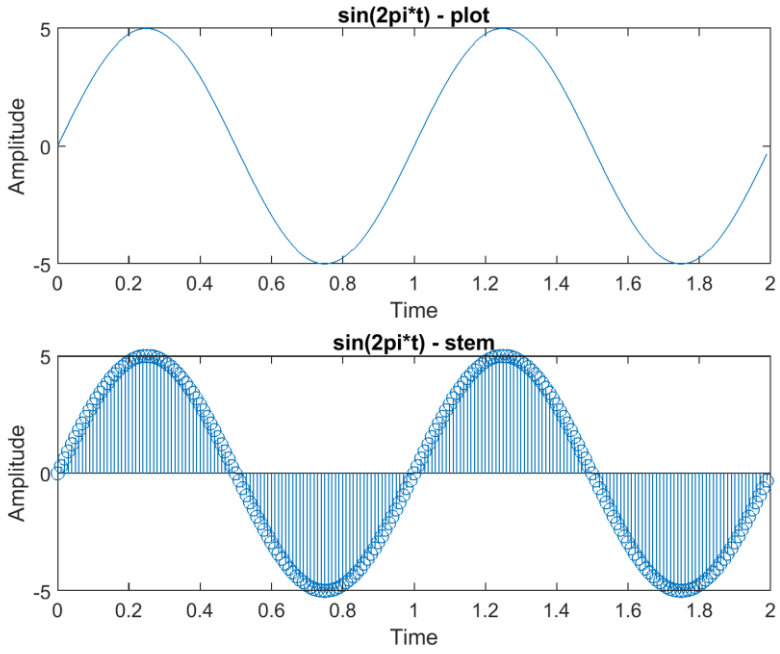
" بسمه تعالی"

## گزارش اول آزمایشگاه DSP – زهرا لطیفی – ۹۹۲۳۰۶۹

### بخش ۱-۱)

```
t = 0:0.01:1.99;
A = 5;
f = 1;
sig = A*sin(2*pi*f*t);

figure(1)
subplot(2,1,1);
plot(t, sig);
title("sin(2pi*t) - plot");
ylabel("Amplitude");
xlabel("Time");

subplot(2,1,2);
stem(t, sig);
title("sin(2pi*t) - stem");
ylabel("Amplitude");
xlabel("Time");
```

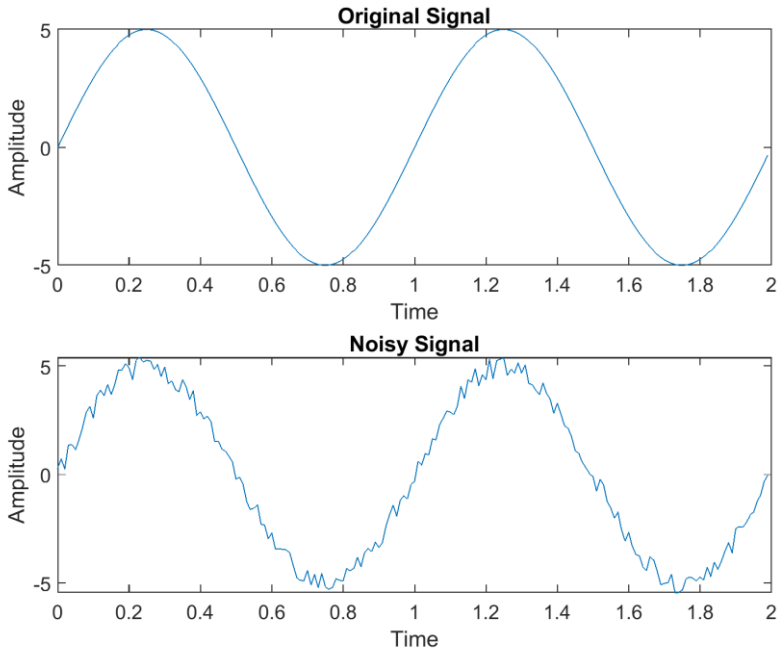



*شکل ۱*

در این بخش با استفاده از دستورات stem و plot، سیگنال سینوسی با فرکانس ۱ هرتز و دامنه ۵ را در بازه زمانی ۲-۰ ثانیه رسم کردیم. البته با استفاده از دستور subplot هردو را در یک تصویر رسم کردیم.

**بخش ۲-۱)**

```
noise = rand(1, 200) - 0.5; % minus 0.5 to set mean at zero
newSig = sig + noise;

figure(2)
subplot(2,1,1);
plot(t, sig);
title("Original Signal");
ylabel("Amplitude");
xlabel("Time");

subplot(2,1,2);
plot(t, newSig);
title("Noisy Signal");
ylabel("Amplitude");
xlabel("Time");
```

