به نام خدا

مصطفی چرکزی

آزمایش سوم پردازش سیگنال های دیجیتال

آزمایش ۳: سیگنال‌های زمان گسسته در حوزه فرکانس

در این آزمایش به بررسی خواص سیگنال‌های زمان گسسته از منظر فرکانس می‌پردازیم. به همین منظور، سه نواع نمایش معروف و پرکاربرد تبدیل فوریه گسسته (DTFT)، تبدیل فوریه سریع (FFT) و تبدیل Z استفاده خواهد شد و ویژگی‌های آن‌ها بررسی خواهند شد.

برخی توابع متلی مورد استفاده عبارتند از:

Elementary matrices and matrix manipulation: ماتریس‌های اولیه و به کار بردن ماتریس:

fliplr	i	pi	zeros	:

Elementary functions: توابع اولیه:

abs	angle	conj	exp	imag	real
rem					

Polynomial and interpolation functions: توابع چندجمله‌ای و تعاملی :

conv

Data analysis and fourier transform functions: آنالیز داده و توابع تبدیل فوریه:

fft	ifft	max	min

Signal processing toolbox: جعبه ابزار پردازش سیگنال:

freqz	impz	residuez	Tf2zp	Zp2sos	Zp2tf
zplane					

.^ , ^

$fliplr \Rightarrow flip\ rotate\ (left\ to\ right)$

$rem \Rightarrow$ باقیمانده

$fft \quad , \quad ifft$

$freqz \Rightarrow$ محاسبه‌ی پاسخ فرکانسی تابع تبدیل

$[h, w] = freqz(b, a, n)$

پاسخ فرکانسی

فرکانس زاویه‌ای

طول پاسخ فرکانسی

ضرایب صورت(num)

ضرایب مخرج(den)

$$[z\,,p,k] = tf2zp \quad , \quad zp2tf$$

$$H(s) = \frac{B(s)}{A(s)} = \frac{b_1 s^{n-1} + \cdots + b_{n-1}s + b_n}{a_1 s^{m-1} + \cdots + a_{m-1}s + a_m} \longrightarrow H(s) = \frac{Z(s)}{P(s)} = k\frac{(s-z_1)(s-z_2)\cdots(s-z_m)}{(s-p_1)(s-p_2)\cdots(s-p_n)}$$

$$zp2sos$$

$$H(z) = k\frac{(z-z_1)(z-z_2)\cdots(z-z_n)}{(z-p_1)(z-p_2)\cdots(z-p_m)} \longrightarrow H(z) = g\prod_{k=1}^{L} H_k(z) = g\prod_{k=1}^{L}\frac{b_{0k} + b_{1k}z^{-1} + b_{2k}z^{-2}}{1 + a_{1k}z^{-1} + a_{2k}z^{-2}}$$

$$\text{sos} = \begin{bmatrix} b_{01} & b_{11} & b_{21} & 1 & a_{11} & a_{21} \\ b_{02} & b_{12} & b_{22} & 1 & a_{12} & a_{22} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ b_{0L} & b_{1L} & b_{2L} & 1 & a_{1L} & a_{2L} \end{bmatrix}$$

بخش ا: محاسبه‌ی تبدیل فوریه گسسته DTFT

میدانیم که تبدیل فوریه سیگنال گسسته، تابعی پیوسته در حوزه فرکانس است. روابط مربوط به تبدیل فوریه گسسته و معکوس آن عبارتند از:

$$X(\omega) = \sum_{n=-\infty}^{n=\infty} x[n]e^{-j\omega n}$$

$$x[n] = \frac{1}{2\pi}\int_{\omega=-\pi}^{\omega=\pi} X(\omega)e^{j\omega n}$$

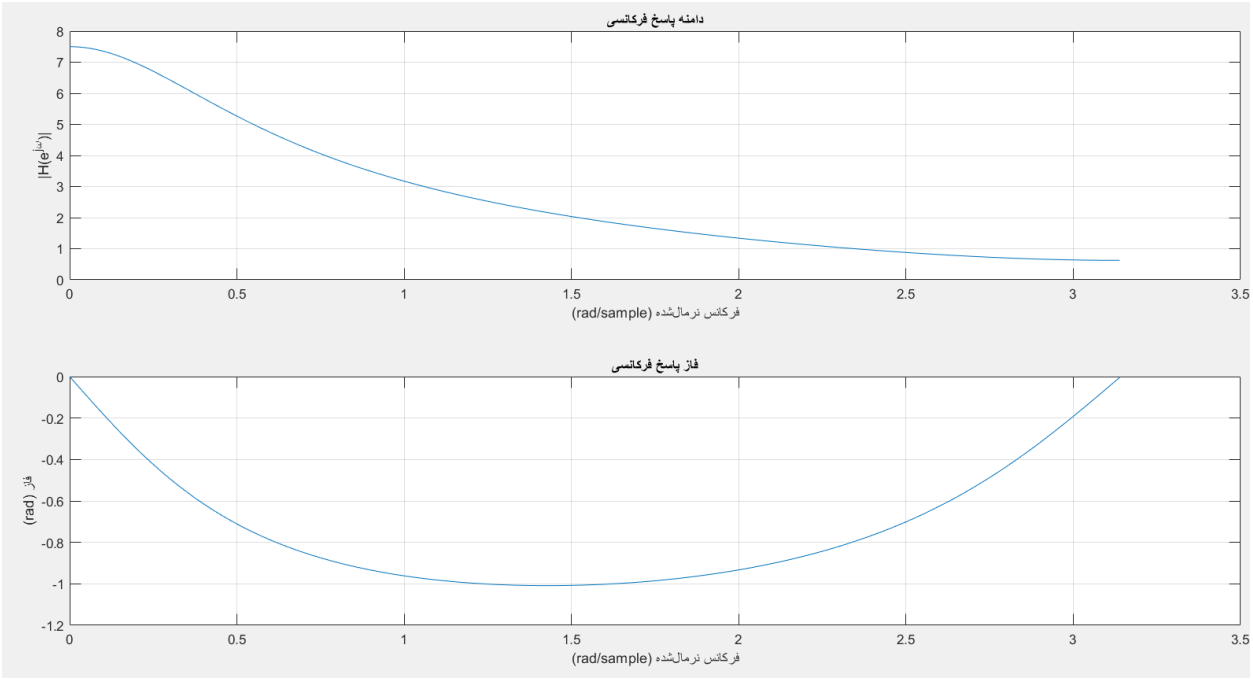
به دو دلیل امکان پیاده‌سازی تبدیل فوریه گسسته در متلب وجود ندارد. دلیل اول این است که در رابطه تبدیل فوریه طول سیگنال نامحدود است لذا در کامپیوتر قابل نمایش و محاسبه نیست. دلیل دوم اینکه تبدیل فوریه‌ی گسسته تابعی پیوسته از ω است، اما در متلب مقادیر بصورت گسسته ذخیره شده و نمایش داده میشوند. بنابراین برای محاسبه تبدیل فوریه‌ی گسسته میتوانیم از دستور **freqz** استفاده نمود و در ضرایب فرکانسی گسسته، مقادیر تبدیل فوریه را مشاهده کنیم.

