به نام خدا

گزارش کار آزمایشگاه پردازش سیگنال دیجیتال

آزمایش ۱

آیدین روزبه - ۹۹۲۳۰۳۷

پاییز ۱۴۰۲

کد مربوط به بخش های ۱ تا ۶ در فایل part_1.m آورده شده است.
کد مربوط به بخش ۷ و ۹ در فایل part_2.m آورده شده است.
تابع singen در یک فایل با همین نام به صورت مجزا تعریف شده است.

در ادامه به توضیح و گزارش بخش به بخش این آزمایش می‌پردازیم:

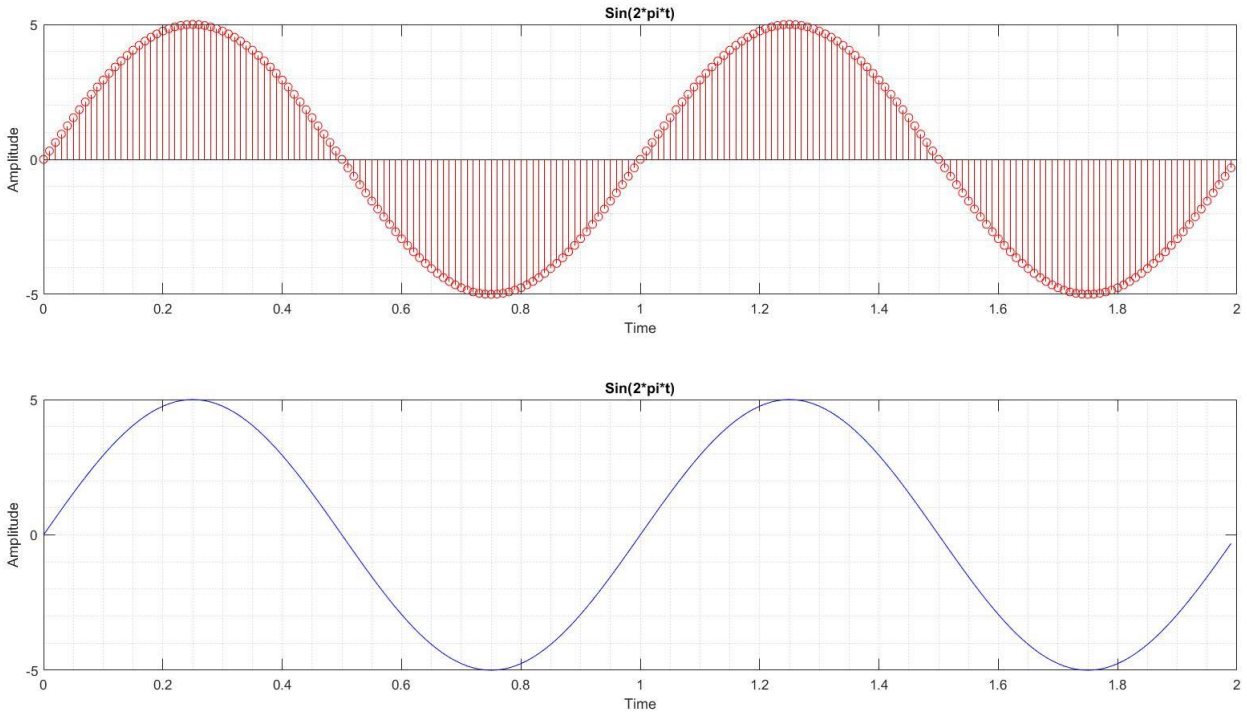
بخش ۱-۱)

```
%% 1-1
close all; clear; clc;

t=0:0.01:1.99;
A=5;
x=A*sin(2*pi*t);

figure(1)
subplot(2,1,1);
stem(t,x, color='red');
xlabel("Time");
ylabel("Amplitude");
title("Sin(2*pi*t)");
grid minor;

subplot(2,1,2);
plot(t,x, color='blue');
xlabel("Time");
ylabel("Amplitude");
title("Sin(2*pi*t)");
grid minor;
```