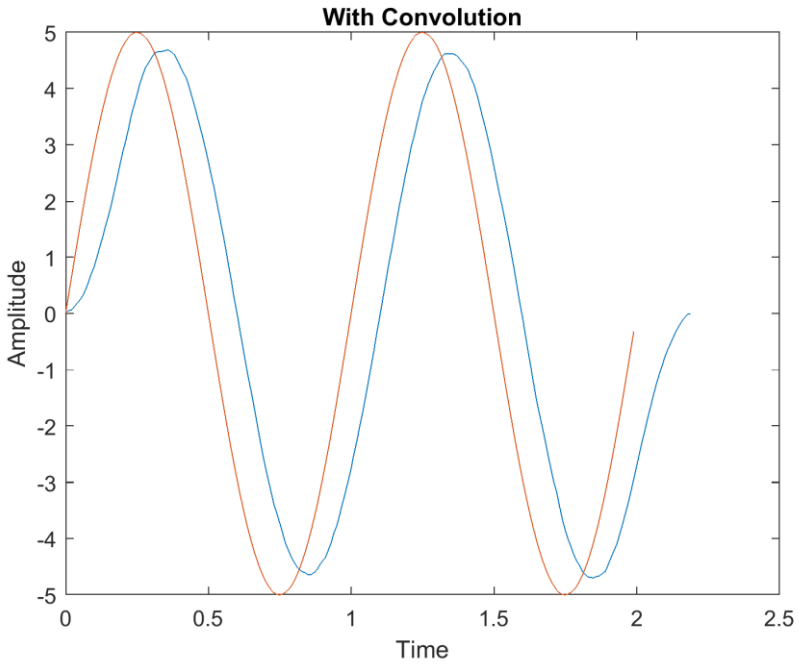



شکل ۲

در این بخش با استفاده از تابع rand، نویز ایجاد کرده و با سیگنال سینوسی بخش قبل جمع کردیم. البته این نویز دارای توزیع یکنواخت بین ۰ و ۱ می‌باشد، لذا لازم است عدد ۰.۵ از آن کاسته شود تا میانگین نویز حاصل، برابر صفر شود. سپس سیگنال اصلی و نویزی را بطور همزمان با دستور subplot رسم کردیم.

**بخش ۳-۱)**

```
t1 = 0:0.01:2.19; % Updating time variable for extension resulting from convolution
movAvg = ones(1, 21)/21; % Dividing by moving average length to avoid increasing signal
energy
conv = conv(newSig, movAvg);

figure(3)
plot(t1, conv);
title("With Convolution");
ylabel("Amplitude");
xlabel("Time");
hold on;
plot(t, sig);
```



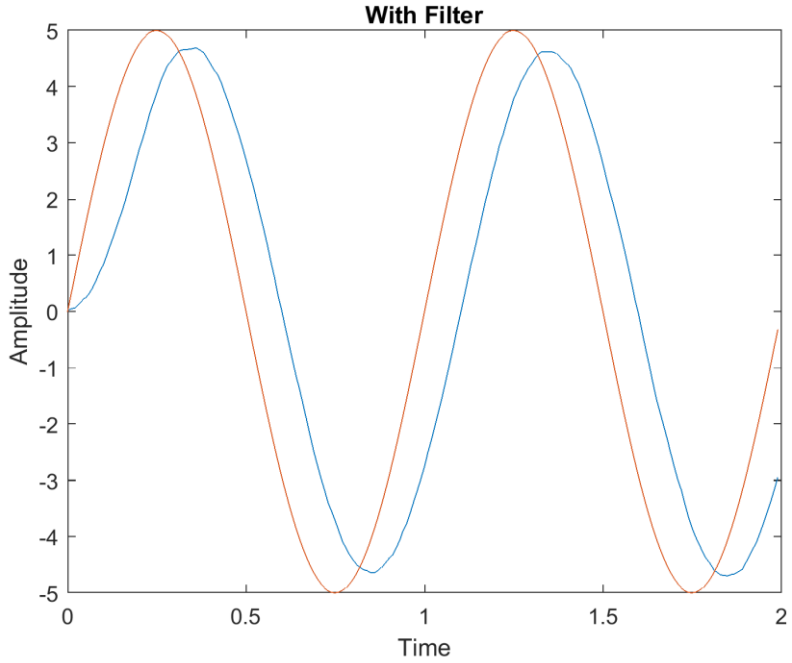

*شکل ۳*

در این بخش بین یک Moving Average و سیگنال نویزی، کانولوشن گرفتیم. سیگنال قرمز همان سیگنال اصلی ما و سیگنال آبی خروجی دستور conv هستند. مشاهده می‌شود که طول خروجی حاصل از دستور conv برابر با ۲۲۰ نمونه است که حاصل جمع طول سیگنال نویزی با طول Moving Average منهای یک است. می‌دانیم که Moving Average یک فیلتر FIR با فاز خطی است که موجب ایجاد تاخیر در پاسخ می‌شود. از آنجا که طول Moving Average برابر با ۲۱ نمونه است، به اندازه نصف آن یعنی ۱۰ نمونه، تاخیر به سیستم اضافه شده است.

هنگام ایجاد Moving Average هم یک تقسیم بر ۲۱ (طول سیگنال) انجام دادیم تا انرژی سیستم افزایش پیدا نکند.

**بخش ۴-۱)**

```
coef = ones(1, 21)/21;
filt = filter(coef, 1, newSig);

figure(4)
plot(t, filt);
title("With Filter");
ylabel("Amplitude");
xlabel("Time");
hold on;
plot(t, sig);
```

