

# گزارش تمرین کامپیوتری 5 درس سیگنال‌ها و سیستم‌ها

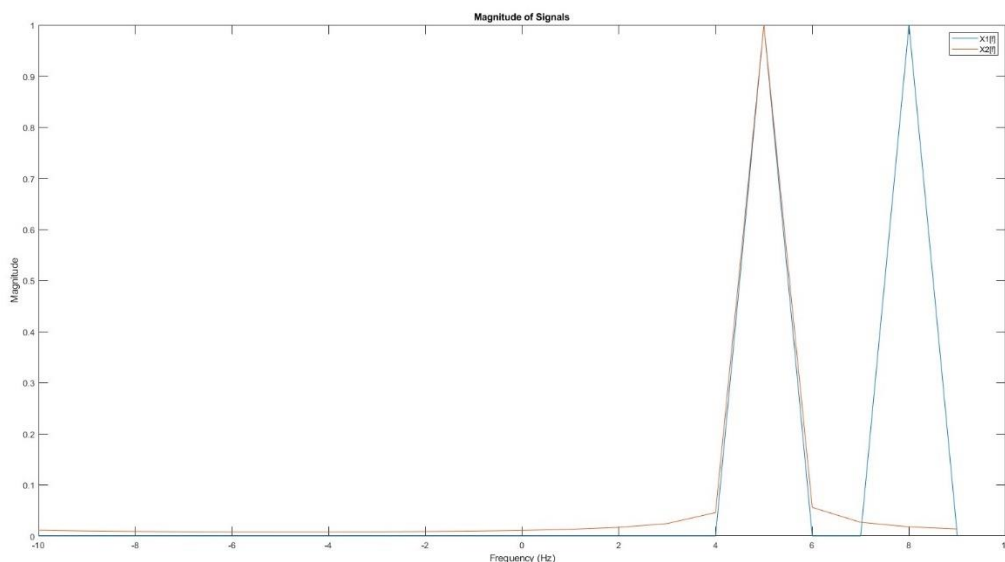
بابک حسینی محتشم 810101408

محمّدسینا پرویزی مطلق 810101394

1403/9

بخش اول:

۰- دو سیگنال  $x_1$  و  $x_2$  خواسته شده را در حوزه فرکانس رسم می‌کنیم و همان طور که انتظار داشتیم در  $x_1$  دو قله به طول 5 و 8 و در  $x_2$  تنها یک قله به طول 5 مشاهده می‌کنیم.



تصویر اسکرپیت:

```

fs=20;
T = 1;
N = fs * T;
t = linspace(0, T-T/N, N);
f = linspace(-fs/2, fs/2-fs/N, N);
x1=exp(1j*2*5*pi*t)+exp(1j*2*8*pi*t);
x2=exp(1j*2*5*pi*t)+exp(1j*2*5.1*pi*t);
X1 = fftshift(fft(x1));
X1=X1/max(abs(X1));
X2 = fftshift(fft(x2));
X2=X2/max(abs(X2));
plot(f,abs(X1));
hold on;
plot(f,abs(X2/max(abs(X2))));
title('Magnitude of Signals');
xlabel('Frequency (Hz)');
legend(['X1[f]';'X2[f]'])
ylabel 'Magnitude'
hold off;