فعالیت ۱: پاسخ فرکانسی تابع تبدیلی با ضرایب زیر را محاسبه کنید.

$$num = [2\ 1] \quad den = [1 \quad -0.6]$$

دامنه و فاز تبدیل فوریه را جداگانه محاسبه کنید. آیا تبدیل فوریه متناوب است؟ اگر بله دوره تناوب آن چیست؟ قسمت‌های نمایش داده شده از لحاظ تقارن چگونه هستند؟

```
num = [2 1];
den = [1 -0.6];
[H, w] = freqz(num, den, 1024);

figure;
subplot(2,1,1);
plot(w, abs(H));
title('دامنه پاسخ فرکانسی');
xlabel('فرکانس نرمال‌شده (rad/sample)');
ylabel('|H(e^{j\omega})|');
grid on;

subplot(2,1,2);
plot(w, angle(H));
title('فاز پاسخ فرکانسی');
xlabel('فرکانس نرمال‌شده (rad/sample)');
ylabel('فاز (rad)');
grid on;
```



آیا تبدیل فوریه متناوب است؟

بله، پاسخ فرکانسی سیستم‌های گسسته همواره متناوب با دوره تناوب 2π است.

دوره تناوب آن چیست؟

2π رادیان بر نمونه.

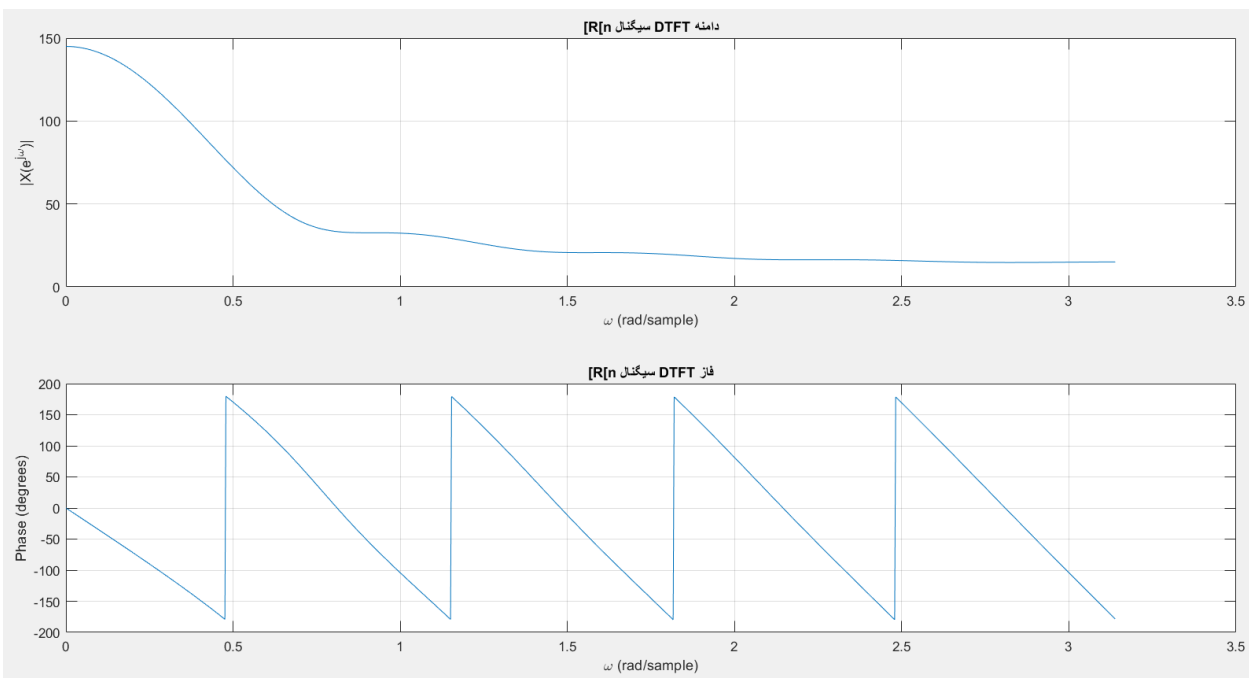
قسمت‌های نمایش داده شده از لحاظ تقارن چگونه هستند؟

دامنه، تقارن زوج و فاز، تقارن فرد نسبت به $\omega = 0$ و $\omega = \pi$ دارد.

فعالیت ۲: برنامه را طوری تغییر دهید که پاسخ فرکانسی سیگنال محدود زیر را نمایش دهد. همچنین فاز را بصورت درجه محاسبه کرده و نمایش دهید.

$$R[n] = [1\ 4\ 7\ 10\ 13\ 16\ 19\ 22\ 25\ 28]$$

```
clear; close all; clc;
R = [1 4 7 10 13 16 19 22 25 28];

[H, w] = freqz(R, 1, 1024);

figure;
subplot(2,1,1);
plot(w, abs(H));
title('دامنه DTFT سیگنال R[n]');
xlabel('\omega (rad/sample)');
ylabel('|X(e^{j\omega})|');
grid on;

subplot(2,1,2);
plot(w, rad2deg(angle(H))); % فاز بر حسب درجه
title('فاز DTFT سیگنال R[n]');
xlabel('\omega (rad/sample)');
ylabel('Phase (degrees)');
grid on;
```