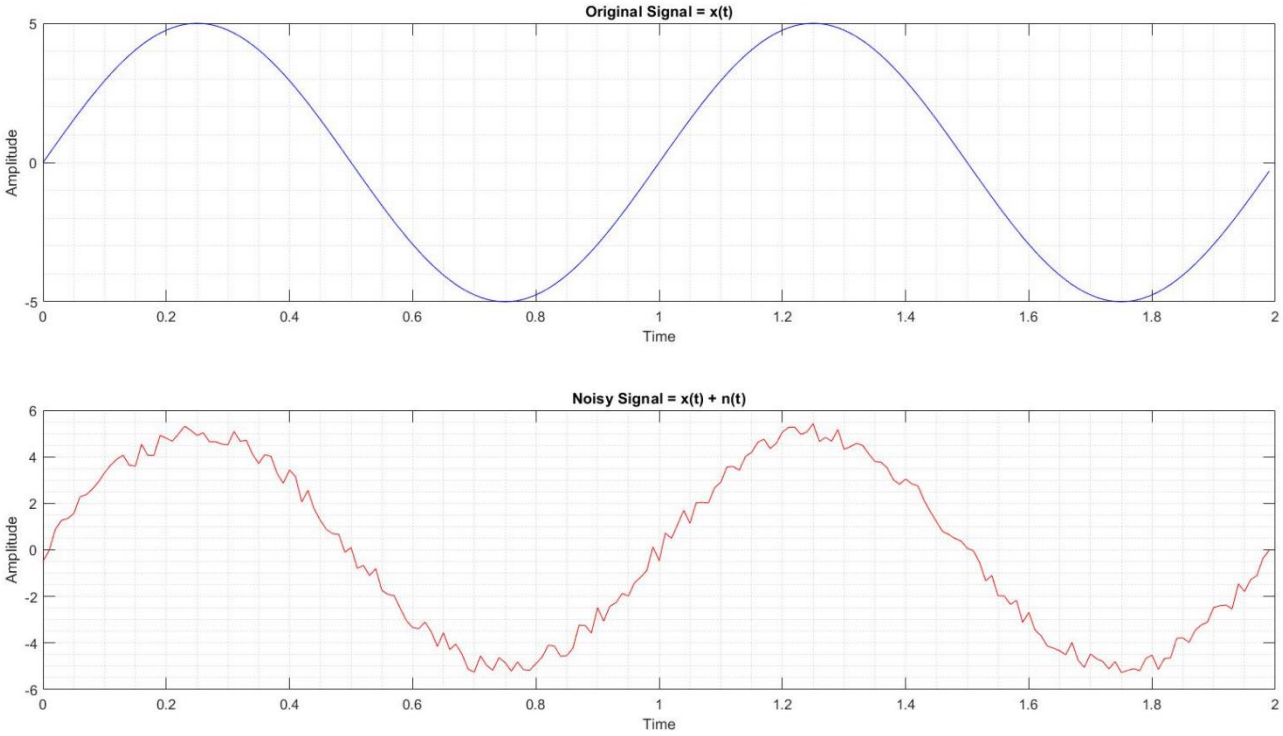
شکل ۱-۱

بخش ۲-۱)

در این بخش برای تعریف نویز از تابع rand استفاده شده است که خروجی آن، نویز با توزیع یکنواخت در بازه 0 تا 1 می باشد. بدین ترتیب لازم است تا عدد 0.5 از آن کاسته شود تا میانگین نویز حاصل، صفر باشد.

```
%% 1-2)
% Introducing noise
n=rand(1,200)-0.5;
xn = x + n;
figure(2);
subplot(2,1,1);
plot(t,x,color='blue');
title("Original Signal = x(t)");
xlabel("Time");
ylabel("Amplitude");
grid minor;

subplot(2,1,2);
plot(t,xn,color='red');
title("Noisy Signal = x(t) + n(t)");
xlabel("Time");
ylabel("Amplitude");
grid minor;
```

