

# سیگنال‌ها و سیستم‌ها

دکتر حمید حسن‌پور

# فصل ششم

مقدمه‌ای بر طراحی فیلترها

# فهرست مطالب فصل

۷- ۱- مقدمه

۷- ۲- سیستم‌های LTI و فیلترها

۷- ۳- فیلتر انتخابگر فرکانس

۷- ۴- تقریب فیلترها

۷- ۵- طراحی فیلتر با کمک نرم افزار Matlab

## مقدمه

- در بسیاری از مسائل، تغییر دامنه فرکانسی برخی از مولفه‌های فرکانسی و یا حتی حذف آن‌ها می‌تواند مفید باشد. فیلتر کردن سیگنال‌ها به این موضوع اشاره دارد.
- برخی از کاربردها
  - حذف نویز از صدا می‌تواند به کمک فیلتر کردن سیگنال صدا اتفاق بیفتد.
  - انتخاب یک باند فرکانسی در سیستم‌هایی مثل گیرنده‌ها و پردازشگرها
  - حذف یک فرکانس یا باند فرکانسی نامطلوب
  - کاهش توان نویز سیگنال‌ها
- فیلترها را می‌توان به دو گروه انتخابگر فرکانس (Frequency selective) و عمومی یا دلخواه (General) تقسیم نمود.

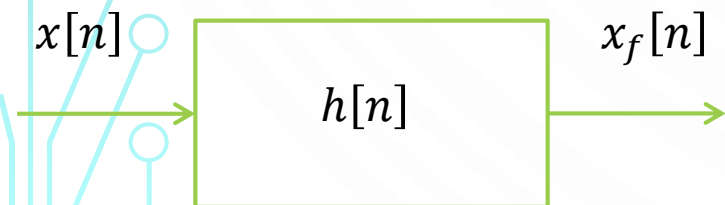
# سیستم‌های LTI و فیلترها

- سیستم‌های LTI می‌توانند به عنوان فیلتر عمل کنند.
- در شکل زیر سیگنال  $x[n]$  به وسیله‌ی سیستم با پاسخ ضربه  $h[n]$  فیلتر شده است.
- همانطور که می‌دانید سیگنال فیلتر شده  $x_f[n]$  از رابطه زیر محاسبه می‌شود.

$$x_f[n] = x[n] * h[n]$$

از آنجایی که هدف ما فیلتر کردن در حوزه فرکانس است، می‌توان نوشت:

$$x_f[n] = \mathfrak{F}^{-1}\{X_f[k]\} = \mathfrak{F}^{-1}\{X[k] \cdot H[k]\}$$



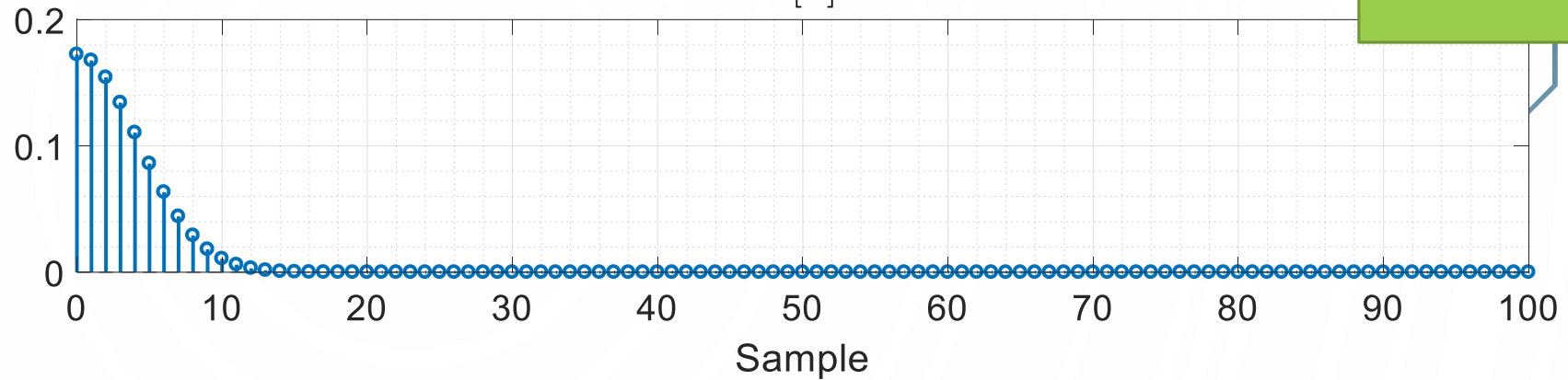
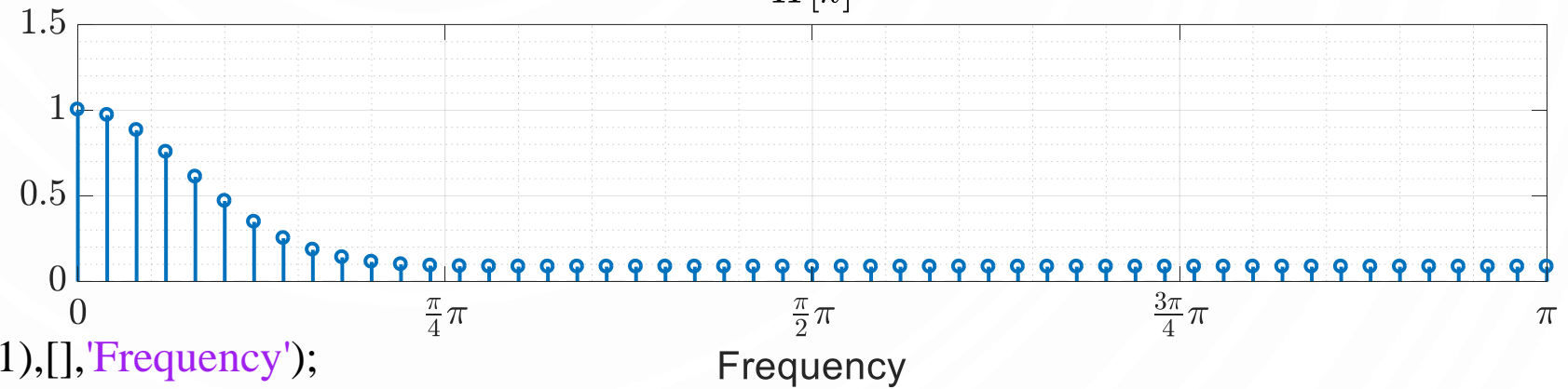
مثال: سیستم LTI با پاسخ ضرب  $h[n] = \frac{1}{5.8} e^{-\frac{n^2}{36}}, 0 \leq n \leq 100$  را در نظر بگیرید.