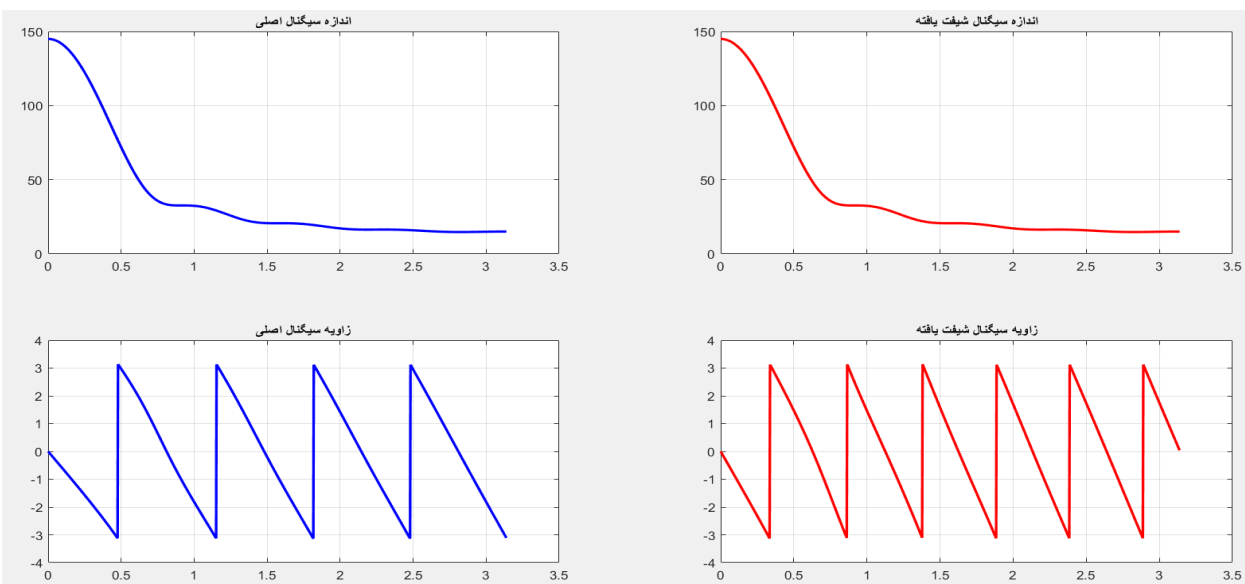


بخش ۲: شیفت زمانی تبدیل فوریه گسسته

$$x[n-n_0] \quad \overset{F}{\Rightarrow} \quad e^{-jn_0\omega}X(e^{j\omega})$$

فعالیت ۳: پاسخ فرکانسی سیگنال شیفت یافته فعالیت ۲ را محاسبه کنید. شیفت زمانی چه تاثیری در تبدیل فوریه سیگنال داشته است؟

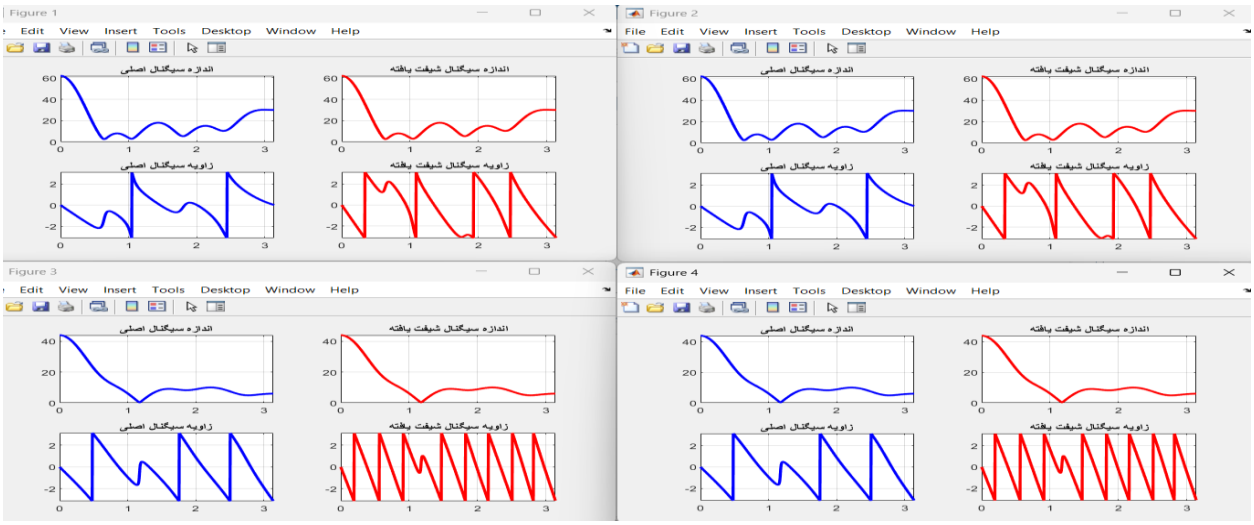
```
R = [1 4 7 10 13 16 19 22 25 28];
R_shift = [zeros(1,3) R]; % شیفت به اندازه 3 نمونه

[H_org, w] = freqz(R, 1, 1024);
[H_shift, w_shift] = freqz(R_shift, 1, 1024);

figure;
subplot(2,2,1);
plot(w, abs(H_org), 'b', 'LineWidth', 2);
title('اندازه سیگنال اصلی');
grid on;

subplot(2,2,2);
plot(w_shift, abs(H_shift), 'r', 'LineWidth', 2);
title('اندازه سیگنال شیفت یافته');
grid on;

subplot(2,2,3);
plot(w_shift, angle(H_org), 'b', 'LineWidth', 2);
title('زاویه سیگنال اصلی');
grid on;

subplot(2,2,4);
plot(w_shift, angle(H_shift), 'r', 'LineWidth', 2);
title('زاویه سیگنال شیفت یافته');
grid on;
```



شیفت زمانی تنها باعث تغییر در فاز می‌شود (ضرب در $e^{-jn_0\omega}$) و دامنه بدون تغییر می‌ماند.

فعالیت ۴: به ازای دو رشته متفاوت و دو مقدار مختلف شیفت زمانی، خروجی ها را بررسی کنید و نتیجه را بیان کنید.

```
shift_vals = [5 10];
Rs = [[10 2 7 3 15 1 9 2 5 8];
      [1 2 1 2 3 6 9 6 5 9]];
i=1;
for k = 1:4
    R = Rs(i,:);
    R_shift = [zeros(1, shift_vals(i)) R]; % شیفت به اندازه 3 نمونه

    [H_org, w] = freqz(R, 1, 1024);
    [H_shift, w_shift] = freqz(R_shift, 1, 1024);

    figure;
    subplot(2,2,1);
    plot(w, abs(H_org), 'b', 'LineWidth', 2);
    title('اندازه سیگنال اصلی');
    grid on;

    subplot(2,2,2);
    plot(w_shift, abs(H_shift), 'r', 'LineWidth', 2);
    title('اندازه سیگنال شیفت یافته');
    grid on;

    subplot(2,2,3);
    plot(w_shift, angle(H_org), 'b', 'LineWidth', 2);
    title('زاویه سیگنال اصلی');
    grid on;

    subplot(2,2,4);
    plot(w_shift, angle(H_shift), 'r', 'LineWidth', 2);
    title('زاویه سیگنال شیفت یافته');
    grid on;

    if k == 2
        i = i+1;
    end
end
```



نتیجه: دامنه همچنان تغییر نمی‌کند، فقط فاز تغییر می‌کند.