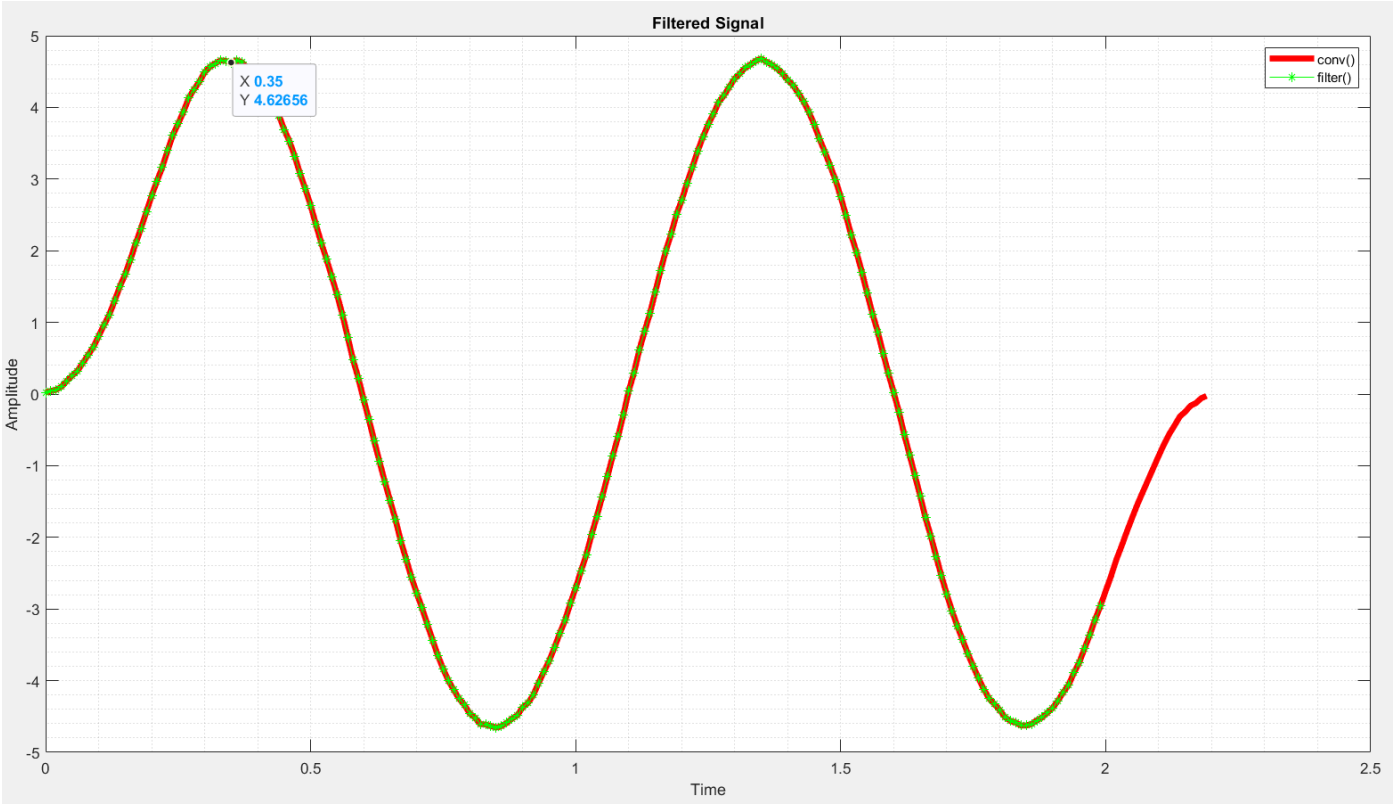


شکل ۲-۱

بخش ۳-۱ و ۴-۱)