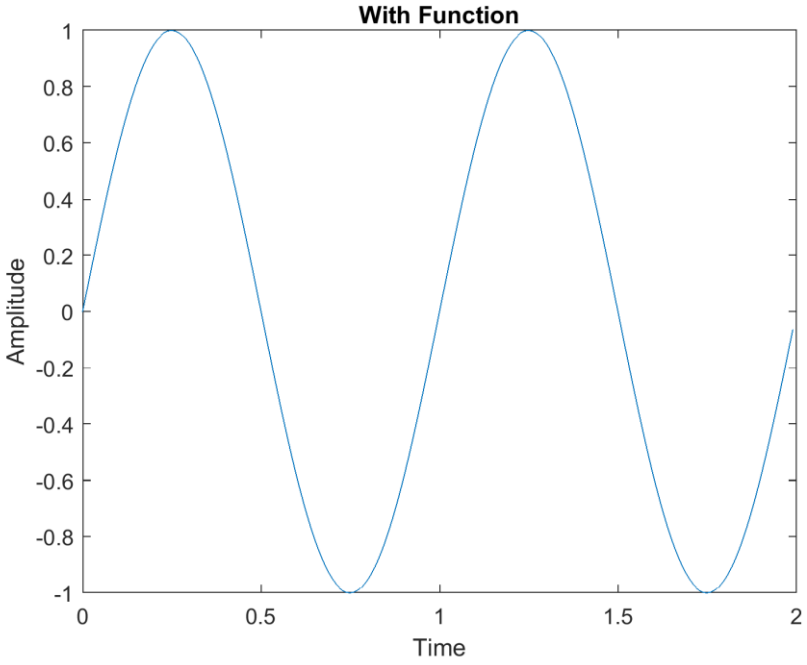



*شکل ۴*

در این بخش همان عملیاتی که در بخش قبل انجام شد را این بار با استفاده از دستور filter انجام دادیم. برای تعریف فیلتر FIR به صورت، ۲۱ ضریب ۱/۲۱ و به مخرج هم تنها یک را دادیم. شاهدیم که باز هم تاخیر ۱۰ نمونه را داریم ولی دیگر افزایش طول را نداریم و سیگال ایجاد شده همانند سیگنال اصلی همان ۲۰۰ نمونه را دارد.

**بخش ۵-۱)**

```
% function [y] = singen(w, n)
%
%     t = 0:1/n:(n-1)/n;
%     y = sin(w.*t);
%
% end

f = 2;
w = 2*pi*f;
n = 200;
func = singen(w, n);

figure(5)
plot(t, func);
title("With Function");
ylabel("Amplitude");
xlabel("Time");
```



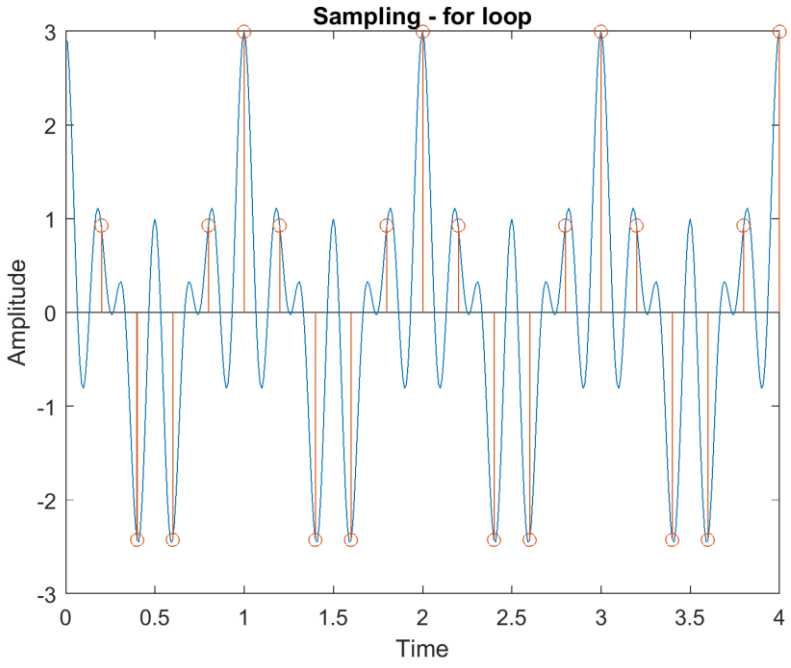

*شکل ۵*

در این بخش تابعی به نام singen رادر فایل singen.m تعریف کردیم که ω و n را به عنوان ورودی گرفته و موج سینوسی sin(ωt) که در آن تعریف t به n بستگی دارد را به عنوان خروجی می‌دهد.

**بخش ۱-۶)**

```
F = 100; % Sine generation frequency
fs = 5; % Sampling frequency
n = 4;
t2 = 1/F:1/F:n;

x = cos(2*pi*t2) + cos(8*pi*t2) + cos(12*pi*t2);
figure(6)
plot(t2, x);
title("Sampling - for loop");
ylabel("Amplitude");
xlabel("Time");
```

```
% Sampling with a for loop
tSamp= 1/fs:1/fs:n;
xSamp = zeros(1, n*fs);
for i = 1:1:n*fs
    xSamp(i) = x(i*(F/fs));
end

hold on;
stem(tSamp, xSamp);
```