الف: پاسخ ضربه سیستم را در حوزه زمان و در حوزه فرکانس رسم نمایید.

ب: سیگنال  $x[n] = e^{-0.03n} + \sin(2n), 0 \leq n \leq 100$  را در نظر بگیرید. این سیگنال را در حوزه زمان و در حوزه فرکانس رسم نمایید.

ج: سیگنال  $x[n]$  را به کمک سیستم مد نظر فیلتر نمایید (سیگنال  $x_f[n]$ ). سپس آن را رسم نمایید.

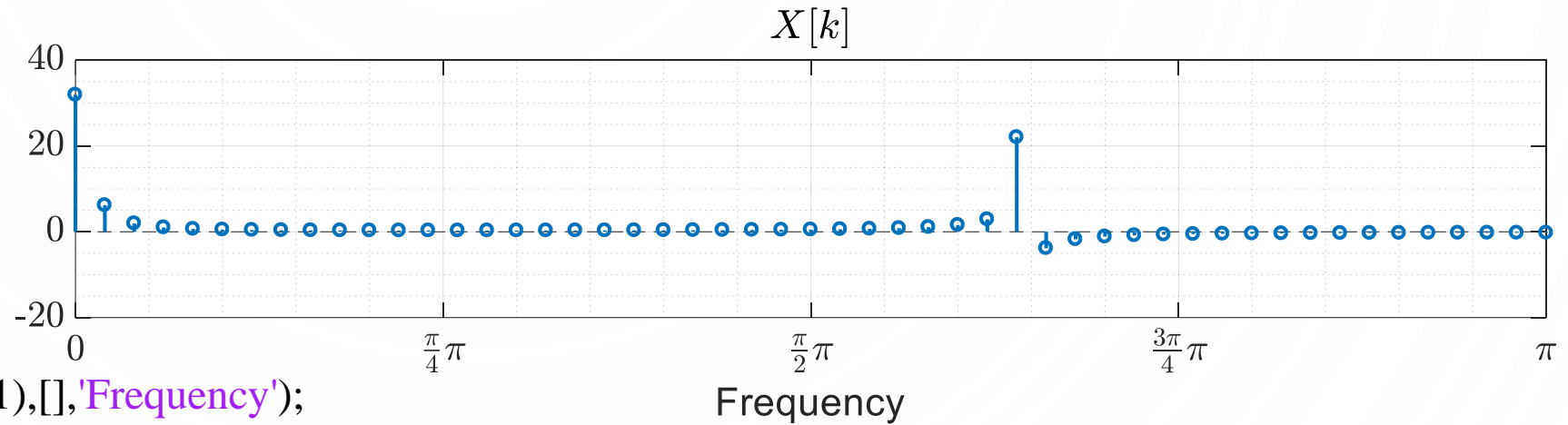
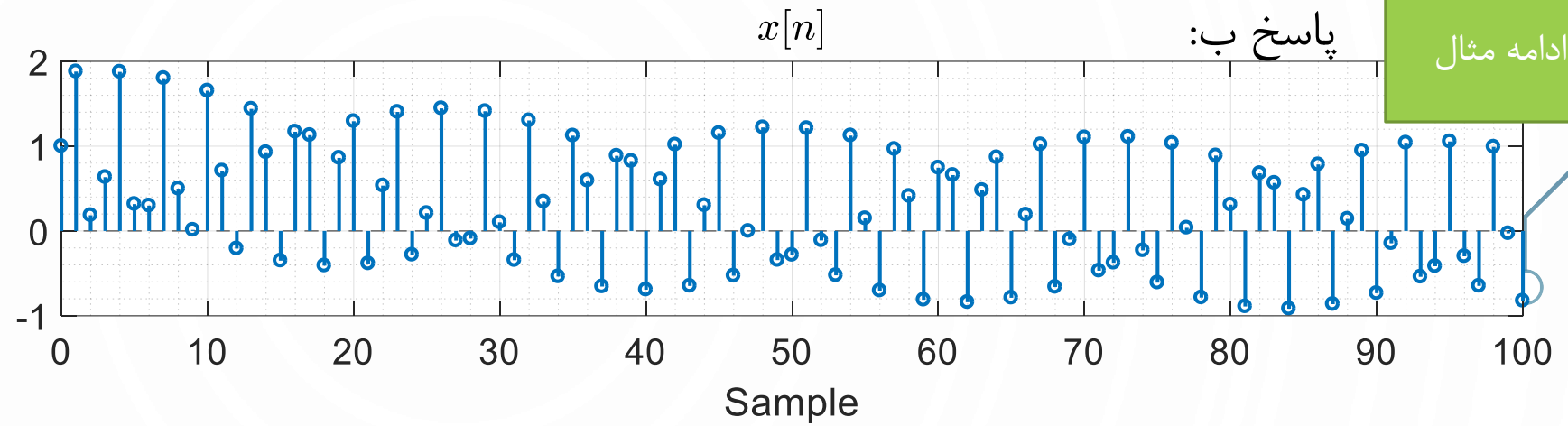
د: سیگنال‌های  $x[n]$ ،  $x_f[n]$  و  $x_1[n] = e^{-0.03n}$  را رسم نمایید. از این تصاویر چه نتیجه‌ای می‌گیرید.

