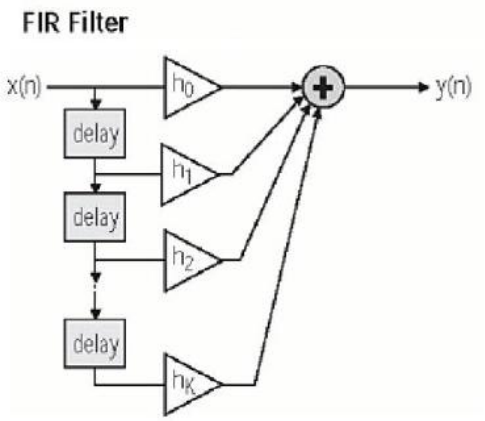
به نام خدا

مصطفی چرکزی

آزمایش پنجم پردازش سیگنال های دیجیتال

آزمایش ۵: طراحی فیلترهای دیجیتال FIR

در طراحی فیلتر، پس از انتخاب نوع فیلتر، باید با توجه به مشخصات مطلوب، درجه تابع تبدیل را تعیین کرده و سپس تابع تبدیل را با توجه به مشخصات فیلتر تعیین کنیم.



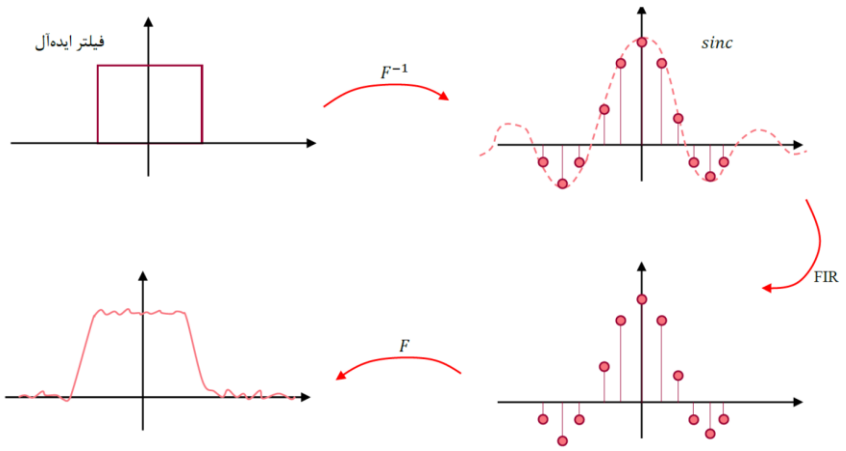
دستورات متلب مورد نیاز:

جعبه ابزار پردازش سیگنال: Signal processing toolbox:

blackman	butter	buttord	chebwin	cheb1ord	Cheb2ord	Cheby1
Cheby2	ellip	ellipord	fir1	fir2	Firpm	firpmord
freqz	hanning	hamming	kaiser			

بخش ۱: طراحی فیلتر FIR

ساده‌ترین روش برای طراحی فیلتر FIR، این است که از پاسخ فرکانسی فیلتر مطلوب، تبدیل فوریه معکوس گرفته و سپس تعداد محدودی از نقاط آن را انتخاب کنیم. اما این کار باعث ایجاد ریپل‌هایی در پاسخ فرکانسی میشود که اثر Gibbs نام دارد. برای حذف این و یا کاهش این اثر، پاسخ ضربه نامحدود را از پنجره‌ی محدود و مناسب عبور می‌دهیم. در متلب فرامینی وجود دارد که با استفاده از آن‌ها می‌توان بردار ضرایب پاسخ ضربه فیلتر FIR با مشخصات مطلوب را بدست آورد.



برای طراحی فیلتر FIR ابتدا باید درجه فیلتر FIR مشخص شود. برای این منظور از دستورات زیر استفاده می‌شود:

$$[N, \omega_n, beta, ftype] = kaiserord(Fedge, Aval, dev)$$

or

$$[N, \omega_n, beta, ftype] = kaiserord(Fedge, Aval, dev, FS)$$

$Fedge =$ فرکانس‌های مرزی , $Aval =$ مقدار دامنه در هر باند , $dev =$ مقدار ریپل در هر باند , $Fs =$ فرکانس‌نمونه‌برداری

$beta =$ فاکتور شکل‌دهی , $ftype =$ نوع فیلتر

میزان تضعیف لوب‌های فرعی

بعد از تعیین مرتبه‌ی فیلتر FIR، با استفاده از دستورات زیر، ضرایب پاسخ ضربه‌ی فیلتر را بدست می‌آوریم:

$b = fir1(N, \omega_n, 'filtertype')$
$b = fir1(N, \omega_n, taper)$
$taper = blackman(N)$, $taper = hamming(N)$, $taper = hanning(N)$, $taper = chebwin(N)$, $taper = kaiser(N, beta)$
$tapper = triang(N)$, $tapper = rectwin(N)$

$$b = fir1(N, W_n)$$

$$b = fir1(N, W_n, 'high')$$

$$b = fir1(N, W_n, 'stop')$$

$$b = fir1(N, W_n, taper)$$

نکته: همان‌گونه که پیش‌تر توضیح داده شد، برای حذف اثر نوسان در فیلتر FIR از پنجره استفاده می‌کنیم. در حالت استفاده بدون taper، فرمان به صورت پیش‌فرض از پنجره‌ی همینگ hamming استفاده خواهد کرد. اما در صورت نیاز به استفاده از پنجره های دیگر، می‌توان با فرمان‌های زیر، پنجره‌های دیگری نیز ایجاد کرد.

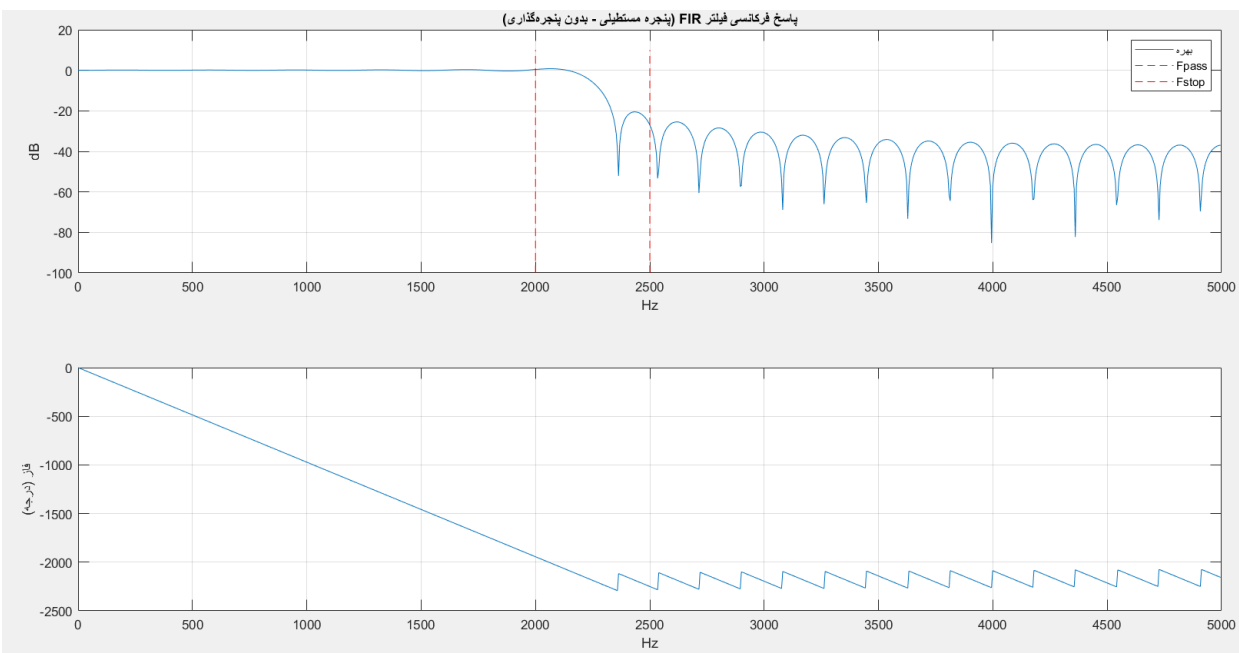
$taper = blackman(N)$ $\quad$ $taper = hamming(N)$ $\quad$ $taper = hanning(N)$