```

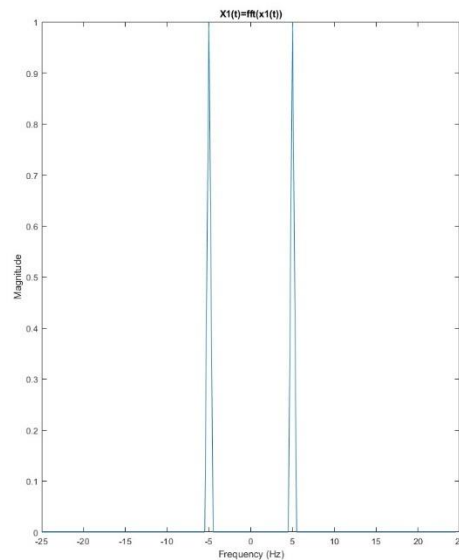
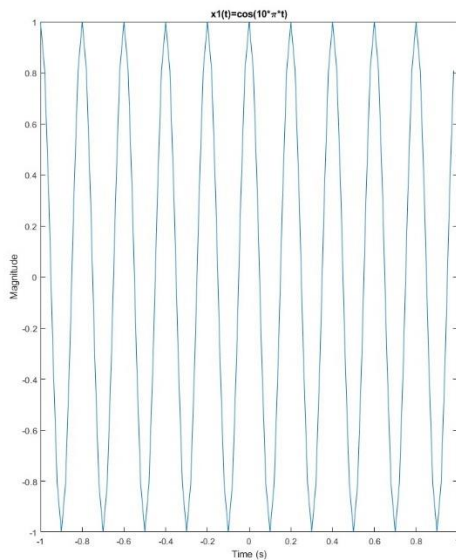
۱- معادله خواسته شده را تولید و رسم می‌کنیم. تصویر اسکرین‌پست:

```

fs=50;
T = 2;
N = fs * T;
t = linspace(-1, T-1-T/N, N);
f = linspace(-fs/2, fs/2-fs/N, N);
x1=cos(10*pi*t);
subplot(1,2,1);
plot(t,x1);
title('x1(t)=cos(10*\pi*t)');
xlabel('Time (s)');
ylabel 'Magnitude'
subplot(1,2,2);
X1=fftshift(fft(x1));
X1=X1/max(abs(X1));
plot(f,abs(X1));
title('X1(t)=fft(x1(t))');
xlabel('Frequency (Hz)');
ylabel 'Magnitude'

```

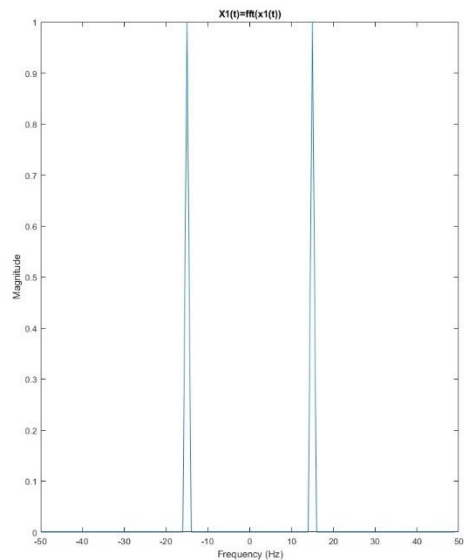
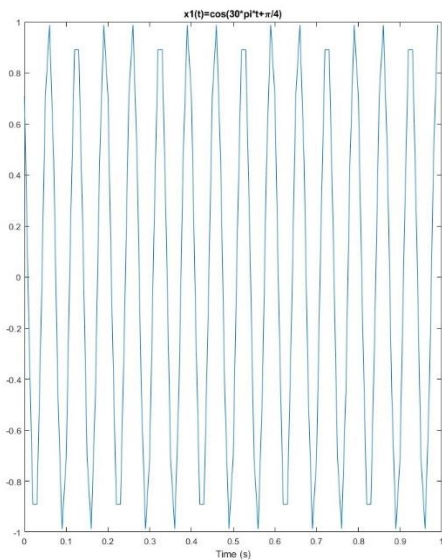
در خروجی همان طور که انتظار داشتیم تنها دو قله در تبدیل فوریه  $X1$  وجود دارد درحالی‌که در حوزه زمان شکل متناوب را مشاهده می‌کنیم.