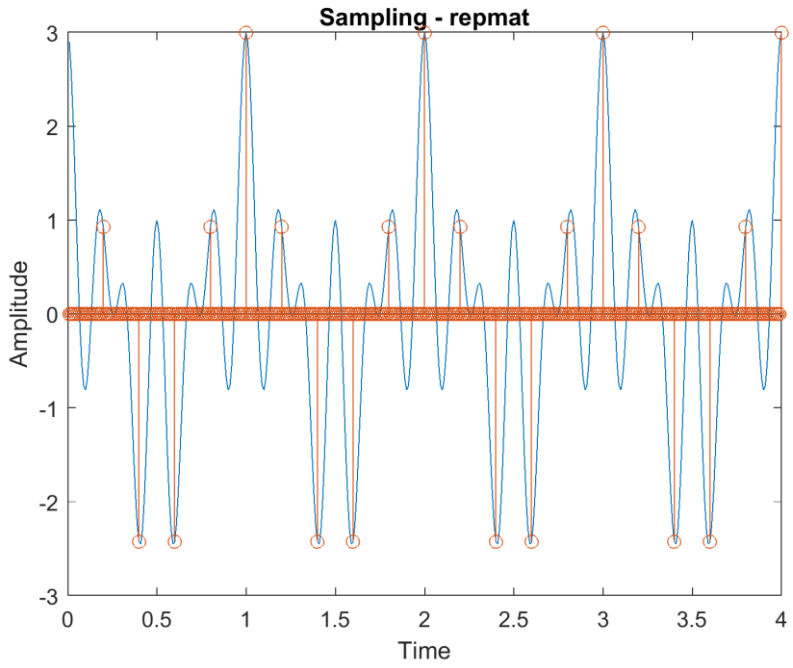



*شکل ۶*

در این بخش خواسته شده سیگنالی به طول ۴ میلی ثانیه را رسم کرده و سپس با نرخ ۵ کیلوهرتز نمونه برداری کنیم. (ضریب میلی و کیلو یکدیگر را خنثی کرده و دیگر لحاظ نشده‌اند.) برای اینکه اثر نمونه برداری مشهود باشد، سیگنال اصلی را با نرخ ۱۰۰ کیلوهرتز ایجاد کردیم. سپس با کمک یک حلقه for نمونه‌های مربوط به نرخ ۵ کیلوهرتز را برداشتیم. (از هر ۲۰ نمونه یک نمونه) سپس با دستور stem آن‌ها را روی سیگنال اصلی نمایش دادیم.

پس از آن یک بار هم این کار را با روش repmat و ایجاد یک ماسک که در آن از هر ۲۰ درایه، یکی ۱ و مابقی صفر هستند و ضرب آن در سیگنال اصلی انجام دادیم. در این روش اما نمونه‌های برابر صفر هم نمایش داده می‌شوند.

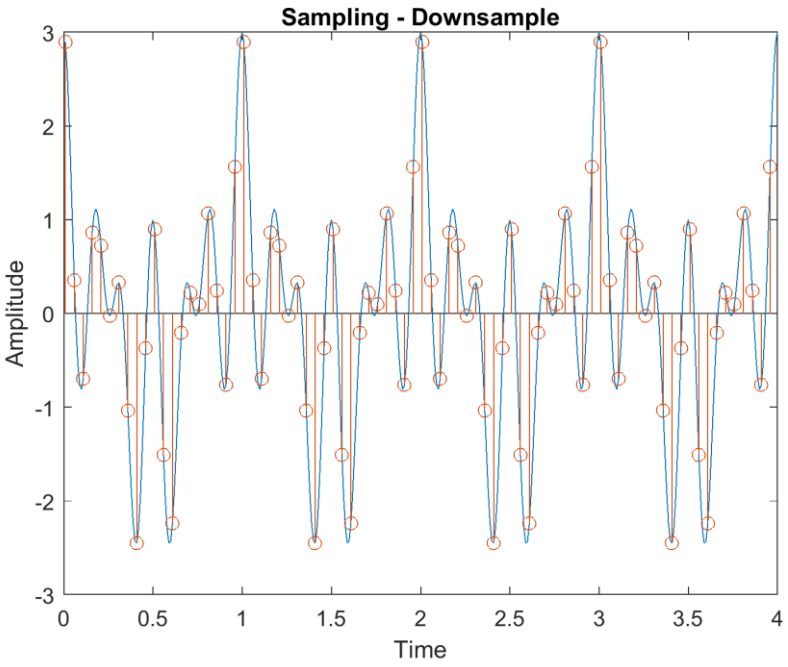
```
% Sampling with repmat
mask = [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1];
mask = repmat(mask, 1, F/fs);
xSamp1 =  x.*mask;
figure(7)
plot(t2, x);
title("Sampling - repmat");
ylabel("Amplitude");
xlabel("Time");
hold on;
stem(t2, xSamp1);
```




شکل ۷

انجام این کار با روش دیگری هم ممکن بود و آن استفاده از دستور downsample بود. در این بخش جهت جلوگیری از تکرار بیش از حد، با نرخ ۲۰ کیلوهرتز نمونه برداری را انجام دادیم تا شاهد نتیجه بهتری باشیم.

```
% Sampling with downsample & better sampling frequency
fs1 = 20;
xSamp2 =  downsample(x, F/fs1);
tSamp1 = 1/F:1/fs1:n-1/F;

figure(8)
plot(t2, x);
title("Sampling - Downsample");
ylabel("Amplitude");
xlabel("Time");
hold on;
stem(tSamp1, xSamp2);
```