نتیجه عبور سیگنال از یک فیلتر میانگین گیر (moving average) با استفاده از کانولوشن همچنین دستور فیلتر در یک نمودار به شکل زیر نشان داده شده اند.

```
%% 1-3 and 1-4
% Moving average
ma = ones(1,21)/21;
num=ones(1,21)/21;
den=1;
res_fil=filter(num,den,xn);
res=conv(xn,ma);
t1 = 0:0.01:2.19;
figure(3);
plot(t1,res,LineWidth=4,color='red');
hold on;
plot(t,res_fil,Marker='*',color='green');
xlabel("Time");
ylabel("Amplitude");
title("Filtered Signal");
legend("conv()","filter()");
disp(max(res_fil)); disp(min(res_fil));
disp(max(res)); disp(min(res));
grid minor;
```



شکل ۳-۱

در نمودار بالا، خط قرمز نشان دهنده خروجی دستور conv و نقاط سبز نشان دهنده خروجی filter هستند. مشاهده می شود که این دو خروجی با دقت بسیار خوبی بر هم منطبق هستند. تفاوت اصلی در آن جا است که خروجی دستور فیلتر، بر خلاف conv، طولی برابر با طول سیگنال اصلی (در اینجا ۲۰۰ سمپل) دارد. اما طول خروجی دستور conv برابر با ۲۲۰ سمپل است (طول سیگنال به علاوه طول سیگنال منهای یک).

سیگنال moving average یک فیلتر FIR با فاز خطی بوده که موجب ایجاد تاخیر در پاسخ می‌شود. همانگونه که در شکل بالا مشخص شده، بزرگ ترین مقدار در سیگنال خروجی در سمپل شماره ۳۵ ثبت شده است، با این که در سیگنال ورودی ماکسیمم خروجی در سمپل ۲۵ رخ می‌داد. از آنجایی که طول سیگنال moving average برابر با ۲۱ سمپل است، همانگونه که انتظار می رفت، به اندازه نصف آن یعنی ۱۰ سمپل، تاخیر به سیستم اضافه شد

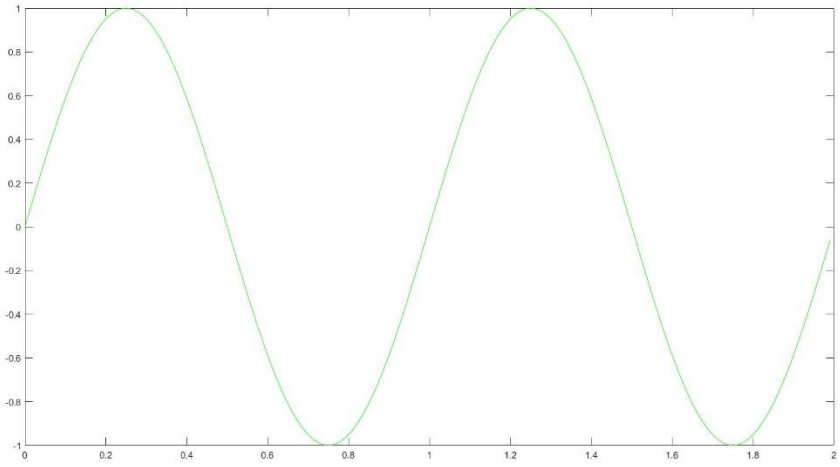
بخش ۵-۱)

تابع singen طبق خواسته سوال، تعریف شده و یک نمونه از آن در کد اصلی نمایش داده شده است.

```
function res = singen(w,n)

array=0:1:n-1;
array=array./n;
res=sin(w.*array);
```

```
%% 1-5
% Function is defined in the 'singen' file. example below
temp=singen(4*pi, 200);
figure(4);
plot(t,temp,color='green');
```



