in frequency domain')
```

تمامی فیلترها را به صورت پایین گذر طراحی می کنیم.

الف) میان گیر لغزان MA با ۴۰ نقطه:

ابتدا فیلتر مورد نظر را طراحی و سپس پاسخ فرکانسی آن را رسم می کنیم.

پاسخ فرکانسی فیلتر مورد نظر:




```

M = 40;
[signal,fs] = audioread('MultiFreq_Sig.wav');
signal = (signal(:,1)+signal(:,2))'/2;
h = ones(1,M)/M;
H = fftshift(fft(h,length(signal)));
frequency = -fs/2:fs/length(signal):fs/2-fs/length(signal);
figure();
subplot(2,1,1);
plot(frequency,abs(H),'b','LineWidth',1)
xlabel('frequency')
ylabel('|H(e^{j2\pi f})|')
title('magnitude of frequency response')
subplot(2,1,2);
plot(frequency,angle(H),'b','LineWidth',1)
xlabel('frequency')
ylabel('<H(e^{j2\pi f})')
title('phase of frequency response')

```

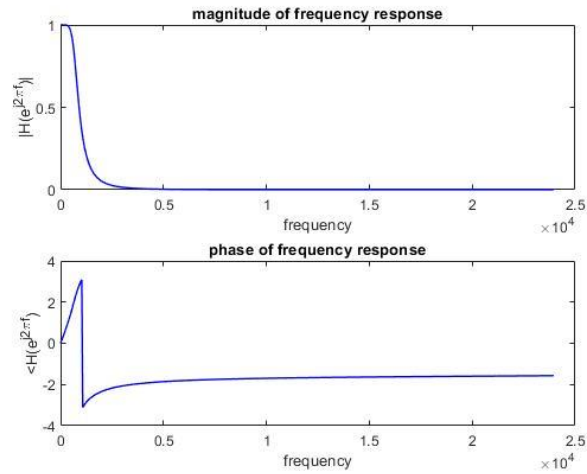
ب) باترورث با مرتبه ۳ و ۳۰:

ابتدا فیلترهای مورد نظر را طراحی و سپس پاسخ فرکانسی هر یک از آن‌ها را رسم می‌کنیم. برای طراحی فیلتر باترورث از دستور **butter** استفاده می‌کنیم. این دستور دو ورودی دریافت می‌کند. ورودی اول مرتبه فیلتر باترورث و ورودی دوم نسبت فرکانس قطع به فرکانس نمونه برداری است. همچنین این دستور دو خروجی دارد. خروجی اول ضرایب صورت و خروجی دوم ضرایب مخرج تبدیل Z فیلتر طراحی شده است.

ماکسیمم مرتبه قابل قبول برای طراحی فیلتری با مشخصات مذکور $N = 10$ است.

$N = 3$:

پاسخ فرکانسی فیلتر مورد نظر:



کد متلب:

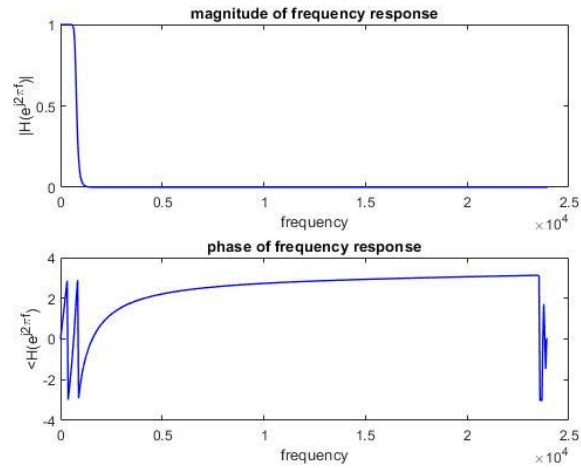
```

N = 3;
fc = 2000;
[~,fs] = audioread('MultiFreq_Sig.wav');
[b,a] = butter(N,fc/fs);
H = freqz(b,a);
frequency = 0:fs/(2*length(H)):fs/2-fs/(2*length(H));
figure();
subplot(2,1,1);
plot(frequency,abs(H),'b','LineWidth',1)
xlabel('frequency')
ylabel('|H(e^{j2\pi f})|')
title('magnitude of frequency response')
subplot(2,1,2);
plot(frequency,angle(H),'b','LineWidth',1)
xlabel('frequency')
ylabel('<H(e^{j2\pi f})')
title('phase of frequency response')

```

:N = 10

پاسخ فرکانسی فیلتر مورد نظر:



کد مطلب:

```

N = 10;
fc = 2000;
[~,fs] = audioread('MultiFreq_Sig.wav');
[b,a] = butter(N,fc/fs);
H = freqz(b,a)';
frequency = 0:fs/(2*length(H)):fs/2-fs/(2*length(H));
figure();
subplot(2,1,1);
plot(frequency,abs(H),'b','LineWidth',1)
xlabel('frequency')
ylabel('|H(e^{j2\pi f})|')
title('magnitude of frequency response')
subplot(2,1,2);
plot(frequency,angle(H),'b','LineWidth',1)
xlabel('frequency')
ylabel('<H(e^{j2\pi f})')
title('phase of frequency response')

```

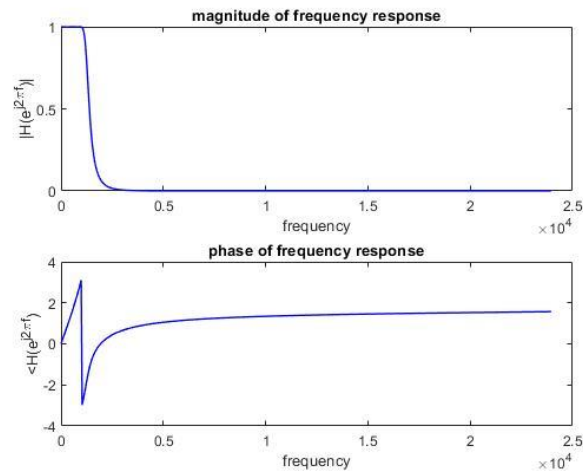
(ج) چبی شف نوع ۱ با حداقل مرتبه:

ابتدا فیلتر مورد نظر را طراحی و سپس پاسخ فرکانسی آن را رسم می‌کنیم. برای طراحی فیلتر چبی شف نوع ۱ از دستور `cheby1` استفاده می‌کنیم. این دستور سه ورودی دریافت می‌کند. ورودی اول مرتبه فیلتر چبی شف نوع ۱، ورودی دوم ماکسیمم مقدار مجاز در باند عبور بر حسب dB و ورودی سوم نسبت فرکانس قطع به فرکانس نمونه برداری است. همچنین این دستور دو خروجی دارد. خروجی اول ضرایب صورت و خروجی دوم ضرایب مخرج تبدیل Z فیلتر طراحی شده است.

با بررسی مقادیر مختلف برای مرتبه فیلتر، $N = 5$ را انتخاب می‌کنیم.

$N = 5$:

پاسخ فرکانسی فیلتر مورد نظر:



کد متلب:

```
N = 5;
fc = 2000;
[~,fs] = audioread('MultiFreq_Sig.wav');
[bmax,amax] = cheby1(N,0.01,fc/fs);
H = freqz(bmax,amax)';
frequency = 0:fs/(2*length(H)):fs/2-fs/(2*length(H));
figure();
subplot(2,1,1);
plot(frequency,abs(H),'b','LineWidth',1)
xlabel('frequency')
ylabel('|H(e^{j2\pi f})|')
title('magnitude of frequency response')
subplot(2,1,2);
plot(frequency,angle(H),'b','LineWidth',1)
xlabel('frequency')
ylabel('<H(e^{j2\pi f})')
title('phase of frequency response')
```

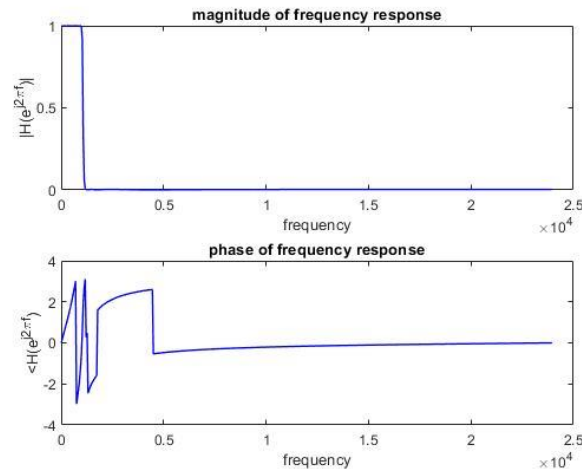
(د) الیپتیک مرتبه ۱۳:

ابتدا فیلتر مورد نظر را طراحی و سپس پاسخ فرکانسی آن را رسم می‌کنیم. برای طراحی فیلتر الیپتیک از دستور `ellip` استفاده می‌کنیم. این دستور چهار ورودی دریافت می‌کند. ورودی اول مرتبه فیلتر الیپتیک، ورودی دوم ماکسیمم مقدار مجاز در باند عبور بر حسب dB، ورودی سوم میزان افت باند قطع نسبت به باند عبور بر حسب dB و ورودی چهارم نسبت فرکانس قطع به فرکانس نمونه برداری است. همچنین این دستور دو خروجی دارد. خروجی اول ضرایب صورت و خروجی دوم ضرایب مخرج تبدیل Z فیلتر طراحی شده است.

ماکسیمم مرتبه قابل قبول برای طراحی فیلتری با مشخصات مذکور $N = 8$ است.

$N = 8$

پاسخ فرکانسی فیلتر مورد نظر:



کد متلب:

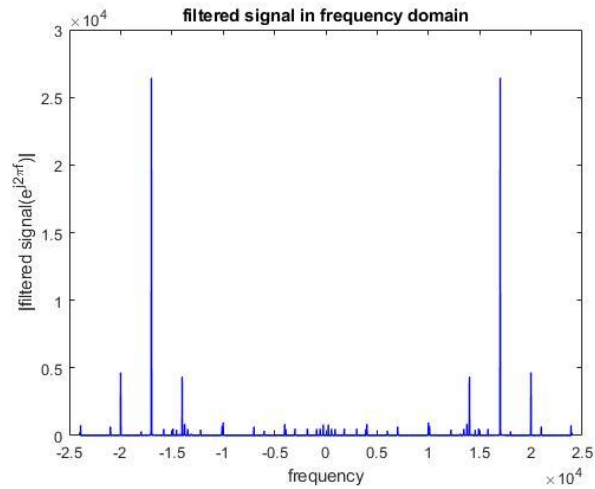
```
N = 8;
fc = 2000;
[~,fs] = audioread('MultiFreq_Sig.wav');
[bmax,amax] = ellip(N,0.01,50,fc/fs);
H = freqz(bmax,amax)';
frequency = 0:fs/(2*length(H)):fs/2-fs/(2*length(H));
figure();
subplot(2,1,1);
plot(frequency,abs(H),'b','LineWidth',1)
xlabel('frequency')
ylabel('|H(e^{j2\pi f})|')
title('magnitude of frequency response')
subplot(2,1,2);
plot(frequency,angle(H),'b','LineWidth',1)
xlabel('frequency')
```