*شکل ۸*

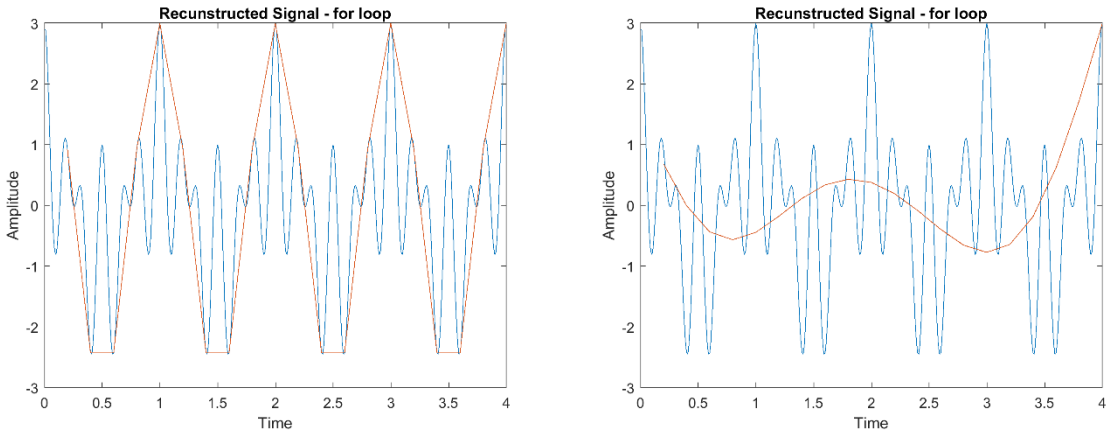
حال با دستورlowpass سیگنال نمونه برداری شده با حلقه for را بازسازی می‌کنیم. مشاهده می‌شود که پس از بازسازی سیگنال از روی نمونه‌ها، با توجه به عدم رعایت نرخ نایکوییست، فرکانس‌های بالا از دست رفته اند. به سیگنال بازسازی شده حول ۰.۵، ۱.۵، ۲.۵ و ۳.۵ دقت کنید؛ تقریبا صاف است و تغییرات سیگنال اولیه را ندارد؛ چرا که تغییرات در این بازه مرتبط با هارمونیک ۶ کیلوهرتز می‌باشد که هنگام نمونه برداری از دست رفته است. در ارتباط با طول فیلتر هم باید گفت اگر فرکانس قطع آن را از ۵ به مثلا نیم کاهش دهیم، فرکانس‌های بیشتری از دست رفته و خروجی دیگر ورودی را حتی دنبال هم نمی‌کند.

```
% Reconstruction
filt = lowpass(xSamp, 5, 5);

figure(9)
plot(t2, x);
hold on;
plot(tSamp, filt);
title("Recunstructed Signal - for loop");
ylabel("Amplitude");
xlabel("Time");
```

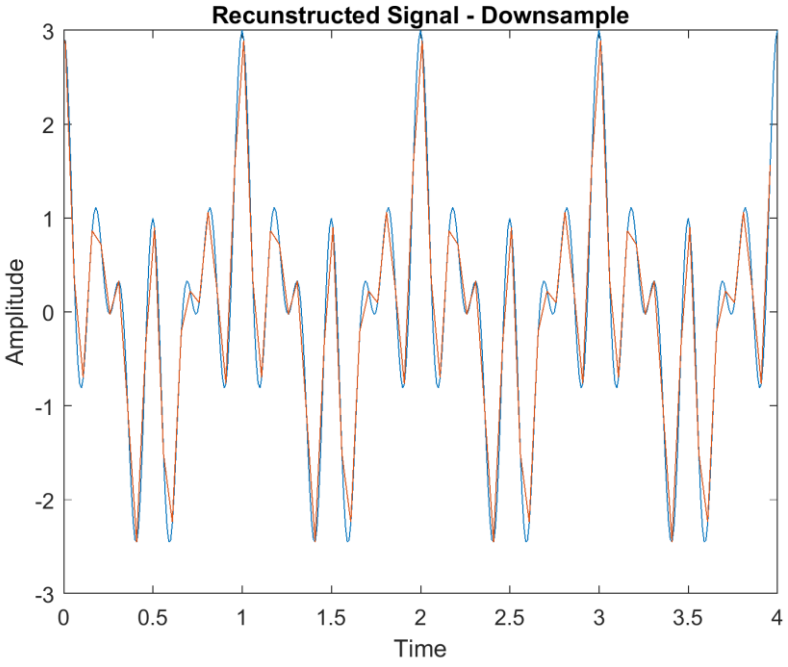
بازسازی را برای سیگنال نمونه برداری شده با دستور downsample هم انجام دادیم.

```
filt1 = lowpass(xSamp2, 5, 5);
figure(10)
plot(t2, x);
hold on;
plot(tSamp1, filt1);
title("Recunstructed Signal - Downsample");
ylabel("Amplitude");
xlabel("Time");
```

نتایج در تصاویر صفحه بعد قابل مقایسه هستند.