$taper = chebwin(N)$ $\quad$ $taper = kaiser(N, beta)$

فعالیت ۱: با استفاده از دستور fir1، فیلتر FIR پایین‌گذر با فاز خطی و دارای مشخصات مطرح شده را طراحی کنید: فرکانس مرز باند عبور 2kHz، فرکانس مرز باند توقف 2.5kHz، فرکانس نمونه برداری 10kHz، ریپل باند عبور و باند توقف $\delta_s = \delta_p = 0.005$.

```
Fs = 10000;
Fpass = 2000;
Fstop = 2500;
Rp = 0.005;
Rs = 0.005;
```

```
% تخمین مرتبه فیلتر
[N, Wn, beta, ftype] = kaiserord([Fpass Fstop], [1 0], [Rp Rs], Fs);

% طراحی فیلتر با پنجره مستطیلی (بدون پنجره‌گذاری)
b = fir1(N, Wn, ftype, rectwin(N+1));   % ← rectwin = 1 همه ضرایب

[H, f] = freqz(b, 1, 1024, Fs);
mag_dB = 20*log10(abs(H));
phase_deg = unwrap(angle(H)) * 180/pi;

figure;
subplot(2,1,1);
plot(f, mag_dB); grid on;
title('پاسخ فرکانسی فیلتر FIR (پنجره مستطیلی - بدون پنجره‌گذاری)');
xlabel('Hz'); ylabel('dB');
xlim([0 Fs/2]);
line([Fpass Fpass], [-100 10], 'Color','r','LineStyle','--');
line([Fstop Fstop], [-100 10], 'Color','r','LineStyle','--');
legend('بهره', 'Fpass', 'Fstop');

subplot(2,1,2);
plot(f, phase_deg); grid on;
xlabel('Hz'); ylabel('فاز (درجه)');
xlim([0 Fs/2]);
```



پس از طراحی فیلتر، گین پاسخ فرکانسی و فاز آن را رسم کنید. آیا فیلتر طراحی شده دارای مشخصات مطلوبی می‌باشد؟ با تغییر چه پارامتری می‌توان فیلتر را مطلوب‌تر طراحی کرد؟

آیا فیلتر طراحی شده دارای مشخصات مطلوب است؟

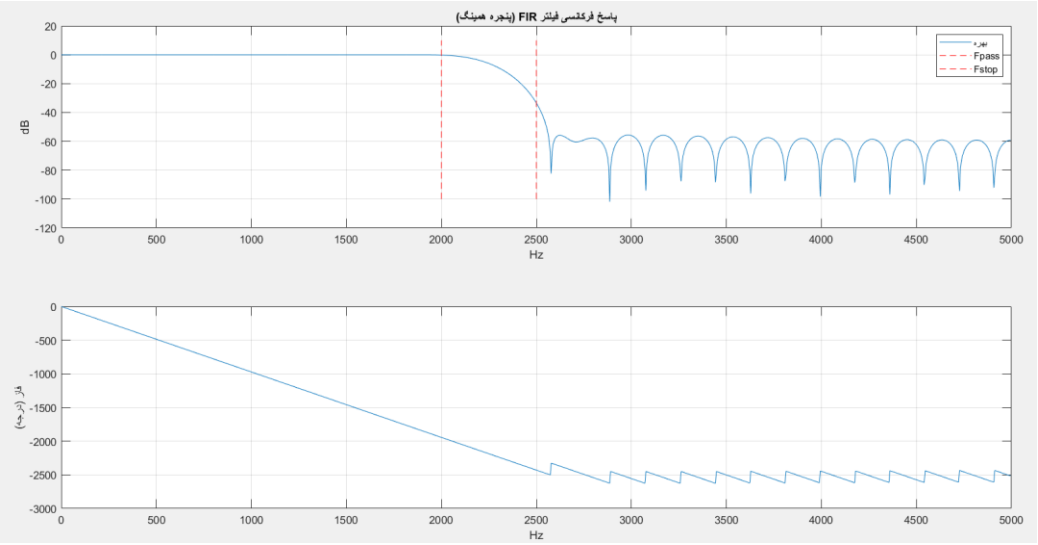
خیر، فیلتر بدون پنجره گذاری دارای لوب‌های جانبی بزرگ و ریپل است. پاسخ فرکانسی نوسانات شدیدی در باند توقف دارد و تضعیف مناسبی را تأمین نمی‌کند. بنابراین مطلوب نیست.

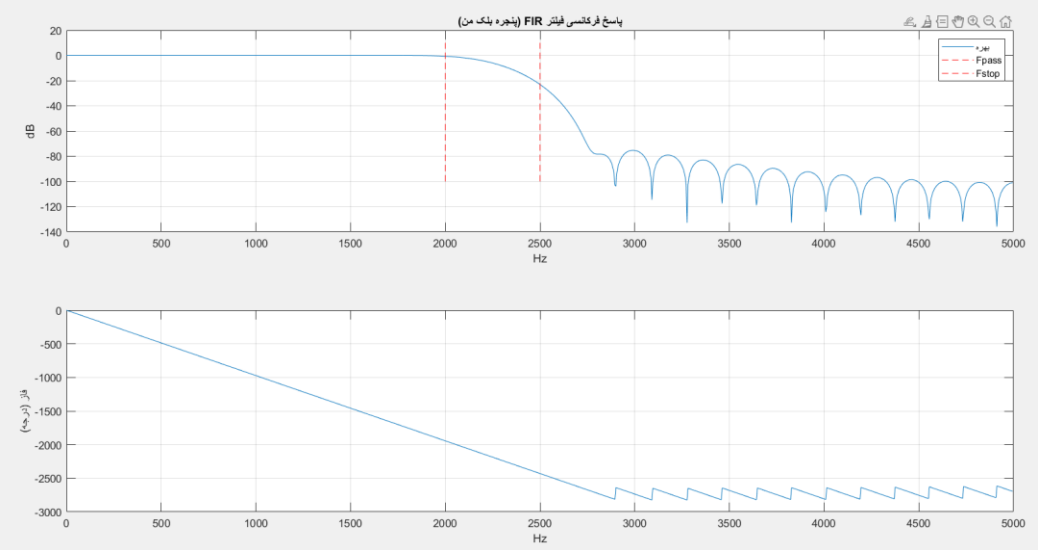
با تغییر چه پارامتری می‌توان فیلتر را مطلوب‌تر طراحی کرد؟

با استفاده از روش های پنجره گذاری مانند Hamming، Blackman (که در فعالیت ۲ انجام می‌شود) می‌توان لوب‌های جانبی را تضعیف کرد و مشخصات را بهبود بخشید. همچنین افزایش مرتبه فیلتر شیب باند گذار را تندتر می‌کند.