بخش ۳: شیفت فرکانسی تبدیل فوریه گسسته

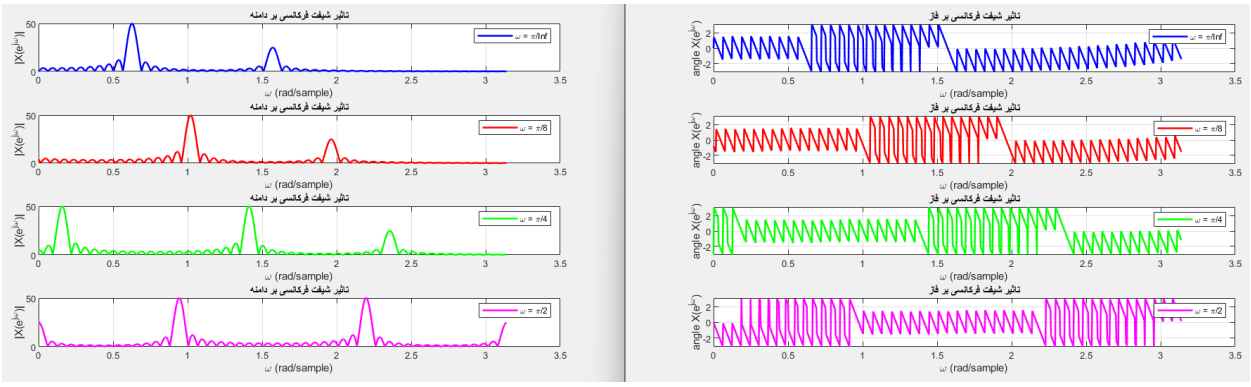
$$e^{j\omega_0 n}x[n] \quad \overset{F}{\Rightarrow} \quad X(e^{j(\omega-\omega_0)})$$

فعالیت ۵: به ازای مقادیر مختلف شیفت فرکانسی، این ویژگی را بررسی کنید. (حداقل دو مقدار)

تاثیر شیفت فرکانسی در تبدیل فوریه چیست؟

شیفت فرکانسی (ضرب در $e^{j\omega 0 n}$) باعث جابجایی کامل طیف سیگنال در حوزه فرکانس می‌شود.

```
n = 0:99;
x = sin(0.2*pi*n) + 0.5*cos(0.5*pi*n);

omega0_vals = [0, pi/8, pi/4, pi/2];
colors = {'b', 'r', 'g', 'm'};

figure;
for i = 1:length(omega0_vals)
    subplot(length(omega0_vals), 1, i);
    omega0 = omega0_vals(i);
    x_shifted = x .* exp(1j*omega0*n);
    [H, w] = freqz(x_shifted, 1, 1024);
    plot(w, abs(H), colors{i}, 'LineWidth', 1.5);
    title('تاثیر شیفت فرکانسی بر دامنه');
    legend(['\omega = \pi/' num2str(1/ (omega0_vals(i)/pi))])
    xlabel('\omega (rad/sample)');
    ylabel('|X(e^{j\omega})|');
    grid on;
end

figure;
for i = 1:length(omega0_vals)
    subplot(length(omega0_vals), 1, i);
    omega0 = omega0_vals(i);
    x_shifted = x .* exp(1j*omega0*n);
    [H, w] = freqz(x_shifted, 1, 1024);
    plot(w, angle(H), colors{i}, 'LineWidth', 1.5);
    title('تاثیر شیفت فرکانسی بر فاز');
    legend(['\omega = \pi/' num2str(1/ (omega0_vals(i)/pi))])
    xlabel('\omega (rad/sample)');
    ylabel('angle X(e^{j\omega})');
    grid on;
end
```



بخش ۴: ویژگی کانولوشن و ضرب تبدیل فوریه گسسته

عمل کانولوشن یکی از کاربردی‌ترین ابزارها در تحلیل سیستم‌های LTI است. بنابراین دانستن رفتار حوزه فرکانس دو سیگنالی که با هم کانوالو میشوند مفید خواهد بود.

فعالیت ۶: ویژگی معادل بودن کانولوشن در حوزه زمان با ضرل در حوزه فرکانس را با استفاده از دو سیگنال مختلف و دلخواه بررسی کنید.

```
x = [1 2 3];
y = [1 1 1];

conv_time = conv(x, y);

X = fft(x, 5);
Y = fft(y, 5);
conv_freq = ifft(X .* Y);

disp('کانولوشن در زمان:'); disp(conv_time);
disp('حاصل ضرب در فرکانس و تبدیل معکوس:'); disp(conv_freq);
```

```
Command Window
کانولوشن در زمان:
     1     3     6     5     3

حاصل ضرب در فرکانس و تبدیل معکوس:
    1.0000    3.0000    6.0000    5.0000    3.0000
```

```
x = [3 2 3 5 7];
y = [1 5 6 1 2];

conv_time = conv(x, y);

X = fft(x, 9);
Y = fft(y, 9);
conv_freq = ifft(X .* Y);

disp('کانولوشن در زمان:'); disp(conv_time);
disp('حاصل ضرب در فرکانس و تبدیل معکوس:'); disp(conv_freq);
```

```
Command Window
کانولوشن در زمان:
     3    17    31    35    58    72    53    17    14

حاصل ضرب در فرکانس و تبدیل معکوس:
    3.0000   17.0000   31.0000   35.0000   58.0000   72.0000   53.0000   17.0000   14.0000
```

نتیجه: هر دو برابر هستند.

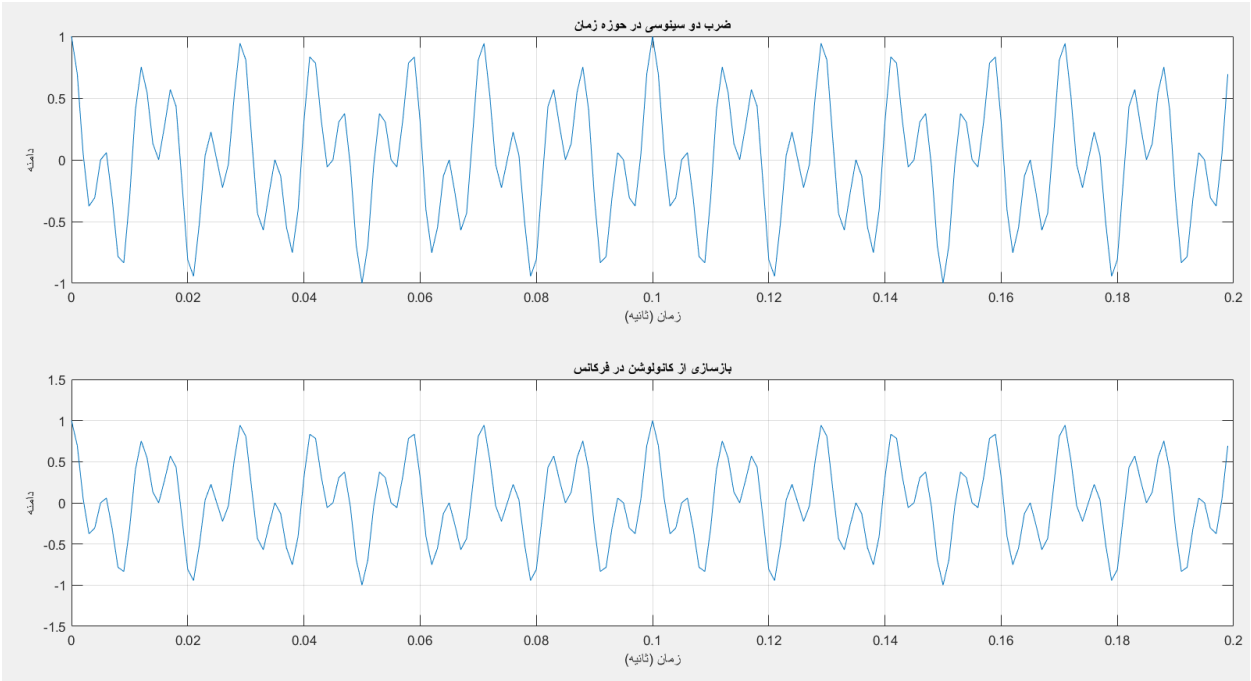
فعالیت ۷: ویژگی (مدولاسیون) یا معادل بودن ضرب در حوزه زمان با کانولوشن در حوزه فرکانس را با استفاده از دو سیگنال مختلف دلخواه بررسی و تحقیق نمایید.

```
x1 = [1 2 3 4];
x2 = [3 2 1 0];

% ضرب در حوزه زمان
prod_time = x1 .* x2;

% محاسبه کانولوشن چرخشی در حوزه فرکانس
N = length(x1);
X1 = fft(x1, N);
X2 = fft(x2, N);
conv_freq_circ = cconv(X1, X2, N) / N;
prod_freq = ifft(conv_freq_circ);

% نمایش نتایج
disp('ضرب در حوزه زمان:');
disp(prod_time);
disp('حاصل از کانولوشن در فرکانس: ');
disp(prod_freq);
```

```
ضرب در حوزه زمان:
     3     4     3     0

حاصل از کانولوشن در فرکانس:
     3     4     3     0
```

```
Fs = 1000;
t = 0:1/Fs:1-1/Fs;
f1 = 50;
f2 = 120;
x_sin1 = cos(2*pi*f1*t);
x_sin2 = cos(2*pi*f2*t);

% ضرب در زمان
prod_sin_time = x_sin1 .* x_sin2;

% کانولوشن در فرکانس
Nfft = length(t);
X_sin1 = fft(x_sin1, Nfft);
X_sin2 = fft(x_sin2, Nfft);
conv_sin_freq = cconv(X_sin1, X_sin2, Nfft) / Nfft;
prod_sin_freq = ifft(conv_sin_freq);
```

```
figure;
subplot(2,1,1);
plot(t(1:200), prod_sin_time(1:200));
title('ضرب دو سینوسی در حوزه زمان');
xlabel('زمان (ثانیه)');
ylabel('دامنه');
grid on;

subplot(2,1,2);
plot(t(1:200), real(prod_sin_freq(1:200)));
title('بازسازی از کانولوشن در فرکانس');
xlabel('زمان (ثانیه)');
ylabel('دامنه');
grid on;
```