شکل ۹



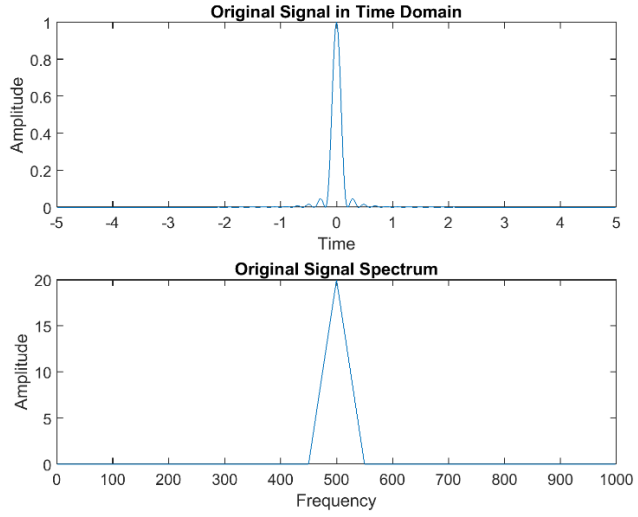

شکل ۱۰

**بخش ۷-۱)**

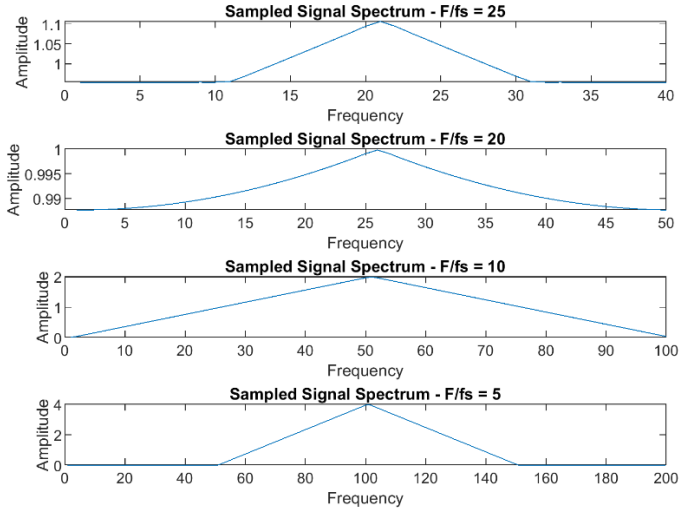
```
t3 = -4.99:0.01:4.99;
y = sinc(5*t3).^2;
figure();
subplot(2, 1, 1),
plot(t3, y);
title('Original Signal in Time Domain');
ylabel("Amplitude");
xlabel("Time");
subplot(2, 1, 2),
plot(abs(fftshift(fft(y))))
title('Original Signal Spectrum');
ylabel("Amplitude");
xlabel("Frequency");
```

```
k = 1;
figure(),
for i = [25, 20, 10, 5] % [100/4 , 100/5, 100/10, 100/20]

    ySamp = downsample(y, i);
    subplot(4, 1, k)
    plot(abs(fftshift(fft(ySamp))))
    title(sprintf("Sampled Signal Spectrum - F/fs = %d", i));
    ylabel("Amplitude");
    xlabel("Frequency");
    k = k+1;

end
```

سیگنال $sinc^2(5t)$ را در بازه -۵ و ۵ با طول گام‌های ۰.۰۱ و طیف آن در حوزه فرکانس را که با دستور abs(fftshift(fft)) بدست آمده، رسم کردیم. سپس این سیگنال را با نرخ‌های ۴، ۵، ۱۰ و ۲۰ هرتز نمونه برداری کردیم. این نمونه برداری‌ها در یک حلقه for انجام شده و برای هرکدام، طیف آن در حوزه فرکانس رسم شده است.




شکل ۱۱

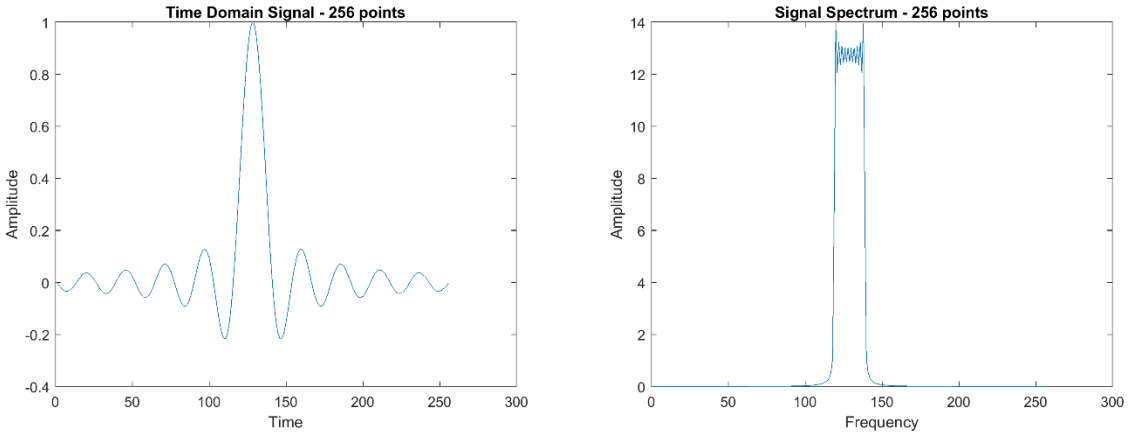



شکل ۱۲

**بخش ۱-۸)**

سیگنال sinc(2t) با تعداد نقاط ۲۵۶، ۱۲۸، ۳۸۴ و ۷۶۸ و دستور linspace نمونه برداری شده است. در هر مورد سیگنال نمونه برداری شده به همراه طیف آن رسم شده است. توجه داریم که محور فرکانس scale شده است. ((0:N-1)*(256/N) )

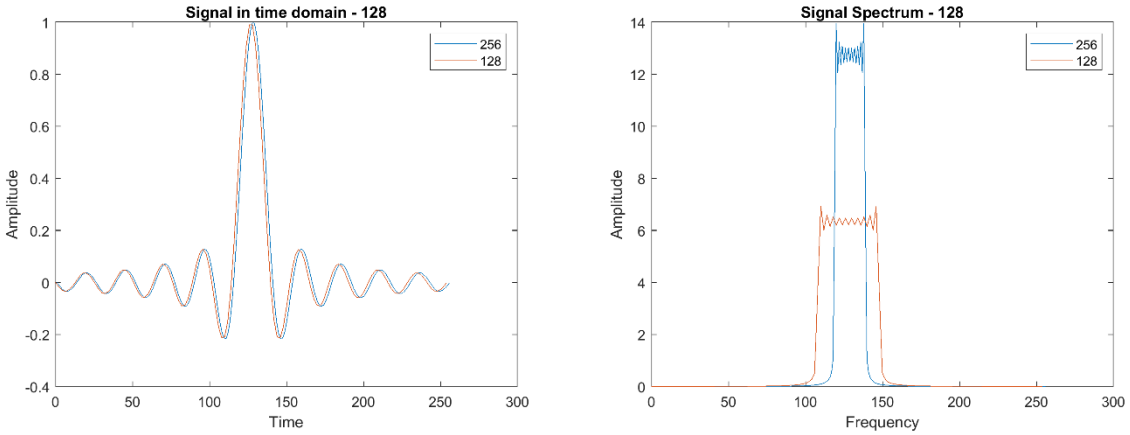
```
% Sampling in 256 points
t4 = linspace(-5, 5, 256);
z1 = sinc(2*t4);

figure()
plot(z1);
title("Time Domain Signal - 256 points");
ylabel("Amplitude");
xlabel("Time");

figure()
plot(abs(fftshift(fft(z1))));
title("Signal Spectrum - 256 points");
ylabel("Amplitude");
xlabel("Frequency");
```