فعالیت ۲: فعالیت ۱ را بار دیگر با استفاده از پنجره‌های hamming و blackman انجام دهید و نتایج را مقایسه کنید و شرح دهید.

```
% پنجره Hamming
b_hamm = fir1(N, Wn, ftype, hamming(N+1));
% پنجره Blackman
b_black = fir1(N, Wn, ftype, blackman(N+1));
```





مشاهده میشود که با استفاده از پنجره گذاری نسبت به فعالیت اول که از پنجره گذاری استفاده نکردیم پاسخ فرکانسی بهتری دریافت شد.

با مقایسه دو روش پنجره گذاری نیز:

پنجره همینگ: لوب اصلی نسبتاً باریک.

پنجره بلکمن: لوب اصلی پهن‌تر.

تضعیف در لوب جانبی بلکمن نیز بهتر است.

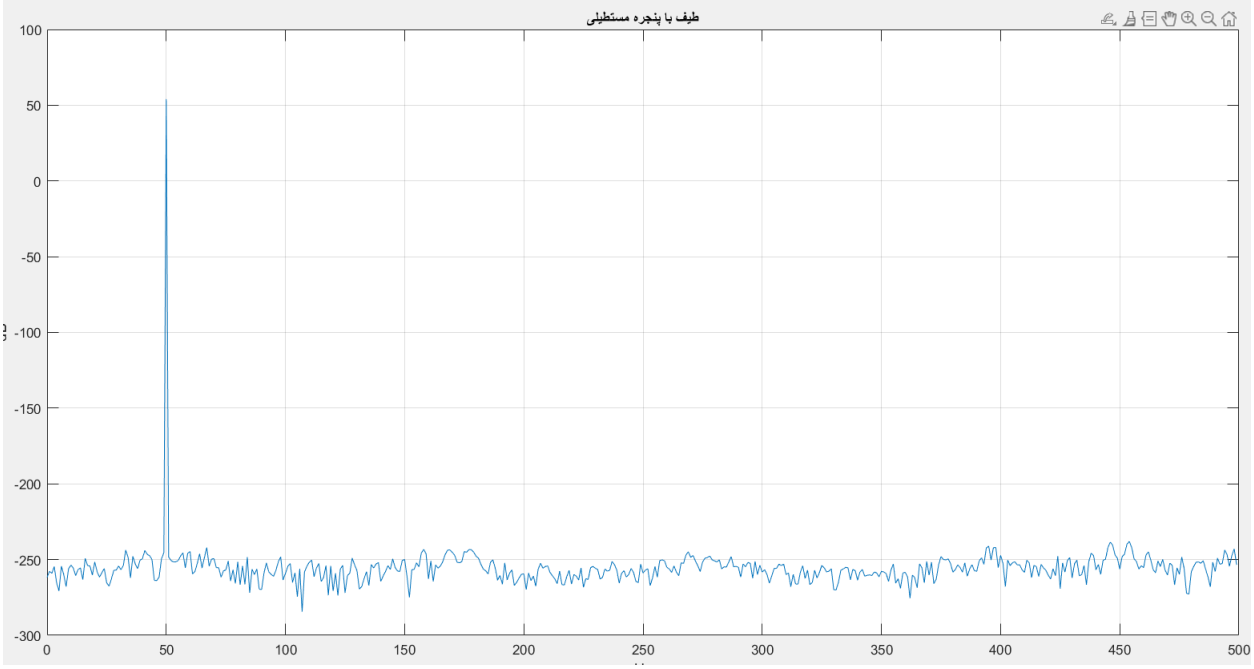
نتیجه: اگر حذف نویز در باند توقف اولویت دارد، بلکمن مناسب‌تر است. اگر تفکیک فرکانسی بالا مهم است، همینگ بهتر عمل می‌کند.

فعالیت ۳: الف) یک سیگنال سینوسی ساده با فرکانس $f_0 = 50$ هرتز و طول یک ثانیه را ایجاد کنید.

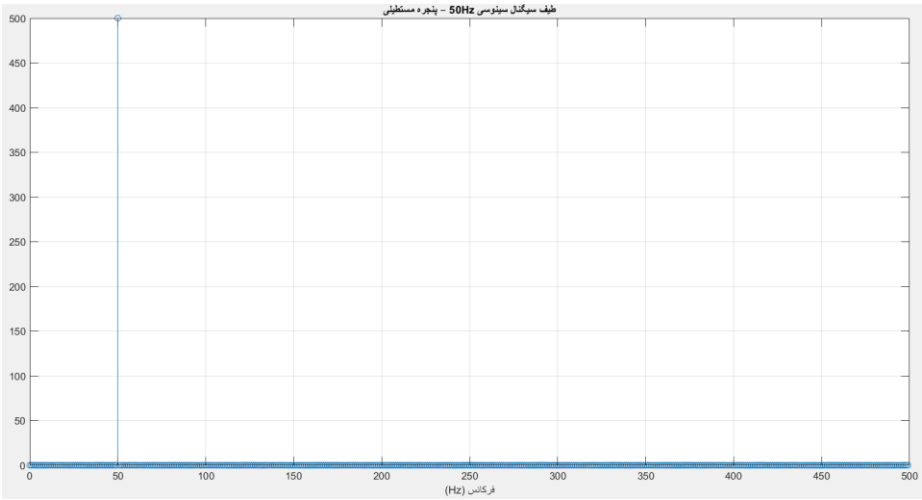
ب) DFT سیگنال را با استفاده از پنجره‌ی مستطیلی محاسبه کنید.

ج) طیف حاصل را نمایش دهید.

```
% A
Fs = 1000;
t = 0:1/Fs:1-1/Fs;
f0 = 50;
x = sin(2*pi*f0*t);

% B
N = length(x);
X = fft(x);
X_mag = abs(X(1:round(N/2)));
f = (0:round(N/2)-1) * (Fs/N);

% C
figure;
plot(f, 20*log10(X_mag)); grid on;
xlabel('Hz'); ylabel('dB'); title('طیف با پنجره مستطیلی');
```




```
fs = 1000;              % فرکانس نمونه‌برداری (Hz)
f0 = 50;                % فرکانس سیگنال (Hz)
t = 0:1/fs:1-1/fs;      % طول 1 ثانیه
x = sin(2*pi*f0*t);

% DFT (بدون ضرب در پنجره) با پنجره مستطیلی
X = fft(x);
N = length(x);
f = (0:N-1)*(fs/N);

% نمایش طیف
figure;
stem(f(1:N/2), abs(X(1:N/2)));
xlabel('فرکانس (Hz)'); ylabel('دامنه');
title('طیف سیگنال سینوسی 50Hz – پنجره مستطیلی');
grid on;
```