$h[n]$  $H[k]$ 

```

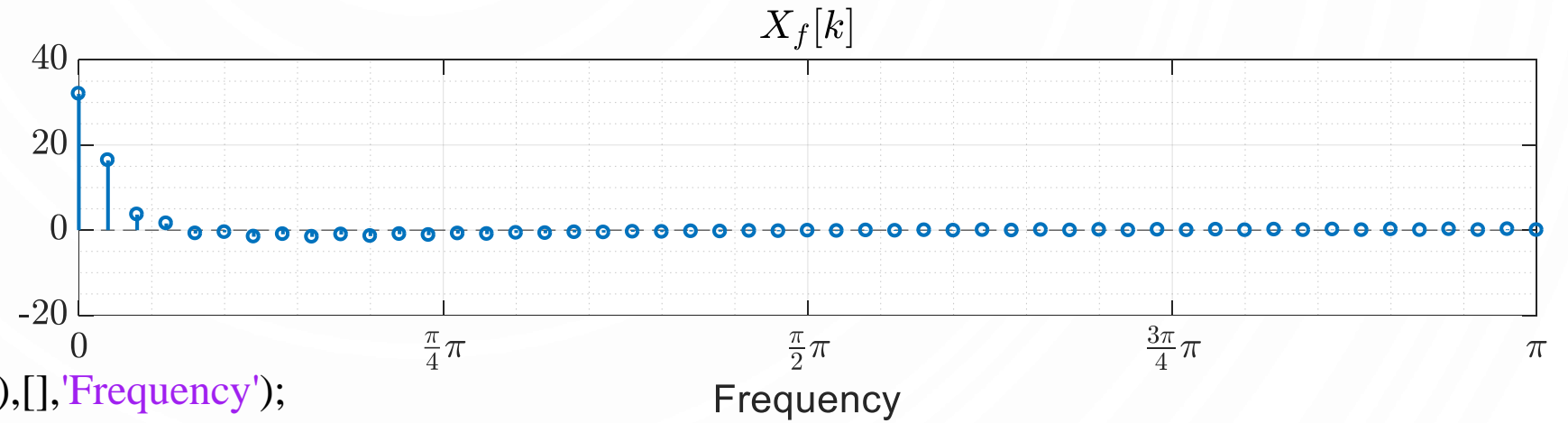
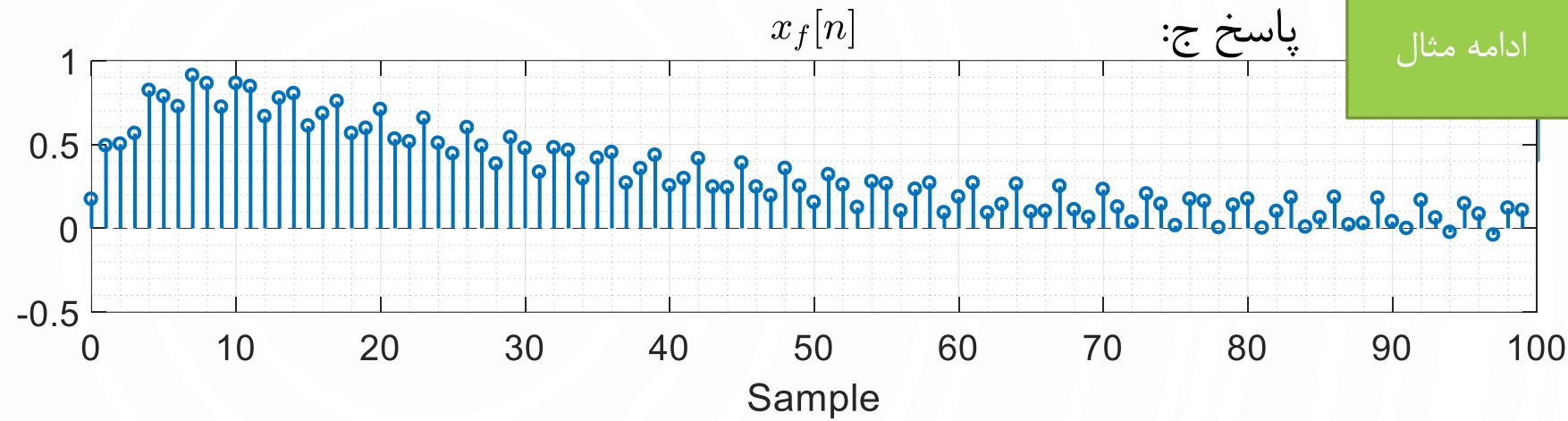
clc;
clear;
close all;
%% Ans A
n=0:100;
sigma=1;
A=1/5.8;
h=A*exp(-(n.^2)./36);
H=fft(h);
subplot(2,1,1)
SS_rep_dt(0:length(h)-1,h,[]);
title('$h[n]$', 'Interpreter', 'latex');
subplot(2,1,2)
SS_rep_dt((0:50)/100*2*pi,H(1:51),[], 'Frequency');
title('$H[k]$', 'Interpreter', 'latex');
ax=gca;
ax.XTick=0:pi/4:pi;
ax.XTickLabel={'$0$' '$\frac{\pi}{4}$' '$\frac{\pi}{2}$' '$\frac{3\pi}{4}$' '$\pi$'};
ax.TickLabelInterpreter='latex';
xlim([0 pi]);