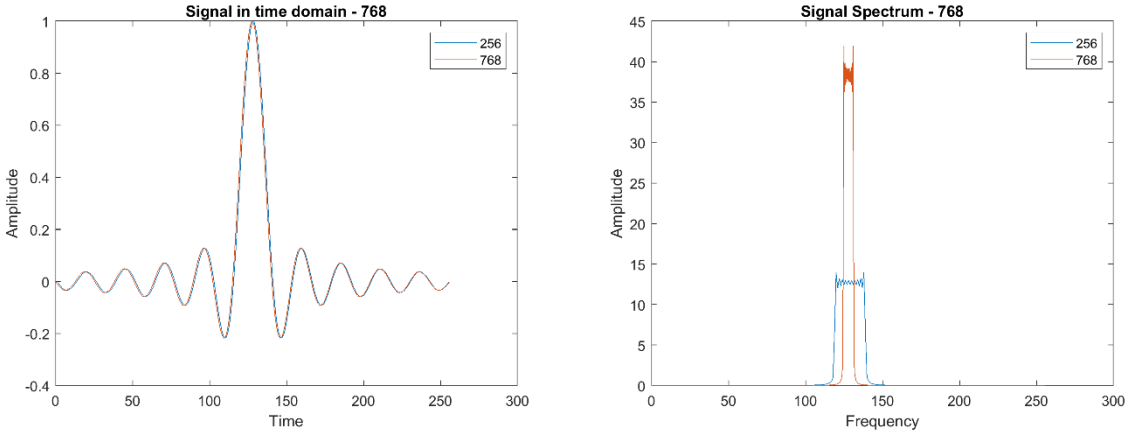

شکل۱۳

```
% Resample to 128, 768, 384 points & plot signal in time & frequency domain
for i = [128, 768, 384]

    t5 = linspace(-5, 5, i);
    z2 = sinc(2*t5);
    figure()
    plot(abs(fftshift(fft(z1))));
    title(sprintf("Signal Spectrum - %d", i));
    ylabel("Amplitude");
    xlabel("Frequency");
    f_axis = (0:i-1) * (256/i);
    hold on;
    plot(f_axis, abs(fftshift(fft(z2))));
    legend("256", sprintf("%d", i));
    figure()
    plot(z1);
    title(sprintf("Signal in time domain - %d", i));
```

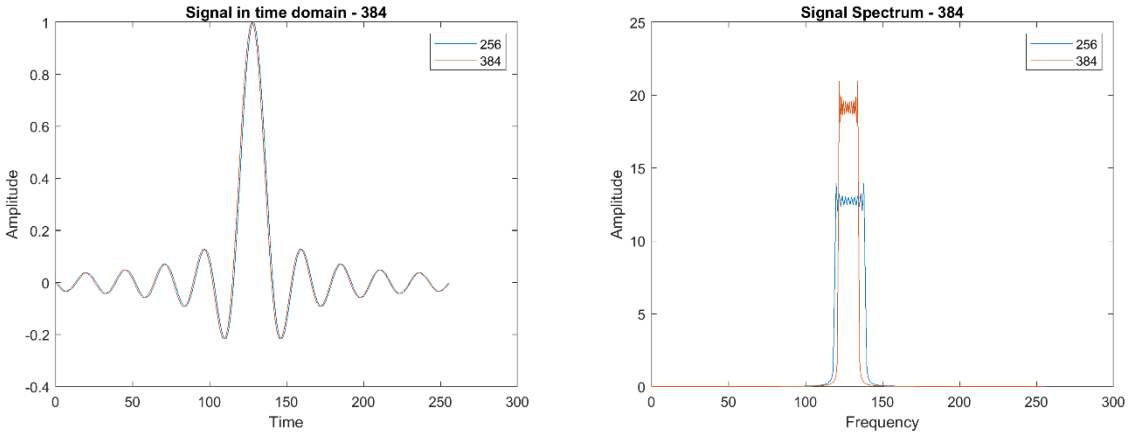
```
    ylabel("Amplitude");
    xlabel("Time");
    t_axis = (0:i-1) * (256/i);
    hold on;
    plot(t_axis, z2);
    legend("256", sprintf("%d", i));
end
```