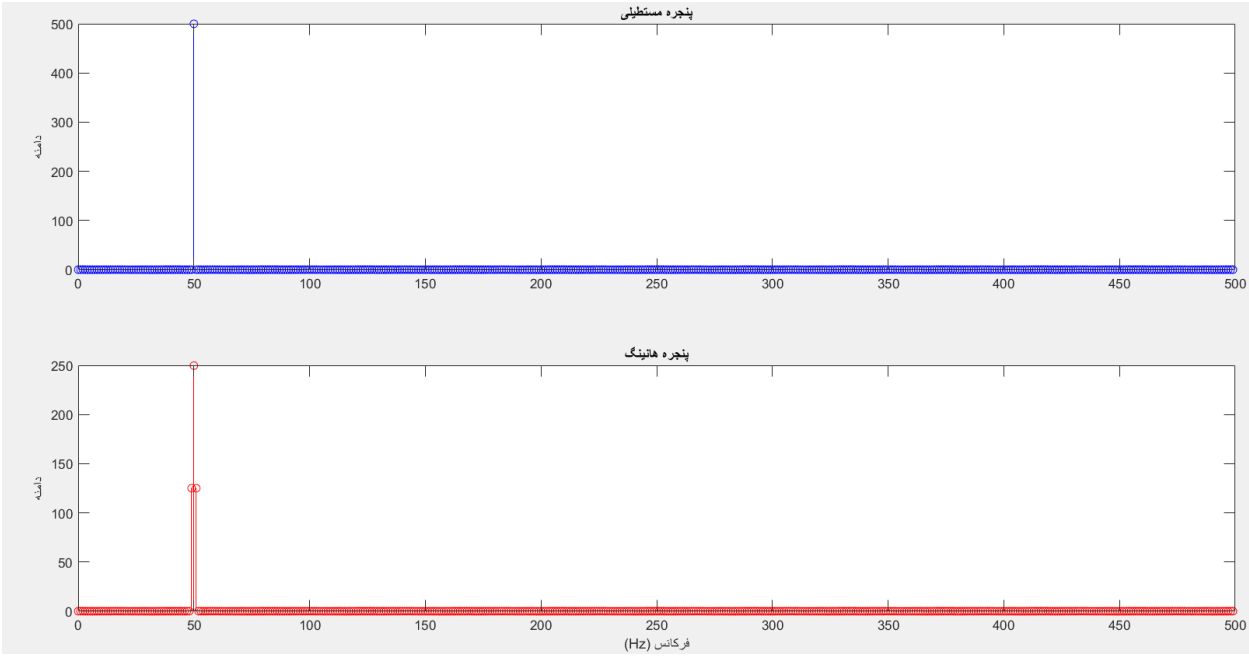


فعالیت ۴: الف) سیگنال مشابه فعالیت قبل را تولید کنید.

ب) DFT سیگنال را با استفاده از پنجره هانینگ محاسبه و طیف آن را مشاهده کنید.

ج) طیف پنجره مستطیلی و هانینگ را در یک شکل مقایسه کنید.

```
fs = 1000;              % فرکانس نمونه‌برداری (Hz)
f0 = 50;                % فرکانس سیگنال (Hz)
t = 0:1/fs:1-1/fs;      % طول 1 ثانیه
x = sin(2*pi*f0*t);

N = length(x);
f = (0:N-1)*(fs/N);

% پنجره هانینگ
win_hann = hann(length(x))';
x_hann = x .* win_hann;
X_hann = fft(x_hann);

% پنجره مستطیلی (همان x)
X_rect = fft(x);

figure;
subplot(2,1,1);
stem(f(1:N/2), abs(X_rect(1:N/2)), 'b');
title('پنجره مستطیلی'); ylabel('دامنه');
subplot(2,1,2);
stem(f(1:N/2), abs(X_hann(1:N/2)), 'r');
title('پنجره هانینگ'); xlabel('فرکانس (Hz)'); ylabel('دامنه');
```



مشاهده:

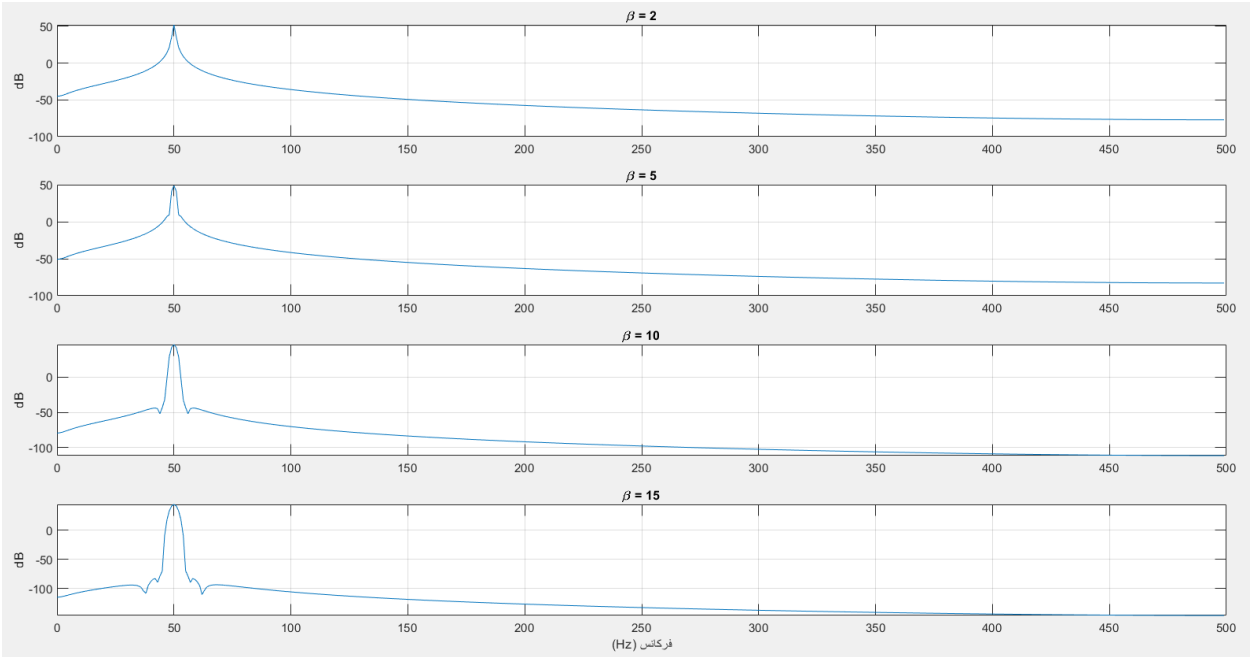
پنجره هانینگ دامنه را کاهش می‌دهد ولی لوب اصلی پهن‌تر می‌شود.

فعالیت ۵: الف) پنجره کیزر با پارامترهای مختلف $\beta = 2, \beta = 10$ را اعمال کنید.

ب) طیف سیگنال را برای هر دو مقدار β محاسبه کرده و نمایش دهید.

ج) اثر تغییرات پارامتر β را روی پخشی باند طیف بررسی کنید.