```
ylabel('<H(e^{j2\pi f})>')
```

```
title('phase of frequency response')
```

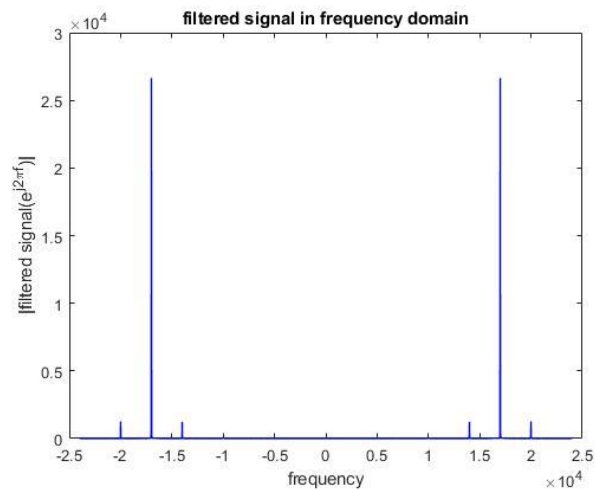
استخراج هارمونیک ۱۷۰۰۰: از آن جایی که تمامی فیلترها پایین گذر هستند، برای استخراج فرکانس مورد نظر باید ابتدا سیگنال صوتی را با حاملی با فرکانس مورد نظر مدوله، سپس با فیلترهای طراحی شده فیلتر و در انتها دمدوله کنیم.

میان گیر لغزان MA با ۴۰ نقطه:

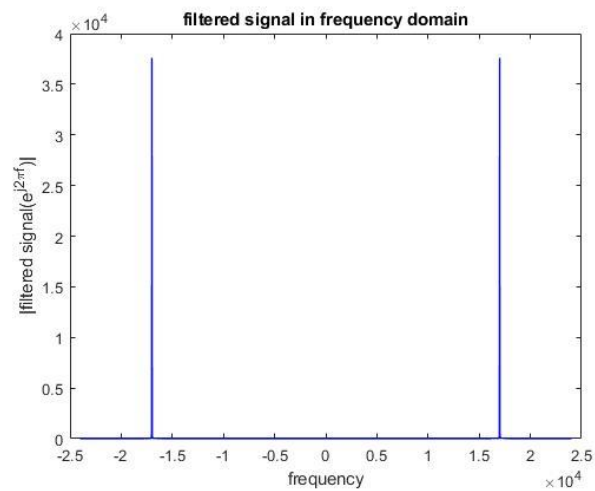


باترورث با مرتبه ۳ و ۳۰:

$N = 3$

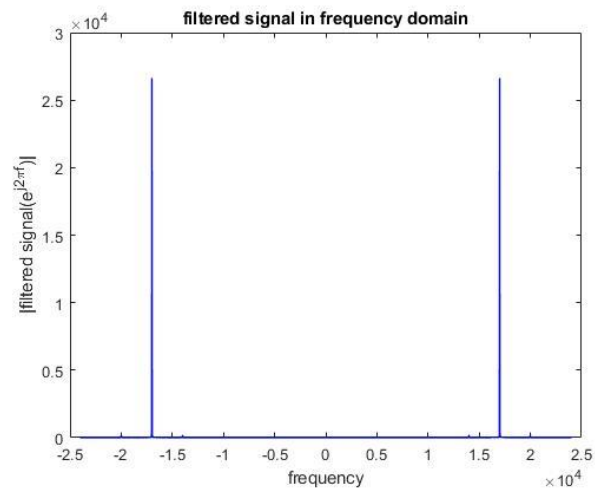


$N = 10$



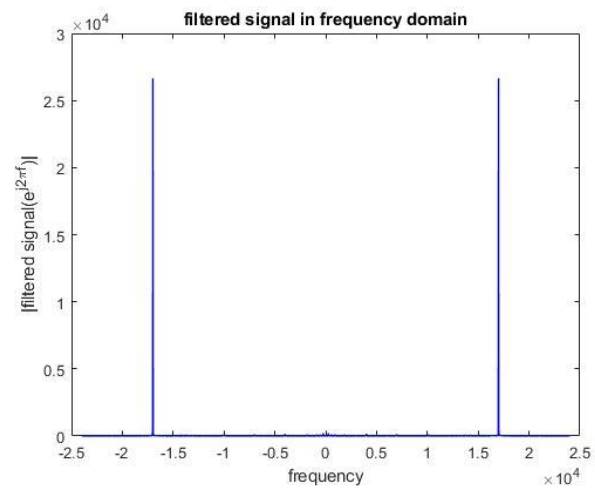
چی شف نوع اول با حداقل مرتبه:

$$N = 5$$



الیپتیک مرتبه ۱۳:

$$N = 8$$



کد متلب:

```
%%
fc = 17000;
M = 40;
[signal,fs] = audioread('MultiFreq_Sig.wav');
signal = (signal(:,1)+signal(:,2))'/2;
n = 0:length(signal)-1;
signal = signal.*cos(2*pi*fc.*n/fs);
h = ones(1,M)/M;
fsignal = filter(h,1,signal);
fsignal = 2*fsignal.*cos(2*pi*fc.*n/fs);
H = fftshift(fft(fsignal));
frequency = -fs/2:fs/length(fsignal):fs/2-fs/length(fsignal);
figure();
plot(frequency,abs(H),'b','LineWidth',1)
xlabel('frequency')
ylabel('|filtered signal(e^{j2\pi f})|')
title('filtered signal in frequency domain')
%%
```