شکل۱۴



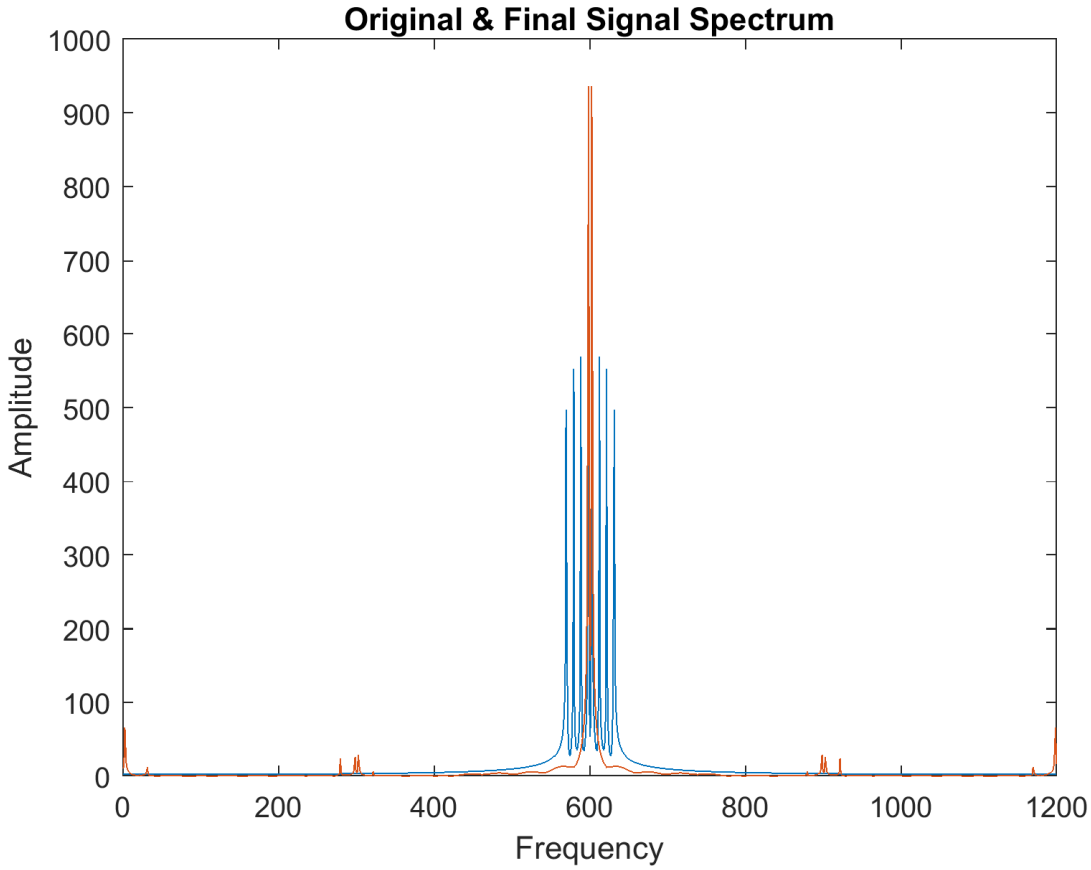

شکل ۱۵




شکل۱۶

بخش ۱-۹)

```
% Generate The Original Signal & plot it's spectrum
t7 = 0:0.01:11.99;
f1 = pi/16;
f2 = 5*pi/16;
f3 = 9*pi/16;
f4 = 13*pi/16;

sig1 = cos(2*pi*f1*t7);
sig2 = cos(2*pi*f2*t7);
sig3 = cos(2*pi*f3*t7);
sig4 = cos(2*pi*f4*t7);
sig = sig1 + sig2 + sig3 + sig4;

figure()
plot(abs(fftshift(fft(sig))));
title("Original & Final Signal Spectrum");
ylabel("Amplitude");
xlabel("Frequency");

% Implort Analysis & Synthesis filter coefficients
coef1 = xlsread('filters.xls', 1);
coef2 = xlsread('filters.xls', 2);

analysis1 = filter(coef1(1,:), 1, sig1); % Analysis Filter
sampledSig1 = downsample(analysis1, 4); % Downsample with 4
pu1 = 2*sampledSig1; % Gain *2
upSig1 = upsample(pu1, 4); % Upsample with 4
synthesis1 = filter(coef2(1,:), 1, upSig1); % Synthesis Filter

analysis2 = filter(coef1(2,:), 1, sig2);
sampledSig2 = downsample(analysis2, 4);
pu2 = 0*sampledSig2; % Gain *0
upSig2 = upsample(pu2, 4);
synthesis2 = filter(coef2(2,:), 1, upSig2);

analysis3 = filter(coef1(3,:), 1, sig3);
sampledSig3 = downsample(analysis3, 4);
pu3 = 1*sampledSig3; % Gain *1
upSig3 = upsample(pu3, 4);
synthesis3 = filter(coef2(3,:), 1, upSig3);

analysis4 = filter(coef1(4,:), 1, sig4);
sampledSig4 = downsample(analysis4, 4);
pu4 = 0.5*sampledSig4; % Gain *0.5
upSig4 = upsample(pu4, 4);
synthesis4 = filter(coef2(4,:), 1, upSig4);

% Generate Output of Filter Bank & Plot it's Spectrum
final = synthesis1 + synthesis2 + synthesis3 + synthesis4;
hold on;
plot(abs(fftshift(fft(final))));
```




شکل ۱۷

در این بخش سیگنالی متشکل از جمع ۴ سیگنال کسینوسی ایجاد شده و طیف آن در حوزه فرکانس رسم شده است. سپس هرکدام از این ۴ سیگنال، با ضرایب import شده از فایل اکسل، فیلتر شده (آنالیز)، با ضریب ۴ downsample شده و عملیات حسابی‌ای که صورت سوال خواسته را بر روی هرکدام اعمال کردیم. سپس با ضریب ۴ upsample انجام داده و بار دیگر با ضرایب دیگری، آن‌ها را فیلتر کرده (سنتز) و نهایتا همه را با هم جمع کردیم تا سیگنال نهایی ایجاد شود. حال طیف این سیگنال را هم برروی نمودار قبلی رسم کردیم.