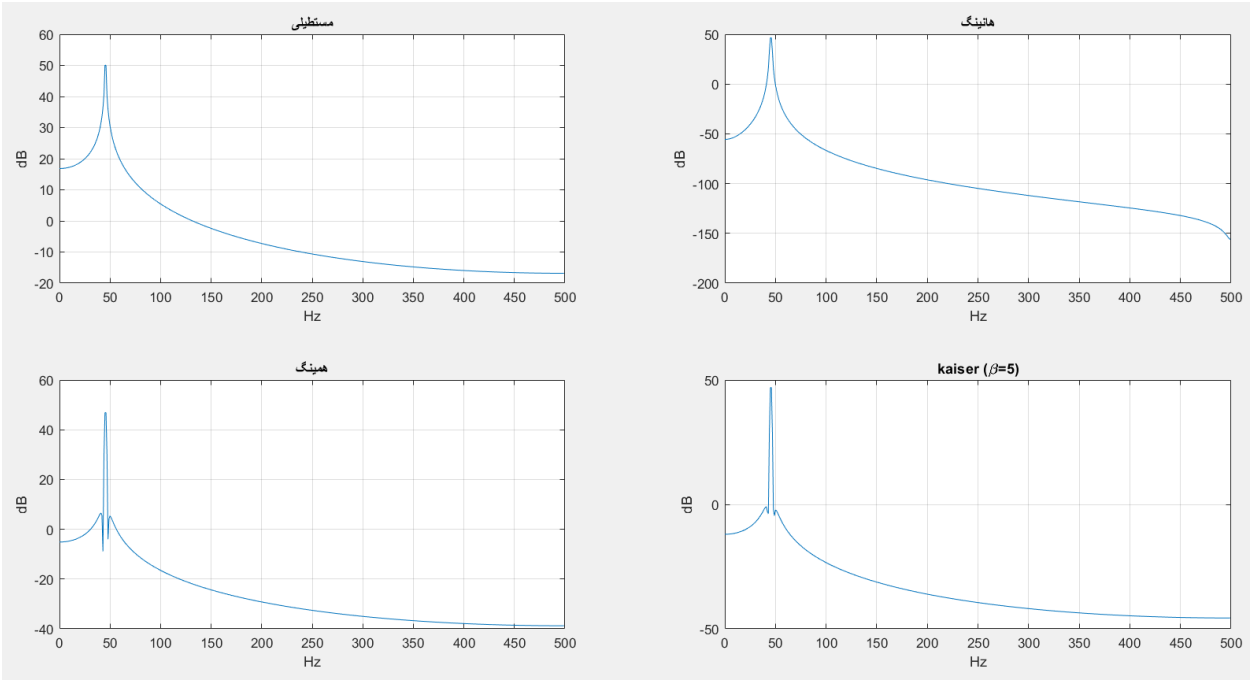
```
fs = 1000;              % فرکانس نمونه‌برداری (Hz)
f0 = 50;                % فرکانس سیگنال (Hz)
t = 0:1/fs:1-1/fs;      % طول 1 ثانیه
x = sin(2*pi*f0*t);

N = length(x);
f = (0:N-1)*(fs/N);

beta_vals = [2 5 10 15];
figure;
for i = 1:4
    win = kaiser(N, beta_vals(i));
    x_win = x .* win';
    X_win = fft(x_win);
    subplot(4,1,i);
    plot(f(1:N/2), 20*log10(abs(X_win(1:N/2))));
    title(['\beta = ' num2str(beta_vals(i))]);
    ylabel('dB'); grid on;
end
xlabel('فرکانس (Hz)');
```



با افزایش β ، لوب‌های جانبی بیشتر تضعیف می‌شوند ولی پهنای لوب اصلی افزایش می‌یابد

فعالیت ۶: یک سیگنال سینوسی با فرکانس غیر مضربی از نرخ نمونه‌برداری (مانند ۴۵.۵ هرتز) ایجاد کنید. سپس از پنجره‌های مختلف (مستطیلی، هانینگ، همینگ، کیزر) استفاده کنید و طیف آن‌ها را نمایش دهید. در هر پنجره نشت فرکانسی را بررسی و مقایسه کنید.

```
fs = 1000;
t = 0:1/fs:1-1/fs;
f0 = 45.5;
x = sin(2*pi*f0*t);
N = length(x);
f = (0:N-1)*(fs/N);

windows = {rectwin(N), hann(N), hamming(N), kaiser(N,5)};
names = {'مستطیلی', 'هانینگ', 'همینگ', 'kaiser (\beta=5)'};

figure;
for i = 1:4
    xw = x .* windows{i}';
    Xw = fft(xw);
    subplot(2,2,i);
    plot(f(1:N/2), 20*log10(abs(Xw(1:N/2))));
    title(names{i}); xlabel('Hz'); ylabel('dB');
    grid on;
end
```



فرکانس 45.5 هرتز مضربی از Δf = fs/N = 1Hz نیست، بنابراین انرژی به باندهای کناری نشت می‌کند.

پنجره مستطیلی بدترین نشت را نشان می‌دهد. پنجره‌های هانینگ، همینگ و کیزر نشت را کاهش می‌دهند.

فعالیت ۷: یک فعالیت در مورد تاخیر گروه و انواع پنجره‌گذاری بر روی سیگنال صوتی طراحی نمایید.

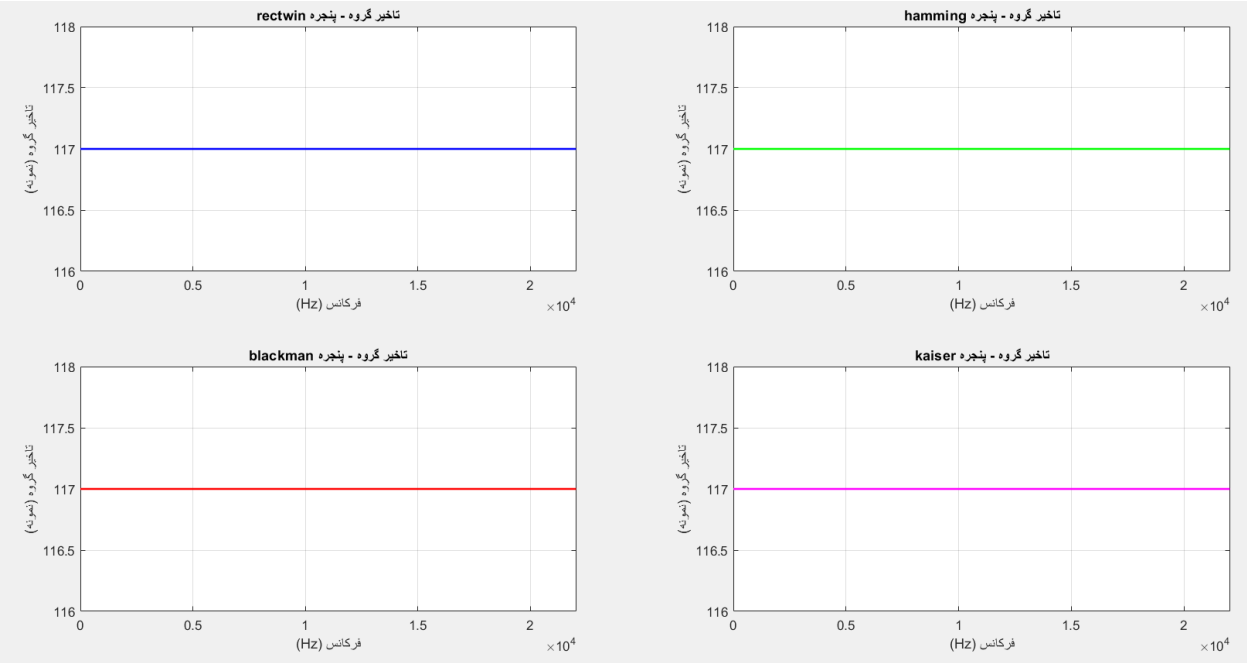
مراحل فعالیت:

خواندن فایل صوتی با دستور audioread

طراحی فیلتر FIR پایین گذر با پنجره گذاری های مختلف

محاسبه و رسم تاخیر گروه

اعمال فیلترها بر روی سیگنال صوتی و شنیدن نتایج آن



فیلترهای FIR با فاز خطی

تمام فیلترهایی که با fir1 و یک پنجره متقارن (حتی rectwin) ساخته می‌شوند، ضرایب متقارن دارند. این تقارن باعث می‌شود فیلتر فاز خطی داشته باشد. ویژگی اصلی فاز خطی این است که تأخیر گروه (Group Delay) ثابت و مستقل از فرکانس است و برابر با نصف مرتبه فیلتر (N/2) به‌دست می‌آید.

```
>> N

N =

   234

>> N/2

ans =

   117
```