```
fc = 17000;
N = 3;
fp = 2000;
[signal,fs] = audioread('MultiFreq_Sig.wav');
signal = (signal(:,1)+signal(:,2))'/2;
n = 0:length(signal)-1;
signal = signal.*cos(2*pi*fc.*n/fs);
[b,a] = butter(N,fp/fs);
fsignal = filter(b,a,signal);
fsignal = 2*fsignal.*cos(2*pi*fc.*n/fs);
H = fftshift(fft(fsignal));
frequency = -fs/2:fs/length(fsignal):fs/2-fs/length(fsignal);
figure();
plot(frequency,abs(H),'b','LineWidth',1)
```



```

xlabel('frequency')
ylabel('|filtered signal(e^{j2\pi f})|')
title('filtered signal in frequency domain')
%%
fc = 17000;
N = 10;
fp = 2000;
[signal,fs] = audioread('MultiFreq_Sig.wav');
signal = (signal(:,1)+signal(:,2))'/2;
n = 0:length(signal)-1;
signal = signal.*cos(2*pi*fc.*n/fs);
[b,a] = butter(N,fp/fs);
fsignal = filter(b,a,signal);
fsignal = 2*fsignal.*cos(2*pi*fc.*n/fs);
H = fftshift(fft(signal));
frequency = -fs/2:fs/length(fsignal):fs/2-fs/length(fsignal);
figure();
plot(frequency,abs(H),'b','LineWidth',1)
xlabel('frequency')
ylabel('|filtered signal(e^{j2\pi f})|')
title('filtered signal in frequency domain')
%%
fc = 17000;
N = 5;
fp = 2000;
[signal,fs] = audioread('MultiFreq_Sig.wav');
signal = (signal(:,1)+signal(:,2))'/2;
n = 0:length(signal)-1;
signal = signal.*cos(2*pi*fc.*n/fs);
[b,a] = cheby1(N,0.01,fp/fs);
fsignal = filter(b,a,signal);
fsignal = 2*fsignal.*cos(2*pi*fc.*n/fs);

```

```

H = fftshift(fft(fsignal));
frequency = -fs/2:fs/length(fsignal):fs/2-fs/length(fsignal);
figure();
plot(frequency,abs(H),'b','LineWidth',1)
xlabel('frequency')
ylabel('|filtered signal(e^{j2\pi f})|')
title('filtered signal in frequency domain')