```



%% Ans B

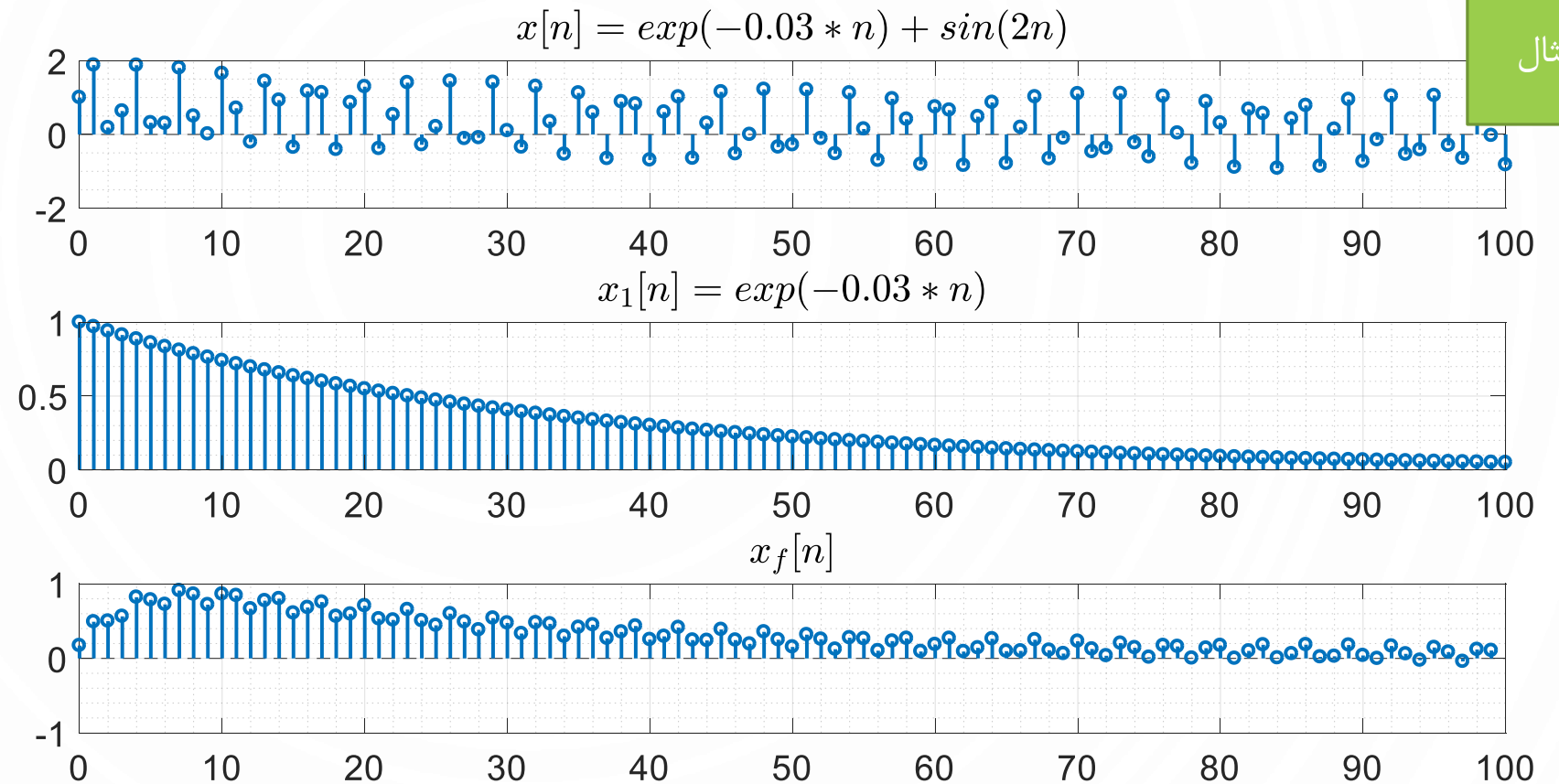
```
figure;
x=exp(-0.03*n)+sin(2*n);
X=fft(x);
subplot(2,1,1)
SS_rep_dt((0:100),x,[]);
title('$x[n]$', 'Interpreter', 'latex');
subplot(2,1,2)
SS_rep_dt((0:50)/100*2*pi,X(1:51),[], 'Frequency');
title('$X[k]$', 'Interpreter', 'latex');
ax=gca;
ax.XTick=0:pi/4:pi;
ax.XTickLabel={'0$' '$\frac{\pi}{4}\pi$' '$\frac{\pi}{2}\pi$' '$\frac{3\pi}{4}\pi$' '$\pi$'};
ax.TickLabelInterpreter='latex';
xlim([0 pi]);
```



%% Ans C

```
figure;
x_f=conv(x,h);
X_f=fft(x_f);
subplot(2,1,1)
SS_rep_dt(0:99,x_f(1:100),[]);
title('$x_f[n]$', 'Interpreter', 'latex');
subplot(2,1,2)
SS_rep_dt((0:50)/100*2*pi,X_f(1:51),[], 'Frequency');
title('$X_f[k]$', 'Interpreter', 'latex');
ax=gca;
ax.XTick=0:pi/4:pi;
ax.XTickLabel={'$0$' '$\frac{\pi}{4}\pi$' ...
'$\frac{\pi}{2}\pi$' '$\frac{3\pi}{4}\pi$' '$\pi$'};
ax.TickLabelInterpreter='latex';
xlim([0 pi]);
```

پاسخ د:



%%

```
figure;
subplot(3,1,1)
SS_rep_dt((0:100),x,[],"");
title('$x[n]=\exp(-0.03*n)+\sin(2n)$','Interpreter','latex');
subplot(3,1,2)
x_1=\exp(-0.03*n);
SS_rep_dt((0:100),x_1,[],"");
title('$x_1[n]=\exp(-0.03*n)$','Interpreter','latex');
subplot(3,1,3)
SS_rep_dt(0:99,x_f(1:100),[],"");
title('$x_f[n]$','Interpreter','latex');
```

با اعمال فیلتر بر روی سیگنال  $x[n]$ ، مولفه سینوسی آن تا حدی حذف گردید و مولفه دیگر باقی ماند.