تاخیر گروه: به معنی اینکه فرکانس های مختلف همزمان به گوش ما میرسند یا با اختلاف زمانی

فعالیت ۸: در مورد تبدیل فوریه زمان کوتاه (STFT) توضیح دهید. فعالیتی در مورد استفاده از دستور زیر طراحی نمایید.

$$spectrogram(x, window, noverlap, nfft, Fs, 'yaxix')$$

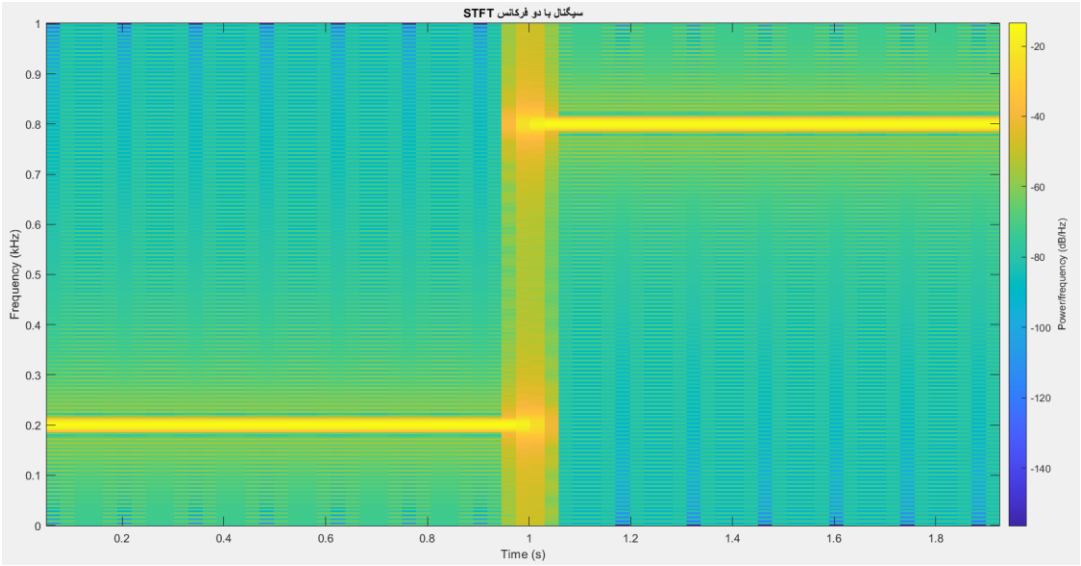
سیگنال ورودی — نوع پنجره و طولش — تعداد نقاط همپوشانی — تعداد نقاط fft — فرکانس نمونه‌برداری

STFT سیگنال را به بخش‌های کوتاه (پنجره‌دار) تقسیم کرده و از هر بخش DFT می‌گیرد. این کار امکان بررسی تغییرات فرکانسی در طول زمان را فراهم می‌کند. در MATLAB از دستور spectrogram استفاده می‌شود.

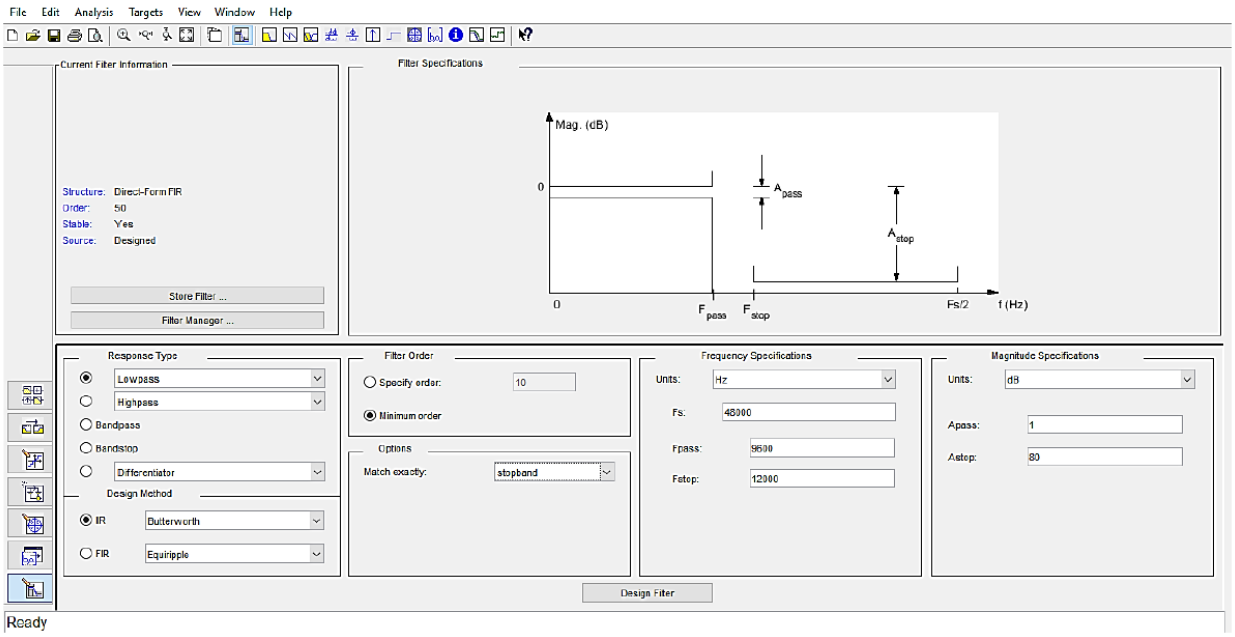
یعنی با استفاده از STFT امکان فهمیدن زمان هر فرکانس وجود دارد.

فعالیت طراحی شده: یک سیگنال شامل دو سینوسی با فرکانس‌های ۲۰۰ هرتز و ۸۰۰ هرتز که در زمان‌های متفاوت فعال می‌شوند تولید کنید. با استفاده از spectrogram نمایش زمان – فرکانس را با استفاده از پنجره گذاری همینگ رسم کنید.

```
fs = 2000;
t = 0:1/fs:2;
x = [sin(2*pi*200*t(t<1)), sin(2*pi*800*t(t>=1))];
figure;
spectrogram(x, hamming(256), 200, 512, fs, 'yaxis');
title('STFT سیگنال با دو فرکانس');
```