شکل ٤-١

بخش ۶-۱)

```
%% 1-6
F=500;
f=5;
L=4;
t2=0:1/F:L-1/F;
t2_sample=1/f:1/f:L;
x1 = singen(2*pi , L*F) + singen(8*pi , L*F) + singen(12*pi , L*F);
x1_sample = zeros(1,L*f);
for i=1:1:L*f
    x1_sample(i)=x1(i*(F/f));
end
x1_sample_fil = lowpass(x1_sample,6,5);

figure(5);
plot(t2,x1,color='red', LineWidth=2 , LineStyle='--');
hold on;
stem(t2_sample , x1_sample , color='green',LineWidth=2);
plot(t2_sample , x1_sample_fil , LineWidth=1.5);
legend("Original Signal" , "Sampled Sequence" , "Reconstructed Signal");
grid minor;
```

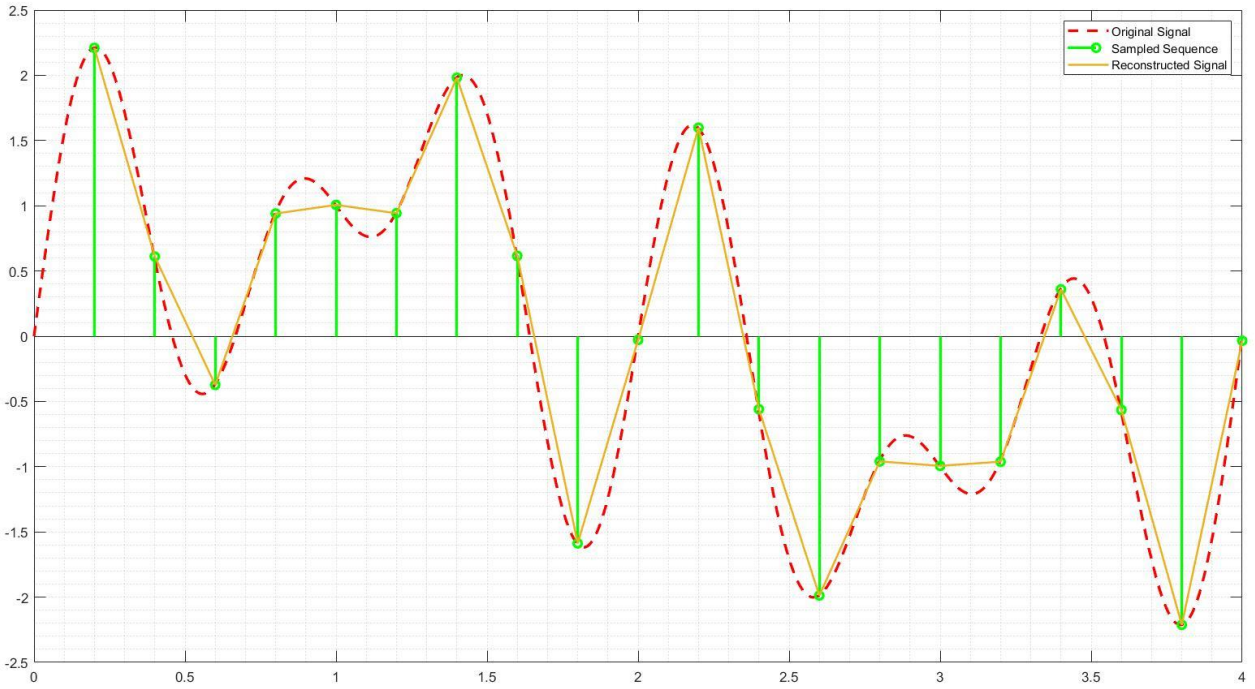
(توجه: در اینجا فرکانس ها به کیلوهرتز و زمان به میلی ثانیه نرمالیزه شده است)

برای نزدیک تر شدن ماتریس در متلب به سیگنال اصلی، با فرکانس ۵۰۰ که بسیار بالاتر از فرکانس های مد نظر سوال است، سیگنال را شبیه سازی می کنیم. (F=500)