ضرب در حوزه زمان معادل کانولوشن چرخشی در حوزه فرکانس است.

بخش ۵: ویژگی معکوس کردن زمان

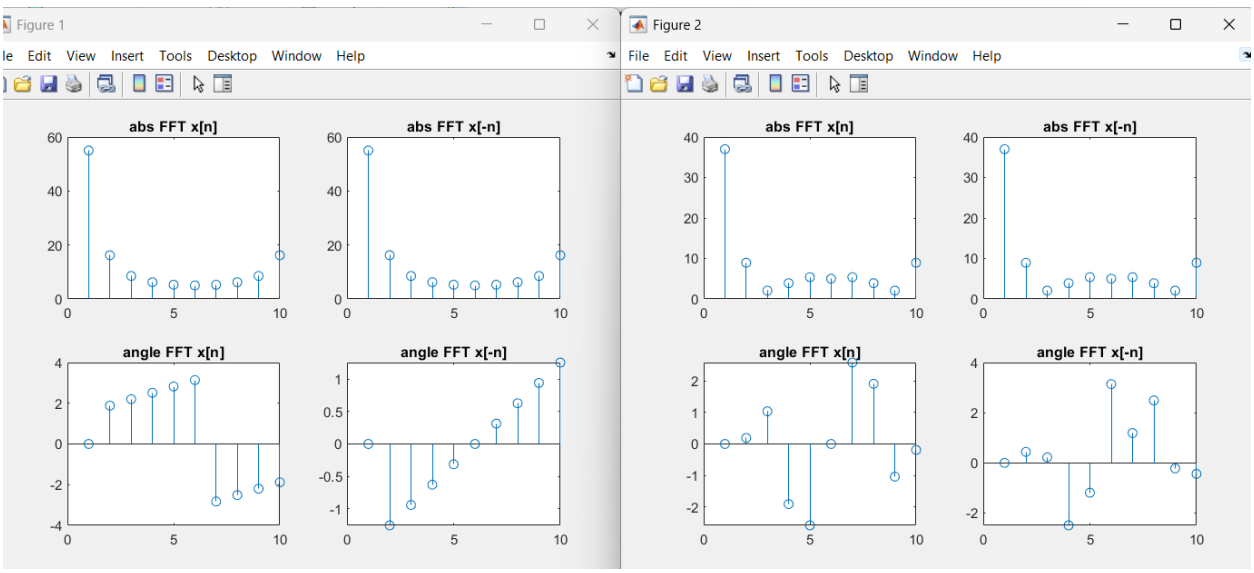
میخواهیم بدانیم که اگر متغیر زمان در سیگنال معکوس شود، تبدیل فوریه آن چه تغییری میکند؟

$$x[-n] \quad \overset{F}{\Rightarrow} \quad X(e^{-j\omega})$$

فعالیت ۸: با استفاده از دو سیگنال دلخواه خاصیت فوق را بررسی و تحقیق نمایید.

```
x = [1 2 3 4 5 6 7 8 9 10];
x_rev = fliplr(x);

X = fft(x);
X_rev = fft(x_rev);
```

```
figure;
subplot(2,2,1); stem(abs(X)); title('abs FFT x[n]');
subplot(2,2,2); stem(abs(X_rev)); title('abs FFT x[-n]');
subplot(2,2,3); stem(angle(X)); title('angle FFT x[n]');
subplot(2,2,4); stem(angle(X_rev)); title('angle FFT x[-n]');

x = [5 6 3 2 4 1 3 2 6 5];
x_rev = fliplr(x);

X = fft(x);
X_rev = fft(x_rev);

figure;
subplot(2,2,1); stem(abs(X)); title('abs FFT x[n]');
subplot(2,2,2); stem(abs(X_rev)); title('abs FFT x[-n]');
subplot(2,2,3); stem(angle(X)); title('angle FFT x[n]');
subplot(2,2,4); stem(angle(X_rev)); title('angle FFT x[-n]');
```



نتیجه: طیف دامنه یکسان، فاز قرینه می‌شود.