# فیلتر انتخابگر فرکانس ( FREQUENCY-SELECTIVE FILTER )

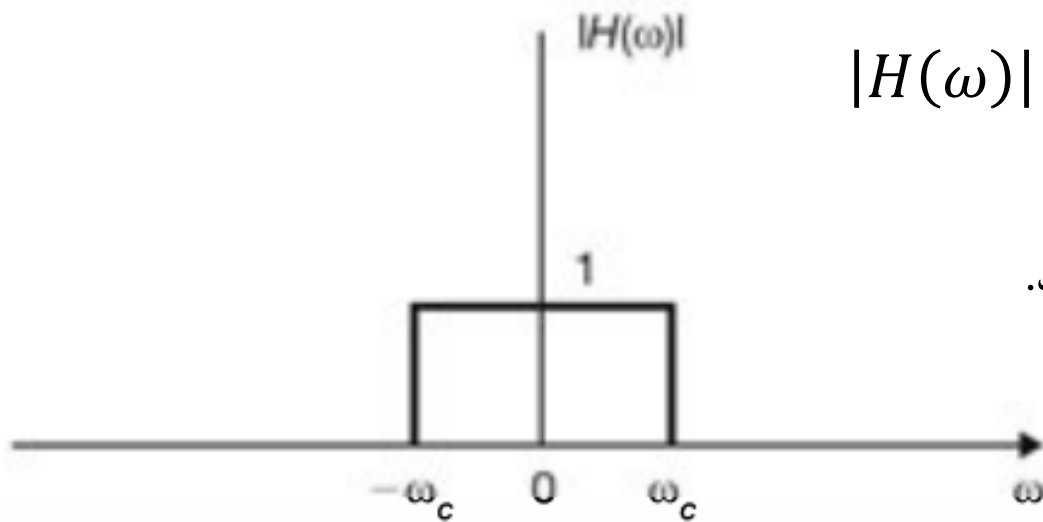
- یکی از انواع فیلترها، فیلتر انتخابگر فرکانس است. در این فیلتر، مولفه‌های یک محدوده فرکانسی بدون تغییر باقی می‌مانند. همچنین سایر مولفه‌های فرکانسی، توسط این فیلتر حذف می‌شوند.
- معروف‌ترین فیلترهای ایده‌آل در این خانواده به صورت زیر هستند.
  - ✓ الف: فیلتر پایین‌گذر ایده‌آل (Ideal Low-Pass Filter)
  - ✓ ب: فیلتر بالاگذر ایده‌آل (Ideal High-Pass Filter)
  - ✓ ج: فیلتر میان‌گذر ایده‌آل (Ideal Bandpass Filter)
  - ✓ د: فیلتر میان‌نگذر ایده‌آل (Ideal Bandstop Filter)

# الف: فیلتر پایین‌گذر ایده‌آل (IDEAL LOW-PASS FILTER)

- تابع تبدیل یک فیلتر پایین‌گذر ایده‌آل به صورت زیر است.

$$|H(\omega)| = \begin{cases} 1 & |\omega| \leq \omega_c \\ 0 & |\omega| > \omega_c \end{cases}$$

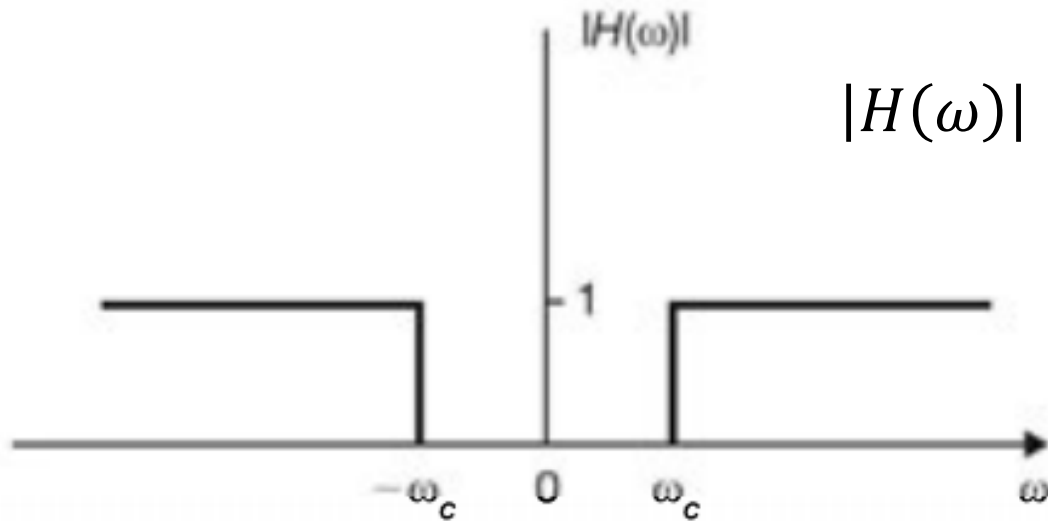
- به  $\omega_c$  فرکانس قطع (cutoff frequency) گفته می‌شود.



## ب: فیلتر بالاگذر ایده آل (IDEAL HIGH-PASS FILTER)

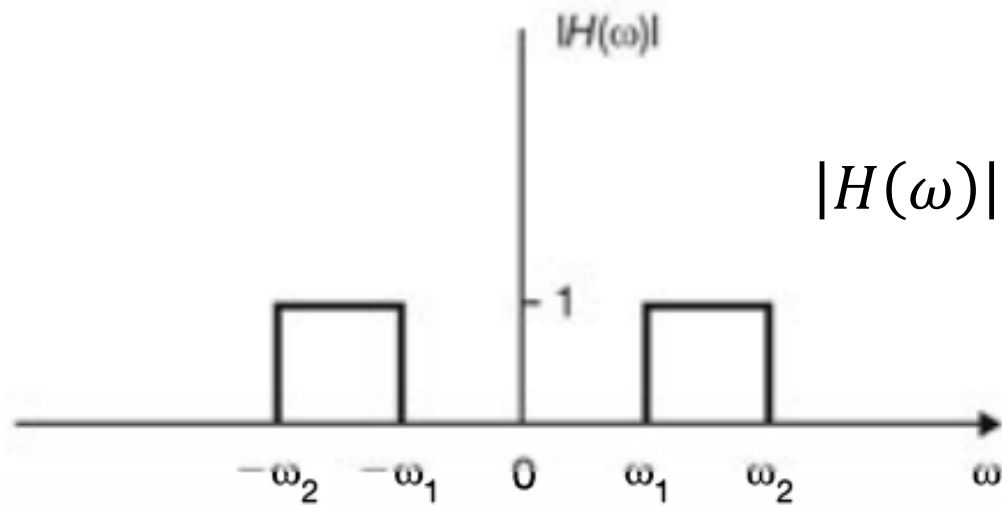
- تابع تبدیل یک فیلتر بالاگذر ایده آل به صورت زیر است.