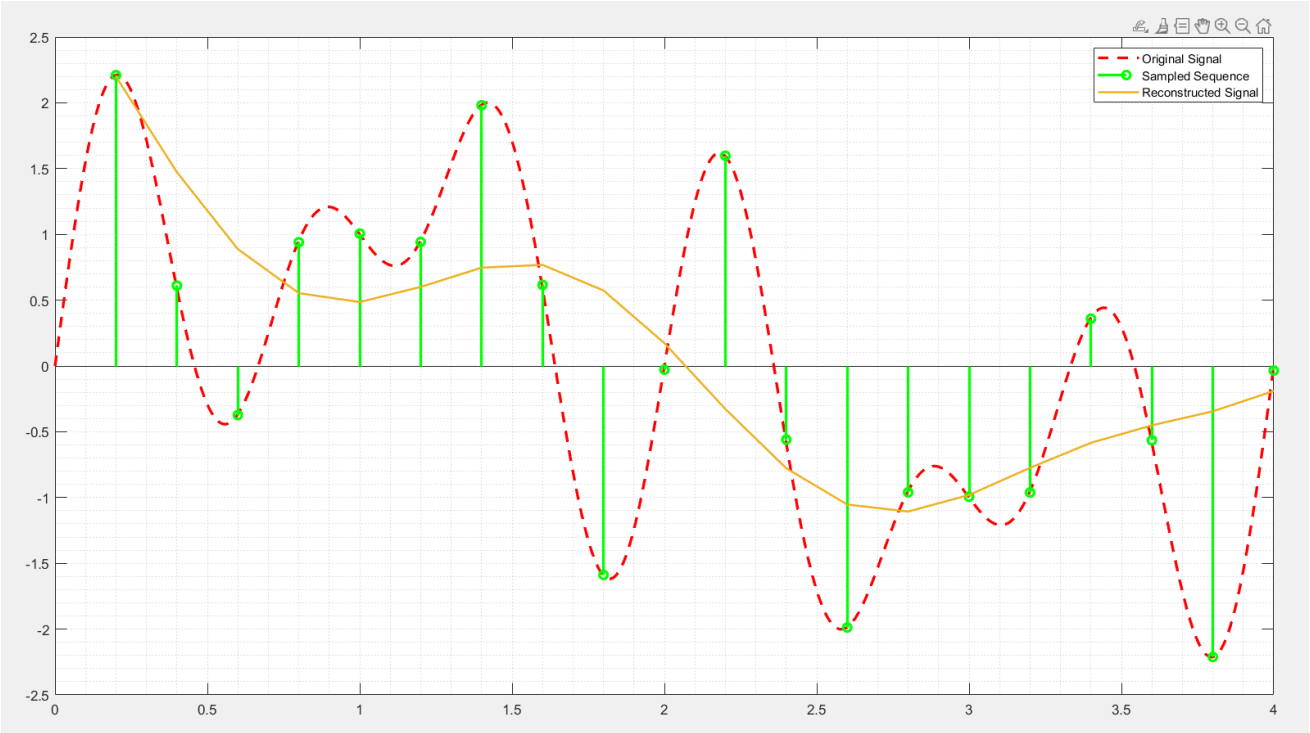
سپس این سیگنال را با فرکانس 5 کیلوهرتز (f=5) نمونه برداری می کنیم.

سپس با دستور lowpass سیگنال نمونه برداری شده را بازسازی می‌کنیم.

در شکل زیر مشاهده می شود که پس از بازسازی سیگنال از روی نمونه ها، با توجه به عدم رعایت نرخ نایکوییست، محتوای فرکانس بالا از دست رفته است. با مشاهده سیگنال بازسازی شده در حوالی مقادیر ۱ و ۳، واضح است که این سیگنال، نسبتا صاف بوده و مانند سیگنال اولیه تغییرات ندارد؛ تغییرات در این محدوده دقیقا مرتبط با هارمونیک ۶ کیلوهرتز می باشد که در این فرایند از دست رفته است. در تصویر بعدی، فرکانس قطع از 6 به 0.6 تغییر کرده است. این تغییر منجر به آن شده که سیگنال فیلتر شده، دیگر سیگنال ورودی را track نکند و بسیار از تناوب ها و تغییرات در سیگنال اصلی از دست رفته اند.