بخش ۶: تبدیل فوریه سریع FFT

در آزمایش قبل دیدیم که تبدیل فوریه گسسته با روابط مذکور قابل محاسبه یست و باید در مقادیر گسسته‌ای از فرکانس، تبدیل فوریه را بیابیم. برای این منظور می‌توانیم تبدیل فوریه گسسته را در مقادیر گسسته‌ی فرکانس با فاصله مساوی در بازه $0 \le \omega \le 2\pi$ به ازای $k = 0,1,\dots,N-1$، $\omega_k = \frac{2\pi k}{N}$ محاسبه کنیم، روابط بصورت زیر خواهد بود:

$$X[k] = \sum_{n=0}^{M-1} x[n] e^{-j2\pi kn/N} \; ; \; k = 0,1,\dots,N-1$$

$$x[n] = \sum_{k=0}^{N-1} x[n] e^{j2\pi kn/N} \; ; \; n = 0,1,\dots,M-1$$

بطور خلاصه:

DFS $\Rightarrow$ $\mathbf{x[n]} \longrightarrow a_k = a_{k+N}$

DTFT $\Rightarrow$ $\mathbf{x[n]} \longrightarrow \mathbf{X(e^{j\omega})}$ پیوسته ،طول نامحدود

DFT $\Rightarrow$ $\mathbf{x[n]} = \mathbf{X[k]}$ گسسته ،طول محدود

FFT $\Rightarrow$ **DFT** "N"points, $N = 2^{\gamma}$ & $\geq$ طول سیگنال

$$\mathbf{X[k]} = \sum_{n=0}^{N-1} x[n] e^{-j\left(\frac{2\pi}{N}\right)kn} \quad , \quad k = 0,1,\dots,N-1$$

برای پیاده سازی در متلب از دستور **fft(x,L)** استفاده می‌شود، که **DFT**، **L** نقطه‌ای سیگنال **x** را محاسبه می‌کند. دستور **ifft** هم معکوس این تبدیل را ارائه می‌دهد.

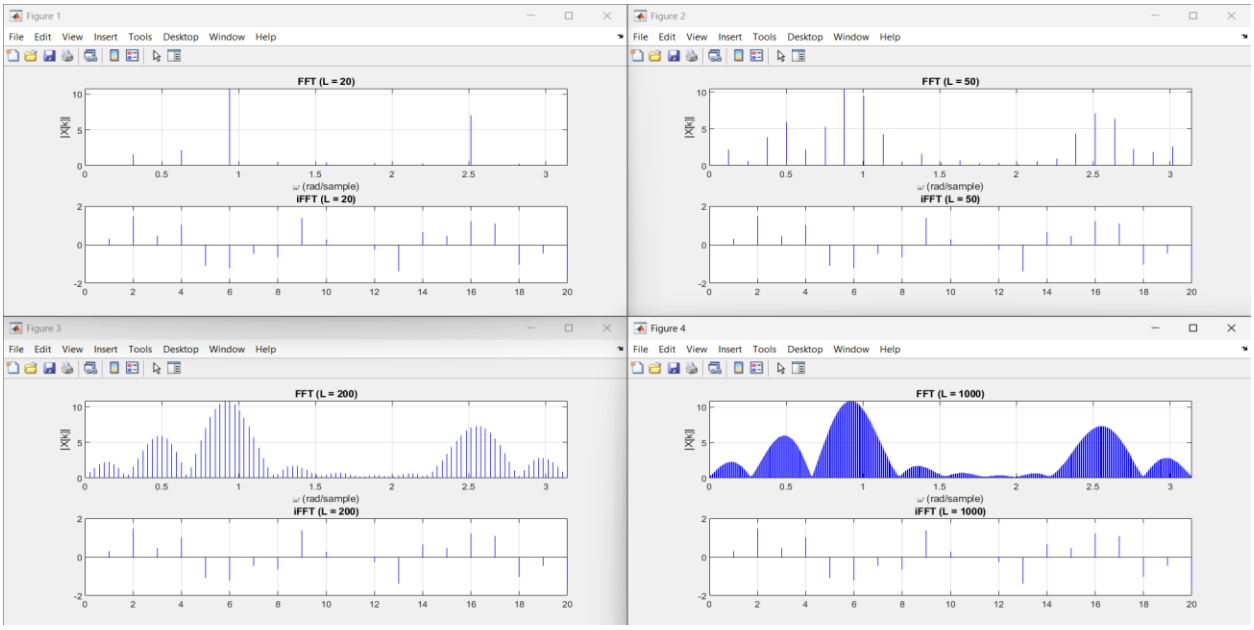
برای رشته محدود $x[n]$، تبدیل فوریه $X[k]$ با دستور $fft(x)$ با طولی برابر $x[n]$ ارائه میشود.

فعالیت ۹: برنامه‌ای بنویسید که تبدیل فوریه L نقطه‌ای سیگنال ورودی دلخواه $x[n]$ دارای N نقطه را $(L \gg N)$ محاسبه و رسم کند، سپس تبدیل فوریه معکوس L نقطه‌ای را نیز رسم کند. به ازای چند ورودی دلخواه و مقادیر مختلفی از L و N خروجی‌ها را مشاهده کنید. چه نتیجه‌ای می‌گیرید؟

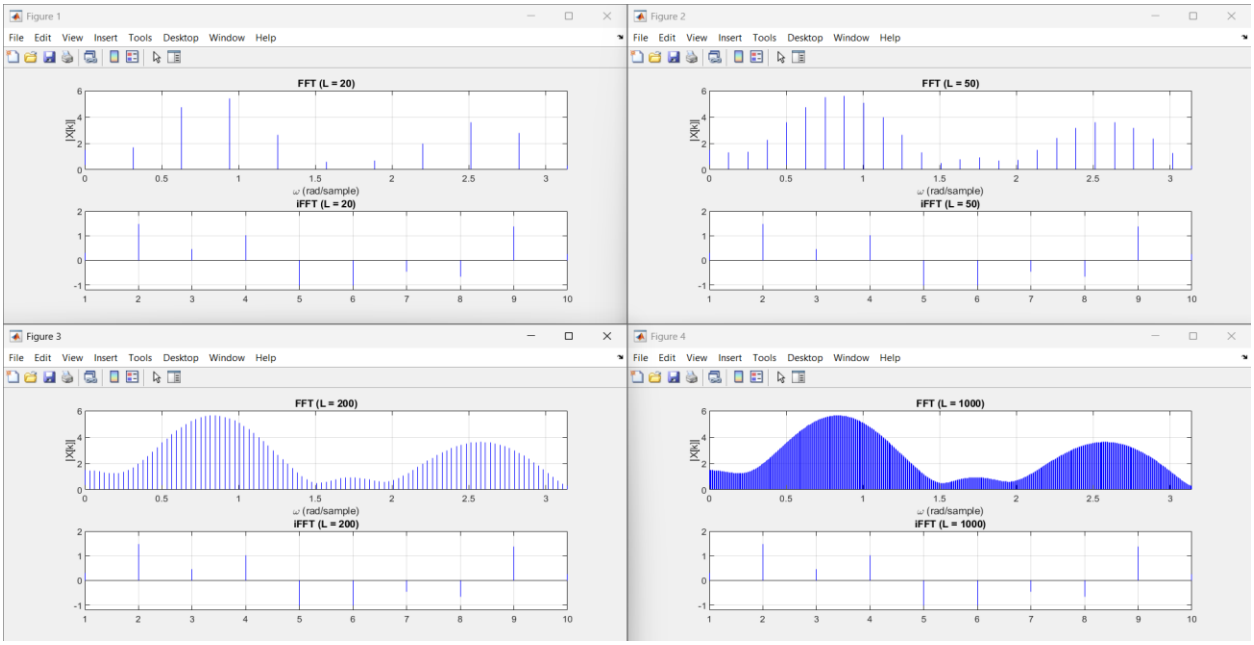
```
N = 20;
n = 0:N-1;
x = sin(0.3*pi*n) + 0.7*sin(0.8*pi*n) + 0.3*cos(0.15*pi*n);
L_vals = [20, 50, 200, 1000];

for I = 1:length(L_vals)
    L = L_vals(i);
    X = fft(x, L);
    x_recon = ifft(X, L);
    x_recon = real(x_recon(1:N));

    figure;
    subplot(2, 1, 1);
    k = 0:L-1;
```

```
    wk = k * (2*pi/L);  % فرکانس‌های نرمال‌شده
    stem(wk, abs(X), 'b', 'Marker', 'none', 'LineWidth', 0.5);
    title( print('FFT (L = %d)', L));
    xlabel('\omega (rad/sample)');
    ylabel('|X[k]|');
    xlim([0, pi]);
    grid on;
    subplot(2, 1, 2);
    stem(x_recon, 'b', 'Marker', 'none', 'LineWidth', 0.5);
    title( print('iFFT (L = %d)', L));
    grid on;
end
```