$$|H(\omega)| = \begin{cases} 0 & |\omega| \leq \omega_c \\ 1 & |\omega| > \omega_c \end{cases}$$



## ج: فیلتر میان‌گذر ایده‌آل (IDEAL BANDPASS FILTER)

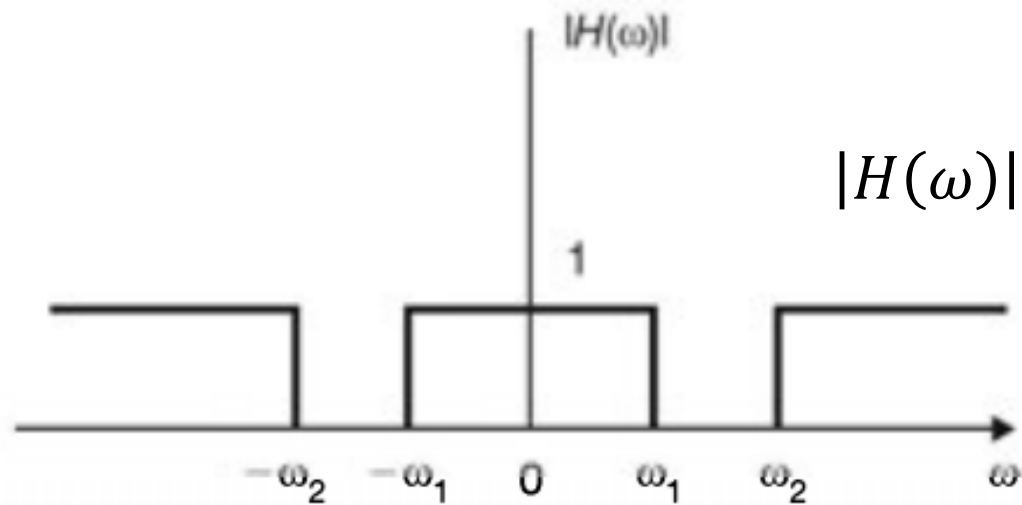
- تابع تبدیل یک فیلتر میان‌گذر ایده‌آل به صورت زیر است.



$$|H(\omega)| = \begin{cases} 1 & \omega_1 < \omega < \omega_2 \\ 0 & \text{در غیر این صورت} \end{cases}$$

## د: فیلتر میان‌گذر ایده‌آل (IDEAL BANDSTOP FILTER)

- تابع تبدیل یک فیلتر میان‌گذر ایده‌آل به صورت زیر است.



$$|H(\omega)| = \begin{cases} 0 & \omega_1 < \omega < \omega_2 \\ 1 & \text{در غیر این صورت} \end{cases}$$

# تقریب فیلترها

- تقریبا تمامی فیلترها در حالت ایده‌آل خود غیر قابل تحقق می‌باشند که علت آن غیر علی (**noncasual**) بودن آن‌ها است. به عنوان مثال پاسخ ضربه فیلتر ایده‌آل پایین‌گذر چنین می‌باشد.

$$h(t) = \frac{\omega_c}{\pi} \text{sinc} \left( \frac{\omega_c}{\pi} t \right)$$

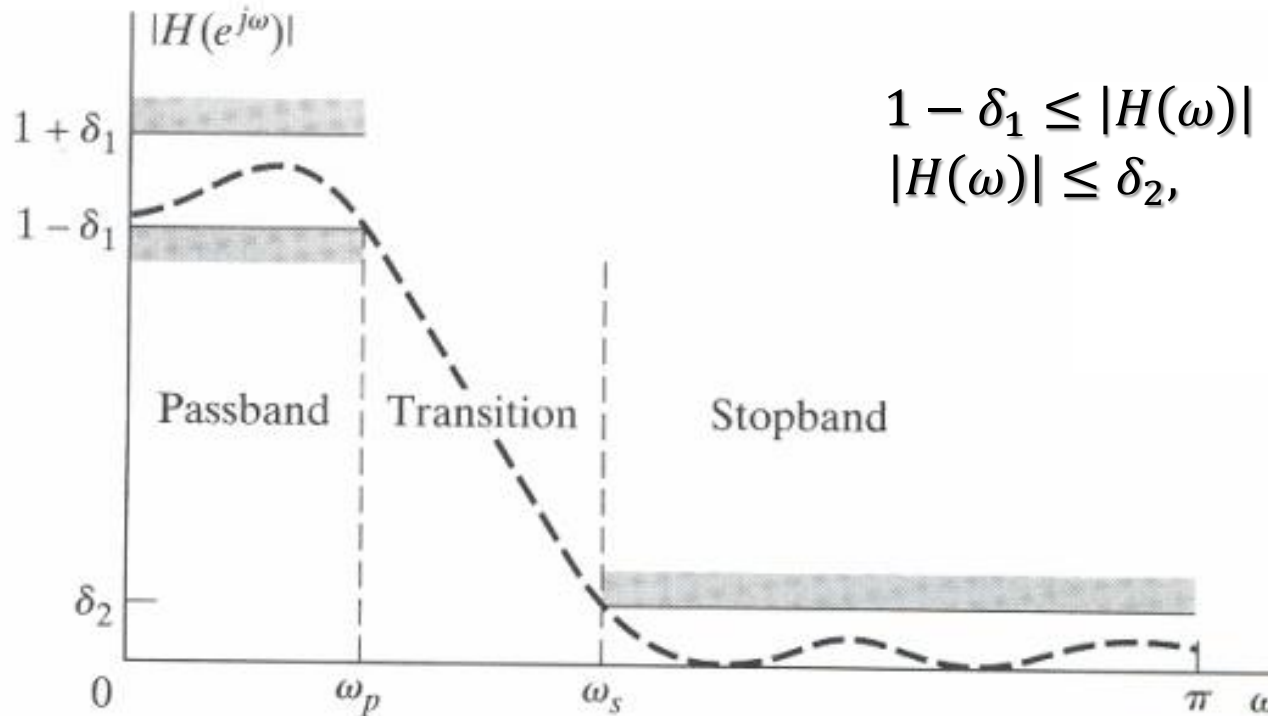
- جهت رفع این اشکال، از تقریب فیلترها (Filter Approximation) استفاده می‌شود.
- هدف از تقریب فیلترها، تقریب زدن مشخصه ایده‌آل آن‌ها توسط تابعی پیوسته بر حسب فرکانس است بطوری که تابع تبدیل مربوطه قابل تحقق باشد.