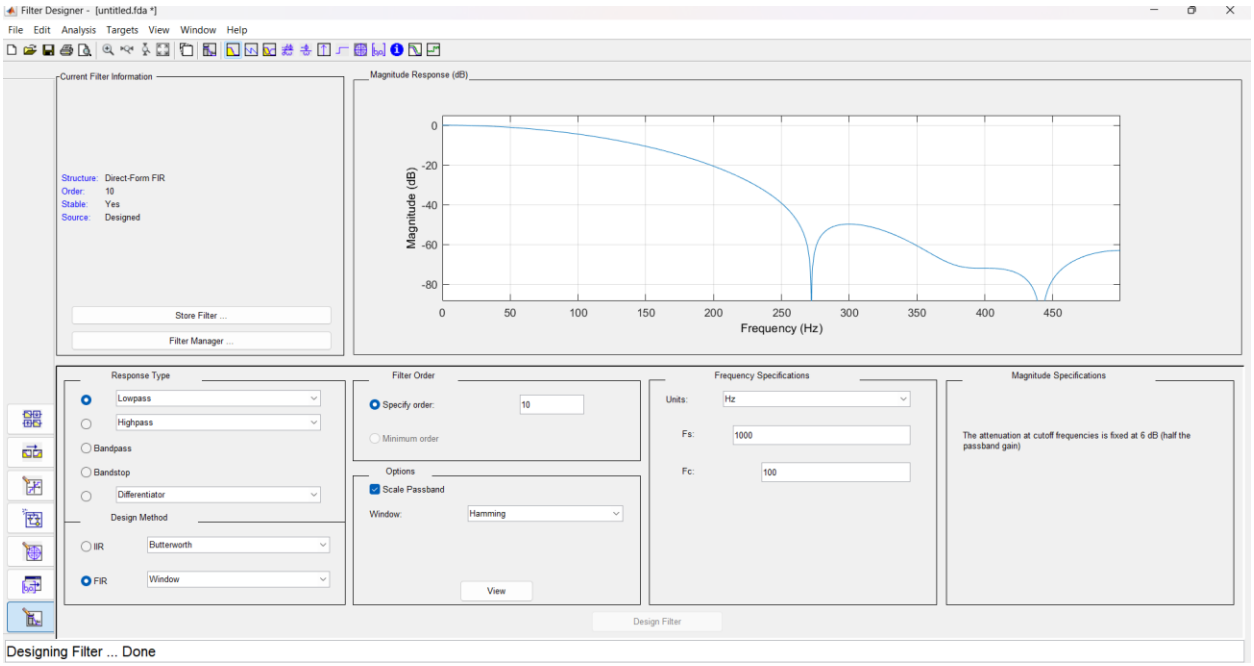
بخش ۲: استفاده از تولباکس filter Designer با fdatool



فعالیت ۹: الف) در FDATool یک فیلتر پایین‌گذر با مشخصات زیر طراحی کنید:

- فرکانس نمونه‌برداری: ۱۰۰۰ هرتز
- فرکانس قطع: ۱۰۰ هرتز
- نوع فیلتر: FIR (با روش طراحی Window)

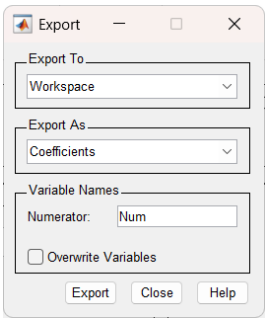
ب) در محیط متلب یک سیگنال تصادفی ایجاد کنید و فیلتر طراحی شده را اعمال کنید.



سمت چپ **Lowpass** رو انتخاب میکنیم و متد **FIR** رو انتخاب میکنیم.

از پنجره **Hamming** استفاده کردم و فرکانس نمونه برداری ۱۰۰۰ و فرکانس قطع ۱۰۰ (طبق فعالیت)

با انتخاب Export... Ctrl+E به فضای **Workspace** انتقال میدهیم.



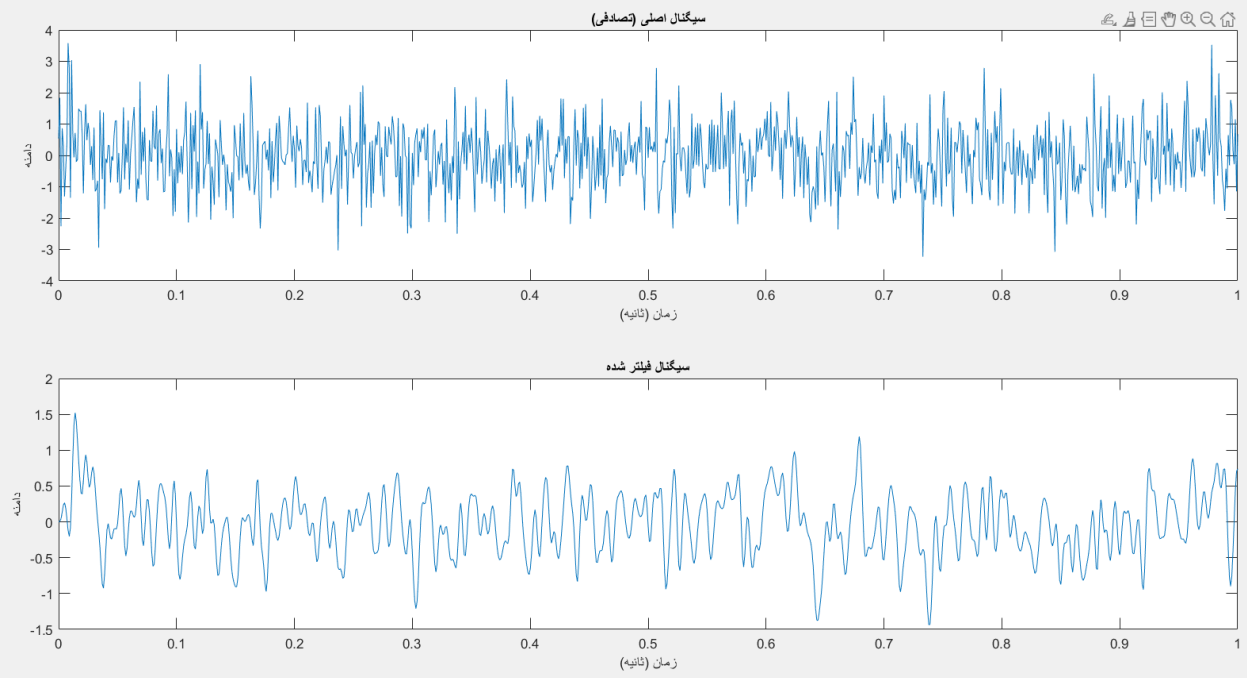
در نهایت ضرایب فیلتر به ورک اسپیس انتقال داده میشود و میتوانیم از آن استفاده کنیم.

Num

1x11 double

	1	2	3	4	5	6	7	8	9	10	11
1	7.3914e-19	0.0093	0.0476	0.1224	0.2022	0.2370	0.2022	0.1224	0.0476	0.0093	7.3914e-19

```
% Num -> fdatool: FIR, LOWPASS, HAMMING, Fs=1000, Fc=100
fs=1000;
t = 0:1/fs:1;

x = randn(size(t));

% اعمال فیلتر FIR
y = filter(Num, 1, x);

% رسم سیگنال ها
figure;
subplot(2,1,1);
plot(t, x);
title('سیگنال اصلی (تصادفی)');
xlabel('زمان (ثانیه)'); ylabel('دامنه');

subplot(2,1,2);
plot(t, y);
title('سیگنال فیلتر شده');
xlabel('زمان (ثانیه)'); ylabel('دامنه');
```