N=20



N=10



هرچه L بزرگتر باشد:

نقاط فرکانسی بیشتری داریم و طیف با جزئیات بیشتری دیده می‌شود.

FFT به DTFT واقعی نزدیک‌تر می‌شود.

رزولوشن فرکانسی ($\Delta\omega = 2\pi/L$) بهبود می‌یابد.

بخش ۷: ویژگی‌های شیفت و کانولوشن چرخشی fft

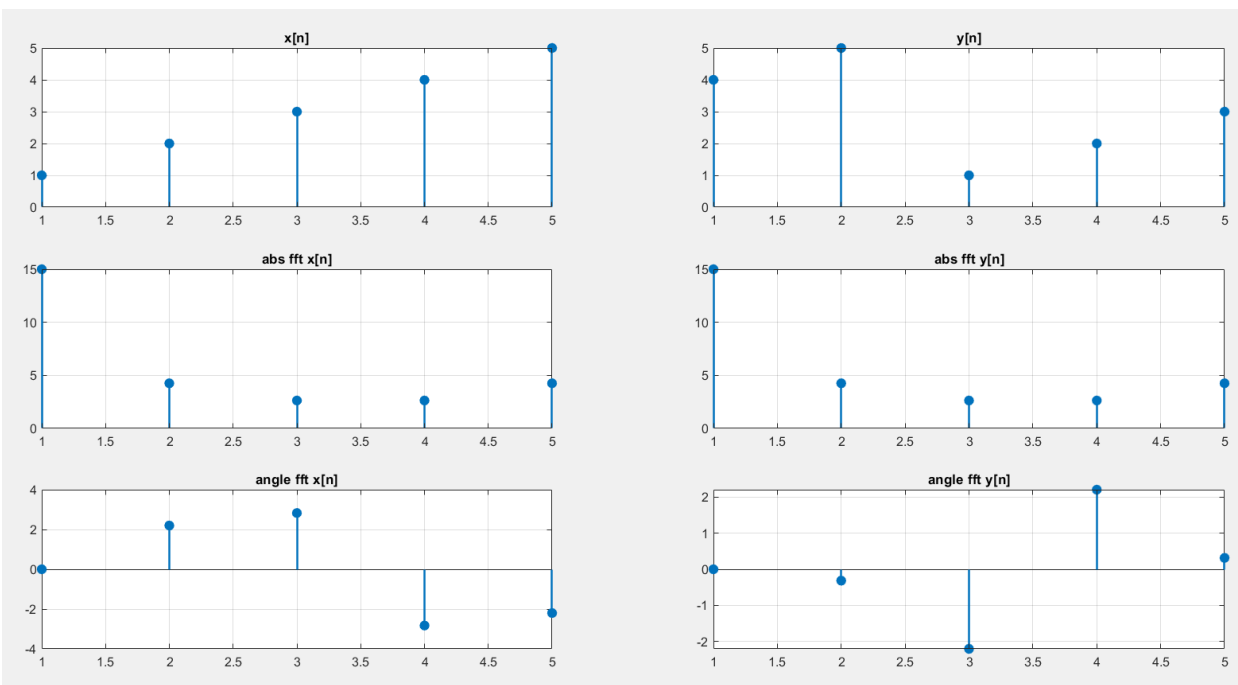
فعالیت ۱۰: تابعی برای انجام شیفت چرخشی بنویسید. نام تابع را *circshift1* بگذارید. در یک فایل دیگر از این تابع بر روی یک سیگنال دلخواه استفاده کنید. اگر مقدار شیفت از طول سیگنال بیشتر باشد چه اتفاقی می‌افتد؟ خاصیت شیفت چرخشی تبدیل فوریه گسسته چیست؟

```
function y = circshift1(x, m)
    % circshift1: شیفت چرخشی سیگنال
    % x: بردار ورودی
    % m: مقدار شیفت
    N = length(x);
    m = mod(m, N);
    if m == 0
        y = x;
    else
        y = [x(N-m+1:end), x(1:N-m)];
    end
end

x_test = [1 2 3 4 5];
N_test = length(x_test);
m_test = 2;

y_test = circshift1(x_test, m_test);
disp('سیگنال اصلی x:'); disp(x_test);
disp('سیگنال شیفت یافته y:'); disp(y_test);
```

```
Command Window
سیگنال اصلی x:
     1     2     3     4     5

سیگنال شیفت یافته y:
     4     5     1     2     3
```

```
x_test = [1 2 3 4 5];
N_test = length(x_test);
m_test = 2;

y_test = circshift1(x_test, m_test);

X_fft = fft(x_test);
Y_fft = fft(y_test);
```

```
subplot(3,2,1);
stem(x_test, 'filled', 'LineWidth', 1.5);
title('x[n]');
grid on;

subplot(3,2,2);
stem(y_test, 'filled', 'LineWidth', 1.5);
title('y[n]');
grid on;

subplot(3,2,3);
stem(abs(X_fft), 'filled', 'LineWidth', 1.5);
title('abs fft x[n]');
grid on;

subplot(3,2,4);
stem(abs(Y_fft), 'filled', 'LineWidth', 1.5);
title('abs fft y[n]');
grid on;

subplot(3,2,5);
stem(angle(X_fft), 'filled', 'LineWidth', 1.5);
title('angle fft x[n]');
grid on;

subplot(3,2,6);
stem(angle(Y_fft), 'filled', 'LineWidth', 1.5);
title('angle fft y[n]');
grid on;
```



۱. اگر مقدار شیفت از طول سیگنال بیشتر باشد چه اتفاقی می‌افتد؟

وقتی مقدار شیفت m از طول سیگنال N بیشتر باشد، معادل شیفت به اندازه mod(m, N) است. به عنوان مثال، روی سیگنال 8 نمونه‌ای، شیفت 10 نمونه معادل شیفت 2 نمونه است.