%%
fc = 17000;
N = 8;
fp = 2000;
[signal,fs] = audioread('MultiFreq_Sig.wav');
signal = (signal(:,1)+signal(:,2))'/2;
n = 0:length(signal)-1;
signal = signal.*cos(2*pi*fc.*n/fs);
[b,a] = ellip(N,0.01,50,fp/fs);
fsignal = filter(b,a,signal);
fsignal = 2*fsignal.*cos(2*pi*fc.*n/fs);
H = fftshift(fft(fsignal));
frequency = -fs/2:fs/length(fsignal):fs/2-fs/length(fsignal);
figure();
plot(frequency,abs(H),'b','LineWidth',1)
xlabel('frequency')
ylabel('|filtered signal(e^{j2\pi f})|')
title('filtered signal in frequency domain')

```

۴ سیستم‌های ماکسیمم فاز

(الف)

$$H(z) = 1 - 4z^{-1} + 8z^{-2} - 64z^{-3} + 201z^{-4} - 450z^{-5} + 650z^{-6}$$

برای محاسبه صفرهای تابع تبدیل، ابتدا بردار \mathbf{b} (ضرایب چندجمله‌ای صورت) را تشکیل می‌دهیم. سپس با استفاده از دستور `roots`، صفرهای تابع سیستم را محاسبه می‌کنیم.

$$z_1 = -1.888 - 3.203i$$

$$z_2 = -1.888 + 3.203i$$

$$z_3 = 3.984$$

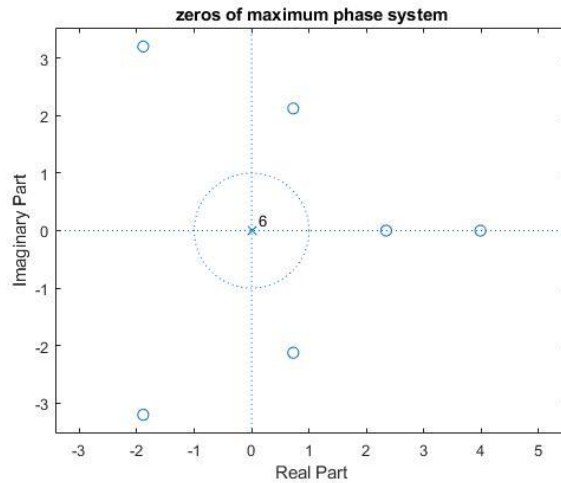
$$z_4 = 2.343$$

$$z_5 = 0.725 - 2.124i$$

$$z_6 = 0.725 + 2.124i$$

سپس با استفاده از دستور zplane نمودار صفر و قطب سیستم ماکسیمم فاز را رسم می کنیم.

نمودار صفر و قطب سیستم ماکسیمم فاز:



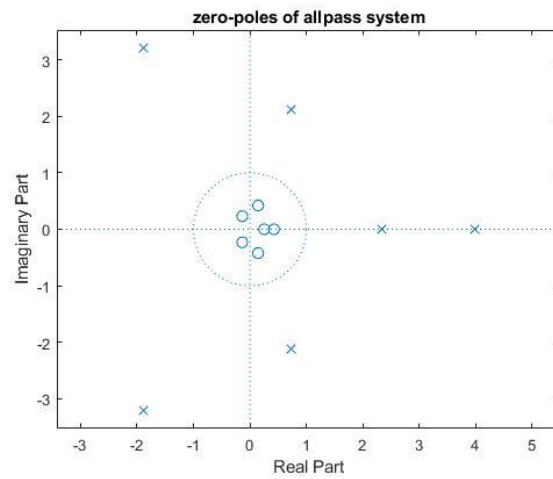
ب) سیستم تمام گذر باید صفرهای خارج دایره ی واحد را به صفرهای داخل دایره ی واحد تبدیل کند. یعنی اگر سیستم ماکسیمم فاز صفری خارج دایره واحد در $z = re^{j\theta}$ داشته باشد، سیستم تمام گذر باید آن را به داخل دایره واحد به $z = \frac{1}{r}e^{j\theta}$ منتقل کند. بنابراین باید صفرهای سیستم ماکسیمم فاز، قطب های سیستم تمام گذر و صفرهای متناظر سیستم مینیمم فاز، صفر های سیستم تمام گذر باشد. در نتیجه داریم:

$$H_{ap} = \prod_{i=1}^6 \frac{z^{-1} - r_i e^{-j\theta_i}}{1 - r_i e^{j\theta_i} z^{-1}} = \prod_{i=1}^6 \frac{-r_i e^{-j\theta_i} (1 - \frac{1}{r_i} e^{j\theta_i} z^{-1})}{1 - r_i e^{j\theta_i} z^{-1}}$$

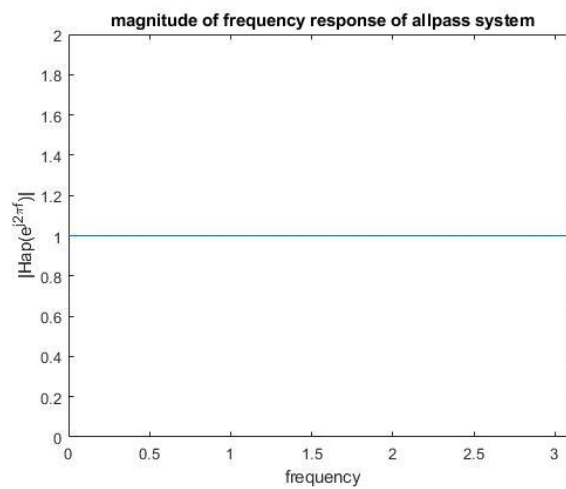
که در آن $z_i = r_i e^{j\theta_i} \quad i = 1, 2, \dots, 6$ صفرهای سیستم ماکسیمم فاز هستند.

با استفاده از دستورهای tf2zpk و zp2tf، بردارهای b (ضرایب چندجمله ای صورت) و a (ضرایب چندجمله ای مخرج) سیستم تمام گذر را بدست می آوریم. سپس نمودارهای صفر و قطب و اندازه پاسخ فرکانسی سیستم تمام گذر را رسم می کنیم.

نمودار صفر و قطب سیستم تمام گذر:

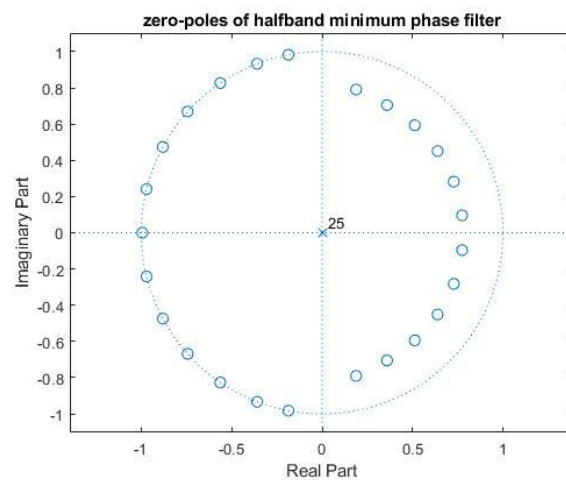


اندازه پاسخ فرکانسی سیستم تمام‌گذر:

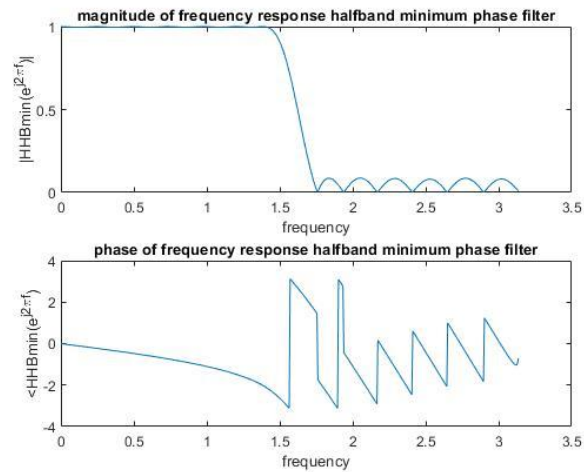


ج) با استفاده از دستور firhalfband بردارهای b (ضرایب چندجمله‌ای صورت) و a (ضرایب چندجمله‌ای مخرج) فیلتر مینیمم فاز halfband با درجه ۲۵ و فرکانس عبور ۰/۴۵ (نرمالیزه شده) را بدست می‌آوریم. سپس نمودارهای صفر و قطب، پاسخ فرکانسی و پاسخ ضربه را رسم می‌کنیم.

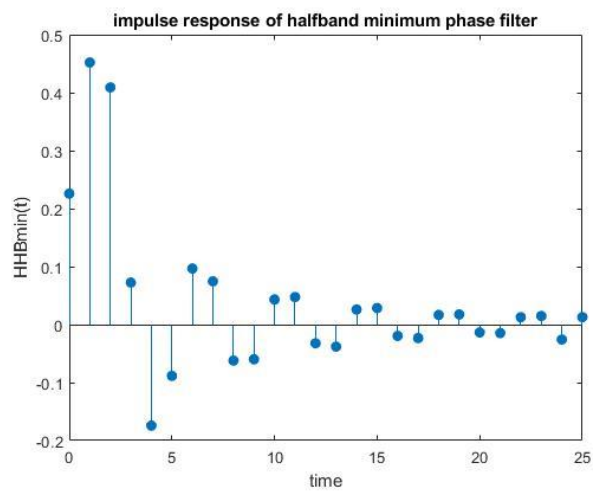
نمودار صفر و قطب:



پاسخ فرکانسی:

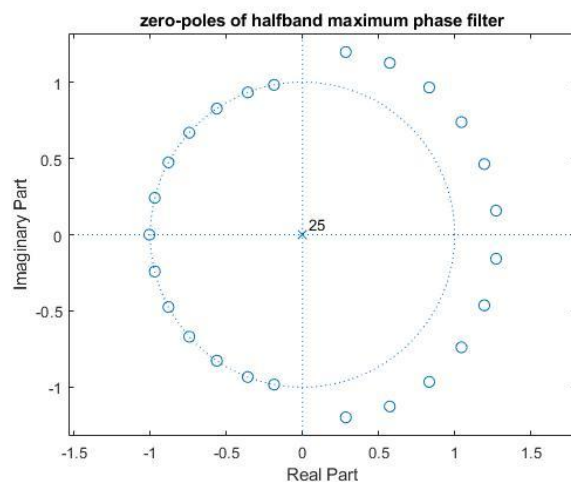


پاسخ ضربه:

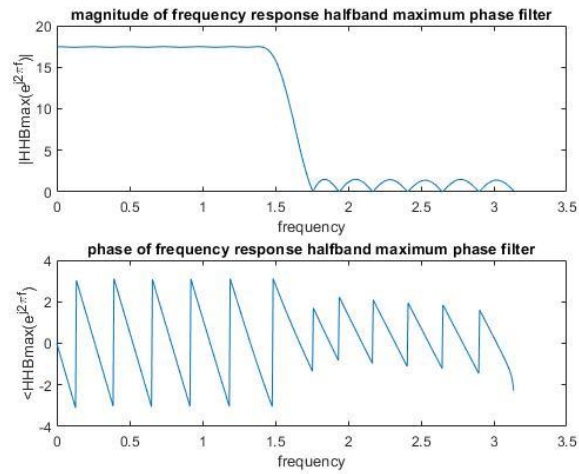


فیلتر ماکسیمم فاز halfband را از روی سیستم مینیمم فاز halfband و با استفاده از روش گفته شده در قسمت ب بدست می آوریم. سپس نمودارهای صفر و قطب، پاسخ فرکانسی و پاسخ ضربه را رسم می کنیم.

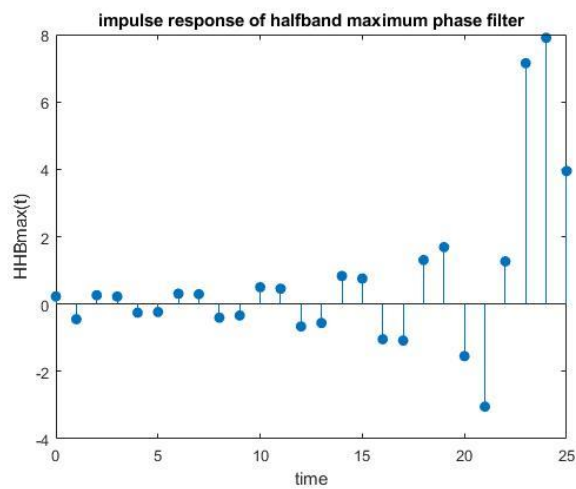
نمودار صفر و قطب:



پاسخ فرکانسی:



پاسخ ضربه:



کد متلب:

```
bHBmin = firhalfband(25,0.45,'minphase');
aHBmin = 1;
HHBmin = freqz(bHBmin,aHBmin);
figure();
zplane(bHBmin,aHBmin)
title('zero-poles of halfband minimum phase filter')
f = 0:pi/length(HHBmin):(length(HHBmin)-1)*pi/length(HHBmin);
figure();
subplot(2,1,1);
plot(f,abs(HHBmin))
xlabel('frequency')
ylabel('|HHBmin(e^{j2\pi f})|')
```

```

title('magnitude of frequency response halfband minimum phase filter')
subplot(2,1,2);
plot(f,angle(HHBmin))
xlabel('frequency')
ylabel('<HHBmin(e^{j2\pi f})')
title('phase of frequency response halfband minimum phase filter')
figure();
impz(bHBmin,aHBmin);
xlabel('time')
ylabel('HHBmin(t)')
title('impulse response of halfband minimum phase filter')
[ZHBmin,PHBmin,KHBmin] = tf2zpk(bHBmin,aHBmin);
ZHBmax = conj(1./ZHBmin);
PHBmax = PHBmin;
KHBmax = KHBmin;
[bHBmax,aHBmax] = zp2tf(ZHBmax,PHBmax,KHBmax);
HHBmax = freqz(bHBmax,aHBmax);
figure();
zplane(bHBmax,aHBmax)
title('zero-poles of halfband maximum phase filter')
f = 0:pi/length(HHBmax):(length(HHBmax)-1)*pi/length(HHBmax);
figure();
subplot(2,1,1);
plot(f,abs(HHBmax))
xlabel('frequency')
ylabel('|HHBmax(e^{j2\pi f})|')
title('magnitude of frequency response halfband maximum phase filter')
subplot(2,1,2);
plot(f,angle(HHBmax))
xlabel('frequency')
ylabel('<HHBmax(e^{j2\pi f})')
title('phase of frequency response halfband maximum phase filter')

```

```
figure();  
impz(bHBmax,aHBmax);  
xlabel('time')  
ylabel('HHBmax(t)')  
title('impulse response of halfband maximum phase filter')
```