Original Signal
Sampled Sequence
Reconstructed Signal
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
0
0.5
1
1.5
2
2.5
3
3.5
4

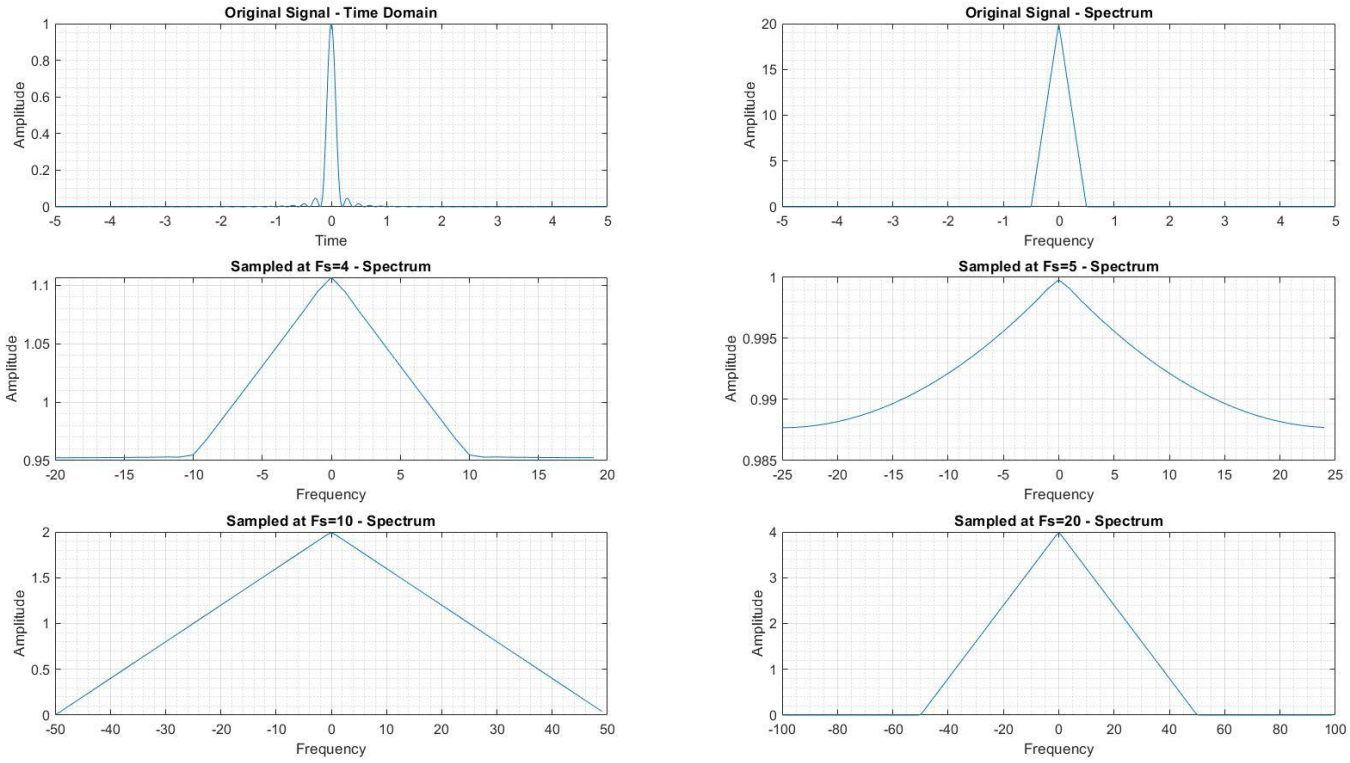
شکل ۵-۱


Original Signal
Sampled Sequence
Reconstructed Signal
2.5
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2.5
0
0.5
1
1.5
2
2.5
3
3.5
4