خاصیت شیفت چرخشی تبدیل فوریه گسسته چیست؟

خاصیت شیفت چرخشی DFT می‌گوید:

$$x[(n-m)\text{mod } N] \overset{DFT}{\rightarrow} X[k] \cdot e^{-j2\pi km/N}$$

یعنی:

دامنه طیف تغییری نمی‌کند: $|Y[k]| = |X[k]|$

فاز طیف تغییر خطی می‌کند: $\angle Y[k] = \angle X[k] - 2\pi km/N$

فعالیت ۱۱: تابعی برای انجام کانولوشن چرخشی بنویسید. نام تابع را *circconv1* بگذارید. در یک برنامه دیگر از این تابع برای انجام کانولشن چرخشی بر روی دو سیگنال دلخواه استفاده کنید. خاصیت کانولوشن چرخشی تبدیل فوریه را توضیح دهید.

خاصیت کانولوشن چرخشی تبدیل فوریه چیست؟

خاصیت اصلی این است که کانولوشن چرخشی در حوزه زمان معادل ضرب مستقیم در حوزه فرکانس است:

$$x[n] \circledast h[n] \overset{DFT}{\rightarrow} X[k] \cdot H[k]$$

```
function y = circconv1(x, h)
    % circconv1: کانولوشن چرخشی دو سیگنال با طول مساوی
    % x, h: بردارهای ورودی با طول یکسان
    N = length(x);
    if length(h) ~= N
        error('طول دو سیگنال باید برابر باشد');
    end

    % روش 1
    X = fft(x, N);
    H = fft(h, N);
    y = ifft(X .* H, N);

    % روش 2
    y = zeros(1, N);
    for n = 1:N
        for m = 1:N
            idx = mod(n-m, N) + 1;
            y(n) = y(n) + x(m) * h(idx);
        end
    end
end

x1 = [1 2 3];
h1 = [1 1 1];
y1 = circconv1(x1, h1);

disp('مثال 1:');
```

```
disp(['x = ', num2str(x1)]);
disp(['h = ', num2str(h1)]);
disp('کانولوشن چرخشی:'); disp(y1);

x2 = [1 1 1 1];
h2 = [1 1 1 1];
y2_circ = circconv1(x2, h2);

disp('مثال 2:');
disp(['x = ', num2str(x2)]);
disp(['h = ', num2str(h2)]);
disp('کانولوشن چرخشی:'); disp(y2_circ);
```

Command Window

```
مثال 1:
x = 1  2  3
h = 1  1  1
کانولوشن چرخشی:
     6     6     6

مثال 2:
x = 1  1  1  1
h = 1  1  1  1
کانولوشن چرخشی:
     4     4     4     4
```

فعالیت ۱۲: با استفاده از کانولوشن چرخشی، کانولوشن خطی را برای دو سیگنال با طول متفاوت پیاده‌سازی کنید. چه خاصیتی بین کانولشن چرخشی و کانولوشن خطی برقرار است؟

چه خاصیتی بین کانولوشن چرخشی و کانولوشن خطی برقرار است؟

رابطه اصلی این است که با صفرپردازی مناسب می‌توان کانولوشن خطی را از کانولوشن چرخشی بدست آورد:

$$\text{conv_linear}(x,h) = \text{cconv}(\text{zero_pad}(x,N),\text{zero_pad}(h,N))$$

که در آن $N = \text{len}(x) + \text{len}(h) - 1$.

```
function y_linear = linear_conv_using_circconv1(x, h)
    Nx = length(x);
    Nh = length(h);
    N = Nx + Nh - 1;  % طول مورد نیاز

    % صفرپردازی (Zero Padding)
    x_pad = [x, zeros(1, N - Nx)];
    h_pad = [h, zeros(1, N - Nh)];

    % کانولوشن چرخشی روی سیگنال‌های صفرپردازی شده
    y_circ = circconv1(x_pad, h_pad);
    y_linear = real(y_circ);  % حذف خطای عددی موهومی
end

x1 = [1 2 3];
```

```
h1 = [4 5];
y1_true = conv(x1, h1);
y1_circ_method = linear_conv_using_circconv1(x1, h1);

disp('x = [1 2 3], h = [4 5]');
y1_true
y1_circ_method



x2 = [1 1 1];
h2 = [1 1 1];
y2_true = conv(x2, h2);
y2_circ_method = linear_conv_using_circconv1(x2, h2);

disp('x = [1 1 1], h = [1 1 1]');
y2_true
y2_circ_method
```

Command Window

```
x = [1 2 3], h = [4 5]

y1_true =

     4    13    22    15


y1_circ_method =

     4    13    22    15

x = [1 1 1], h = [1 1 1]

y2_true =

     1     2     3     2     1


y2_circ_method =

    1.0000    2.0000    3.0000    2.0000    1.0000
```

فعالیت ۱۳: برنامه‌ای بنویسید که کانولوشن دو سیگنال را با استفاده از تبدیل فوریه تک‌تک آن‌ها محاسبه کند.

```
function y = conv_fft(x, h)
    Nx = length(x);
    Nh = length(h);
    N = Nx + Nh - 1;  % طول مورد نیاز برای کانولوشن خطی

    % FFT با صفرپردازی
    X = fft(x, N);
    H = fft(h, N);
```

```
    % ضرب در فرکانس
    Y = X .* H;

    % تبدیل معکوس
    y = ifft(Y, N);
    y = real(y);
end
-----------------------------------------------------------------------------------------
x1 = [1 2 3];
h1 = [4 5];
y1_fft = conv_fft(x1, h1);
y1_conv = conv(x1, h1);

disp('x = [1 2 3], h = [4 5]');
disp('کانولوشن با FFT:'); disp(y1_fft);
disp('کانولوشن مستقیم:'); disp(y1_conv);

x2 = [1 -1 2 -2 3 -3 4 -4];
h2 = [0.2 0.5 0.8 0.5 0.2];
y2_fft = conv_fft(x2, h2);
y2_conv = conv(x2, h2);
disp('x = [1 -1 2 -2 3 -3 4 -4], h = [0.2 0.5 0.8 0.5 0.2]');
disp('کانولوشن با FFT:'); disp(y2_fft);
disp('کانولوشن مستقیم:'); disp(y2_conv);
```

Command Window

```
x = [1 2 3], h = [4 5]
کانولوشن با FFT:
     4    13    22    15

کانولوشن مستقیم:
     4    13    22    15

x = [1 -1 2 -2 3 -3 4 -4], h = [0.2 0.5 0.8 0.5 0.2]
کانولوشن با FFT:
    0.2000    0.3000    0.7000    0.3000    0.9000    0.1000    1.1000   -0.1000    0.3000   -1.8000   -1.2000   -0.8000

کانولوشن مستقیم:
    0.2000    0.3000    0.7000    0.3000    0.9000    0.1000    1.1000   -0.1000    0.3000   -1.8000   -1.2000   -0.8000
```