# تقریب فیلترهای انتخابگر فرکانس

- برای تقریب مشخصه ی دامنه ی فیلترهای انتخاب گر فرکانس دو مرحله زیر را در نظر بگیرید.
  - ✓ برای آنها یک دیواره ی تضعیف تعریف می شود که دارای سه ناحیه عبور (Passband)، قطع (Stopband) و گذر (Transition Band) است.
  - ✓ پس از تعریف دیواره ی تضعیف، از توابع فرکانسی خاصی جهت تقریب دامنه ی فیلترها استفاده می شود، بطوریکه تمامی این توابع در داخل محدوده مجاز دیواره تضعیف قرار داشته باشند.

# تقریب فیلترهای انتخابگر فرکانس (ادامه)

- در عمل طراحی فیلترهای ایده‌آل ممکن نیست. به طور مثال، فیلتر پایین‌گذر را می‌توان به صورت زیر مدل کرد.



$$1 - \delta_1 \leq |H(\omega)| \leq 1 + \delta_1, \\ |H(\omega)| \leq \delta_2,$$

$$|\omega| \leq \omega_p \\ |\omega| \geq \omega_s$$

# طراحی فیلتر با کمک نرم افزار MATLAB

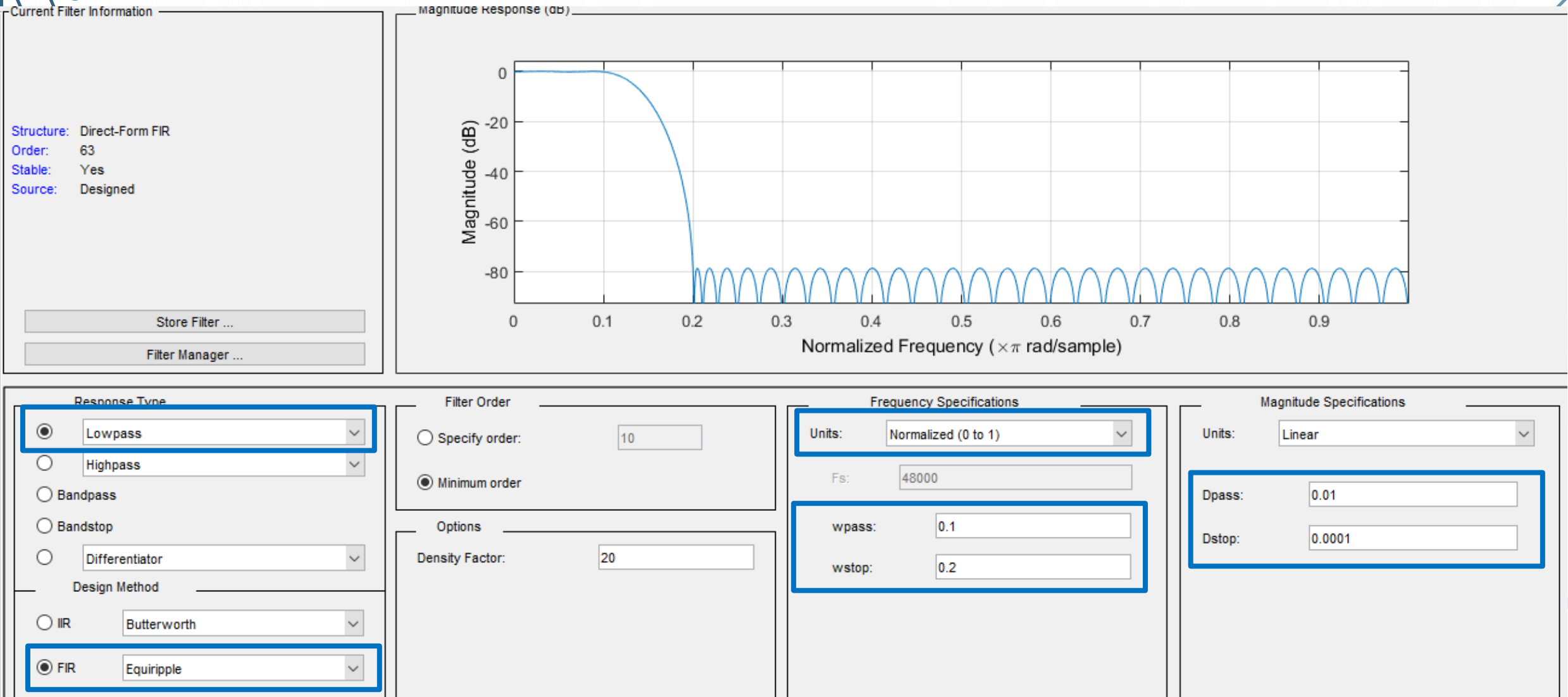
• برای اینکار از ابزار filterDesigner استفاده می کنیم.