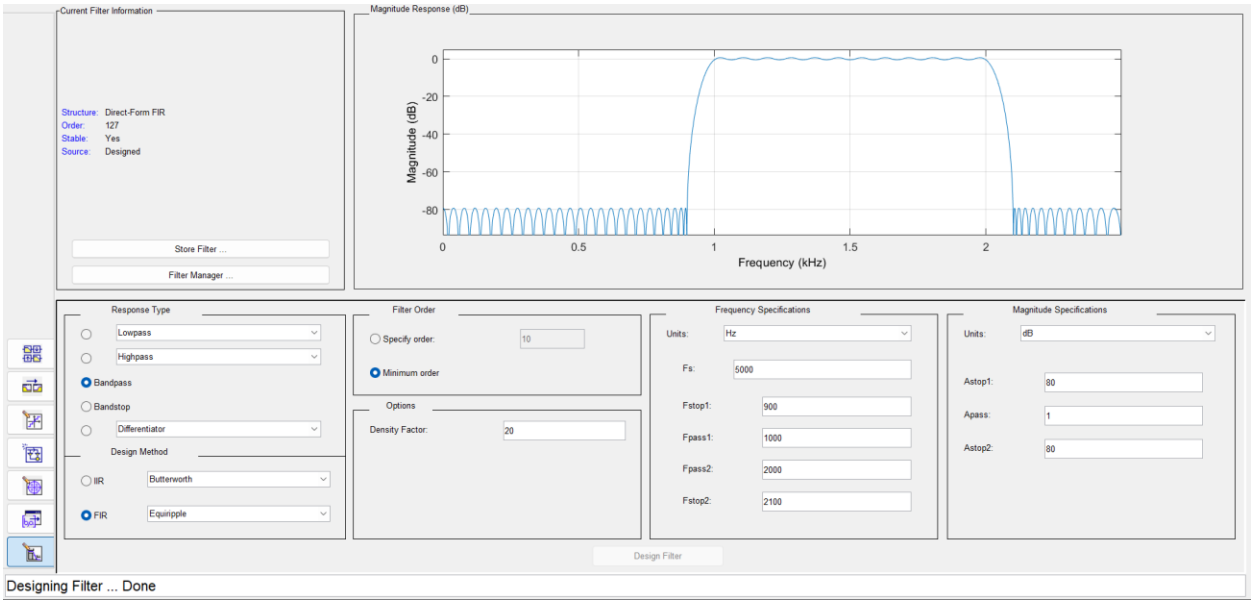


فعالیت ۱۰: الف) در FDATool یک فیلتر میانگذر با مشخصات زیر طراحی کنید:

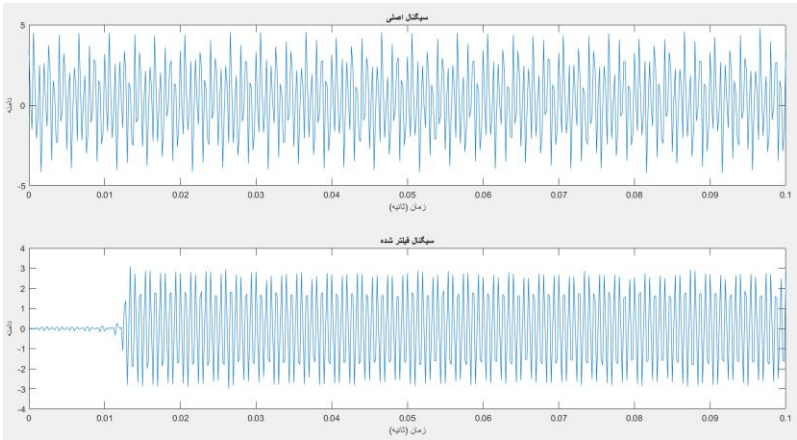
- فرکانس نمونه‌برداری: ۵۰۰۰ هرتز
- فرکانس‌های قطع: ۱۰۰۰ هرتز و ۲۰۰۰ هرتز
- نوع فیلتر: FIR (با روش Equiripple)

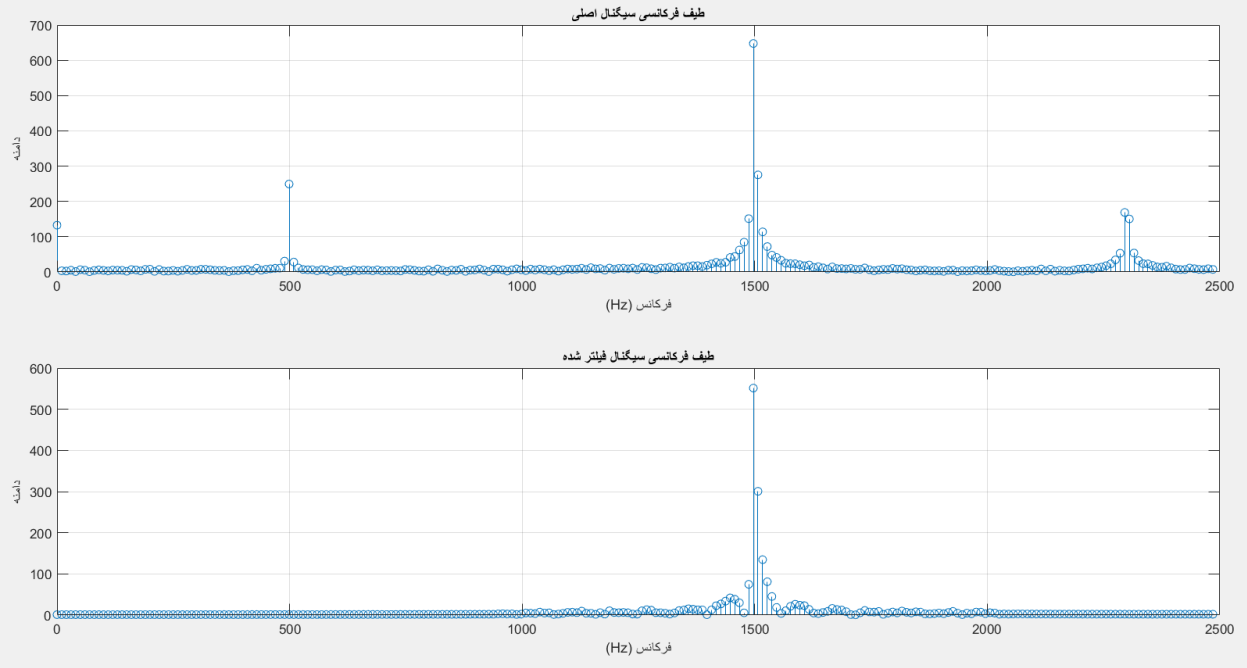
ب) یک سیگنال شامل نویز و چند فرکانس مختلف ایجاد کنید و فیلتر میان‌گذر را برای حذف نویز و بخش‌های غیر مطلوب سیگنال استفاده کنید.

فیلتر میان گذر طراحی شده:



```
fs=5000;
t = 0:1/fs:0.1;

f0 = 500;
f1 = 1500;
f2 = 2300;

noise = 0.5 * rand(size(t));
x = sin(2*pi*f0*t) + 3*cos(2*pi*f1*t) + sin(2*pi*f2*t) + noise;

% اعمال فیلتر FIR
y = filter(Num, 1, x);

% رسم سیگنال ها
figure;
subplot(2,1,1);
plot(t, x);
title('سیگنال اصلی');
xlabel('زمان (ثانیه)'); ylabel('دامنه');

subplot(2,1,2);
plot(t, y);
title('سیگنال فیلتر شده');
xlabel('زمان (ثانیه)'); ylabel('دامنه');

X = fft(x);
Y = fft(y);
N = length(x);
f = (0:N-1)*(fs/N);

% نمایش طیف
figure;
subplot(2,1,1);
stem(f(1:N/2), abs(X(1:N/2)));
xlabel('فرکانس (Hz)'); ylabel('دامنه');
title('طیف فرکانسی سیگنال اصلی');
grid on;

subplot(2,1,2);
stem(f(1:N/2), abs(Y(1:N/2)));
xlabel('فرکانس (Hz)'); ylabel('دامنه');
title('طیف فرکانسی سیگنال فیلتر شده');
grid on;
```





همانطور که در تصویر مشخص است فیلتر های خارجی از بازه ۱۰۰۰ تا ۲۰۰۰ هرتز هستند فیلتر شده اند.