شکل ۶-۱

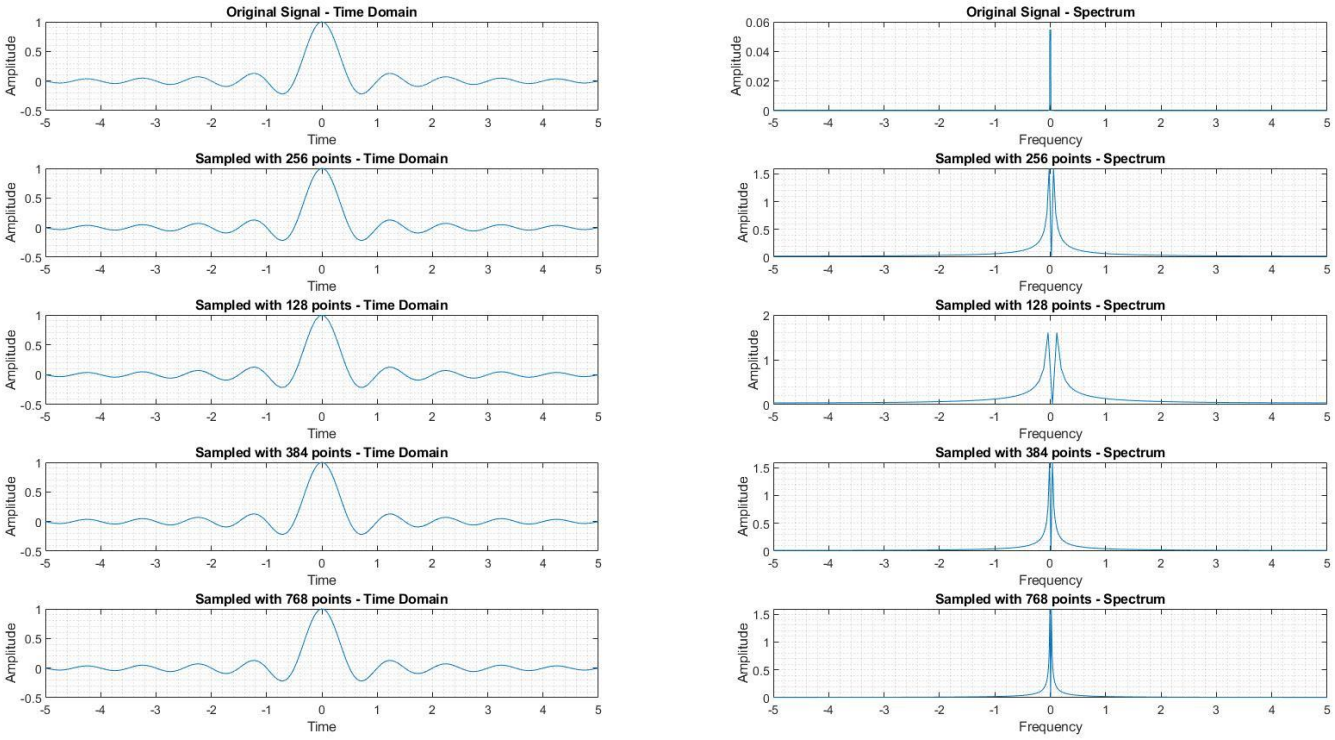
بخش ۷-۱)

طبق خواسته سوال، این بخش نیز پیاده سازی شده است. به دلیل طولانی بود کد این بخش، از آوردن آن به طور مستقیم در گزارش کار خودداری می‌کنیم. نتیجه شبیه سازی با فرکانس های نمونه برداری مختلف در شکل زیر نمایش داده است. به جز سیگنال اصلی، برای سایر موارد، فقط طیف سیگنال ترسیم شده است.



بخش ۸-۱)

سیگنال مورد نظر با تعداد نقاط مختلف نمونه بردار شده است (به ترتیب: ۲۵۶، ۱۲۸، ۳۸۴ و ۷۶۸). در هر مورد سیگنال نمونه برداری شده به همراه طیف آن رسم شده است.



لازم به ذکر است در هر بخش برای نرمالیزه کردن طیف، تبدیل آن سیگنال به اندازه خود تقسیم شده است (1/N).