The screenshot displays the MATLAB Filter Designer window. The top menu bar includes File, Edit, Analysis, Targets, View, Window, and Help. The main workspace is divided into several sections:

- Current Filter Information:** Shows the filter structure as Direct-Form FIR, order 50, stable, and designed.
- Filter Specifications:** A plot of Magnitude (dB) versus frequency  $f$  (Hz). The plot shows a lowpass filter response with a passband edge at  $F_{pass}$  and a stopband edge at  $F_{stop}$ . The passband ripple is  $A_{pass}$  and the stopband attenuation is  $A_{stop}$ . The sampling frequency  $F_s/2$  is also indicated.
- Response Type:** Lowpass (selected), Highpass, Bandpass, Bandstop, Differentiator.
- Design Method:** IIR (Butterworth), FIR (Equiripple, selected).
- Filter Order:** Specify order (10) or Minimum order (selected).
- Options:** Density Factor (20).
- Frequency Specifications:** Units (Hz),  $F_s$  (48000),  $F_{pass}$  (9600),  $F_{stop}$  (12000).
- Magnitude Specifications:** Units (dB),  $A_{pass}$  (1),  $A_{stop}$  (80).

Buttons for "Store Filter ..." and "Filter Manager ..." are visible. The status bar at the bottom indicates "Ready".

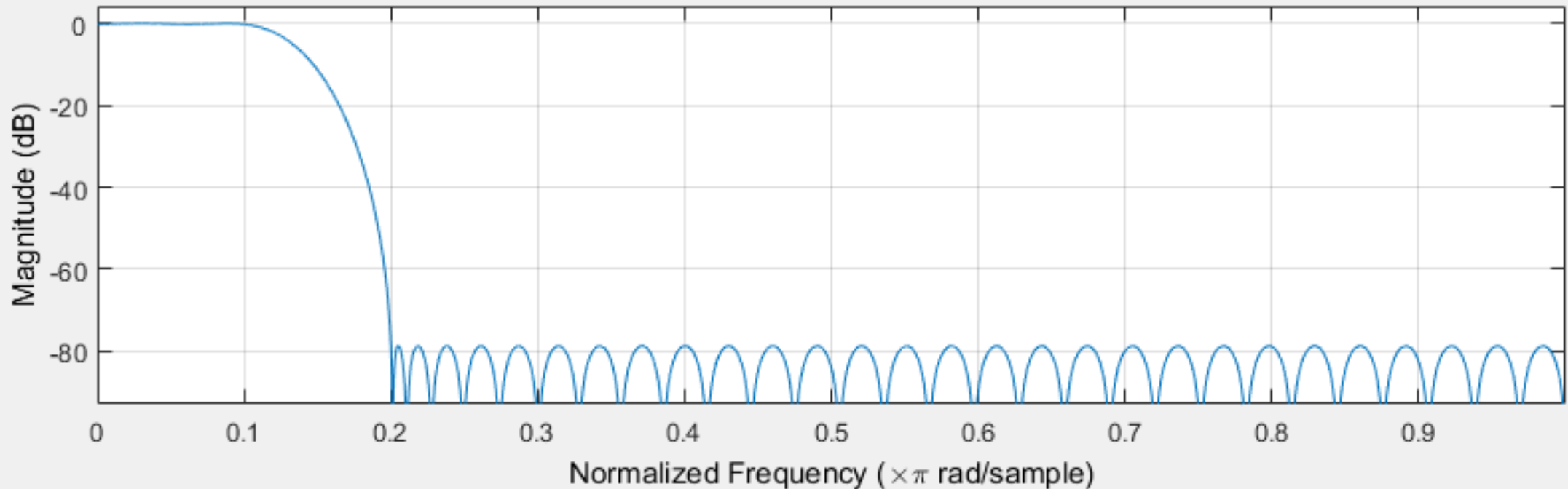
- فرض کنید میخواهیم فیلتری پایین‌گذر با فرکانس گذر و قطع  $0.1\pi, 0.2\pi$  طراحی کنیم. پس از طراحی از منوی *File*، با کمک گزینه *Export* پاسخ ضربه سیگنال را ذخیره می‌نماییم.



- در نمایش طیف فرکانسی فیلتر، از  $dB$  استفاده شده است. برای تبدیل یک عدد به  $dB$ ، ده برابر لگاریتم ده آن را محاسبه می‌کنیم.

$$|H(\omega)|_{dB} = 10 \log_{10}(|H(\omega)|)$$

- این نمایش باعث می‌شود اعداد نزدیک به صفر بهتر نمایش داده شوند.



```

clc; clear; close all;
[OriginalSig, fs]=audioread('OriginalVoice.wav');
[noisySig, fs]=audioread('NoisyVoice.wav');
load Filter.mat
%%
filtered_sig=conv(noisySig,Filter1);
audiowrite('FilteredVoice.wav',filtered_sig,fs);
%%
OriginalSig=fft(originalSig,fs);
NoisySig=fft(noisySig,fs);
Filtered_sig=fft(filtered_sig,fs);

figure;
subplot(221)
plot(abs(OriginalSig(1:end/2)));
title('Original Signal');
subplot(222)
plot(abs(NoisySig(1:end/2)));
title('Noisy Signal');
subplot(223)
plot(abs(Filtered_sig(1:end/2)));
title('Filtered Signal');

```

مثال: یک سیگنال صوتی آلوده به نویز در اختیار شما قرار دارد. برای رفع نویز فیلتری پایین‌گذر با فرکانس گذر و قطع  $3000\text{Hz}$  و  $3500\text{Hz}$  طراحی کنید. سپس سیگنال را به کمک آن فیلتر نمایید.

سیگنال ذخیره شده را مجدداً به صورت صدا ذخیره نمایید.