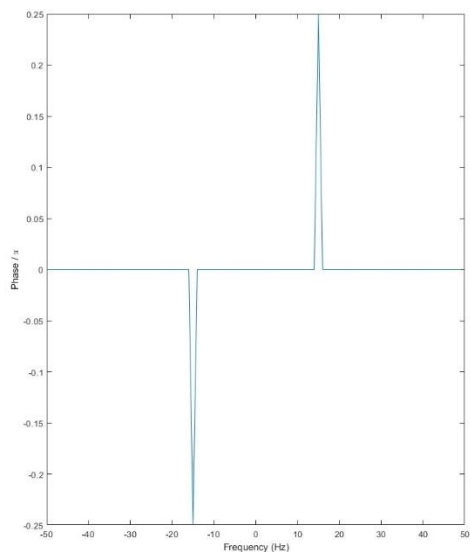
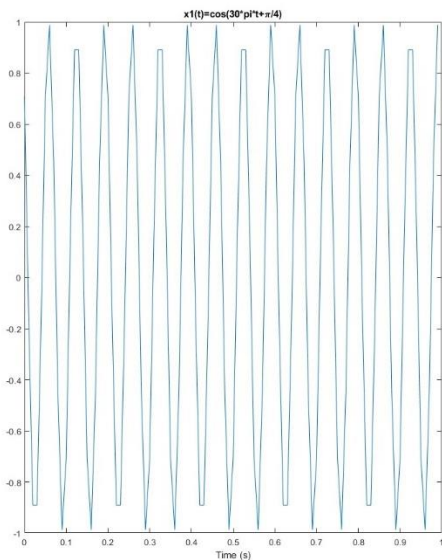
۲- سیگنال خواسته شده را تولید کرده و از اندازه وفاز آن خروجی می‌گیریم. تصویر اسکرپت:

```
fs=100;
T = 1;
N = fs * T;
t = linspace(0, T-T/N, N);
f = linspace(-fs/2, fs/2-fs/N, N);
x1=cos(30*pi*t+pi/4);
subplot(1,2,1);
plot(t,x1);
title('x1(t)=cos(30*pi*t+\pi/4)');
xlabel('Time (s)');
subplot(1,2,2);
X1=fftshift(fft(x1));
X1=X1/max(abs(X1));
plot(f,abs(X1));
title('X1(t)=fft(x1(t))');
ylabel 'Magnitude'
xlabel('Frequency (Hz)');
tol = 1e-6;
X1(abs(X1) < tol) = 0;
theta = angle(X1);
plot(f,theta/pi)
xlabel 'Frequency (Hz)'
ylabel 'Phase / \pi'
```

می‌توان دید که فرکانس خروجی مشابه قسمت قبل است:



ولی با رسم فاز دو قله مشخص که نشان دهنده زاویه یا همان فاز ضرایب مختلط رابطه است:



## بخش دوم:

۱- از کد تمرین قبل استفاده می‌کنیم. هر حرف را تبدیل به رشته باینری کرده و در فایل Mapset ذخیره می‌کنیم:

```

len=32;
Mapset=cell(2,len);
for i=1:len
    Mapset{1,i}=char('a'+i-1);
    Mapset{2,i}=dec2bin(i-1,ceil(log2(len)));
end
Mapset{1,27}=char(' ');
Mapset{1,28}=char('.');
Mapset{1,29}=char(',');
Mapset{1,30}=char('!');
Mapset{1,31}=char(';');
Mapset{1,32}=char('"');
save Mapset Mapset;

```

۲- از تابع `coding_amp` تمرین قبل استفاده می‌کنیم و آن را تغییر می‌دهیم تا به جای دامنه، فرکانس را کد کند. بدین صورت عمل می‌کنیم که در هر ثانیه سیگنال سینوسی که ارسال می‌کنیم، ضریبی از `bit_rate` عدد متوالی در رشته باینری را شامل می‌شود. برای `bit_rate` یک ما تصمیم گرفتیم برای بیشتری اختلاف ممکن، فرکانس ۱ و ۴۹ را انتخاب کنیم و همین طور برای `bit_rate` های بالاتر نیز فرکانس‌ها را تا حد امکان بین ۱ تا ۴۹ پخش می‌کنیم که این کار با ضرب  $fs/2-1$  که همان ۴۹ است در عدد مورد نظر انجام می‌شود.  
تصویر اسکرپ:

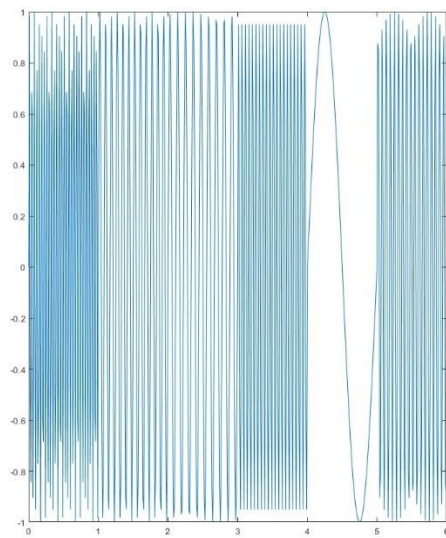
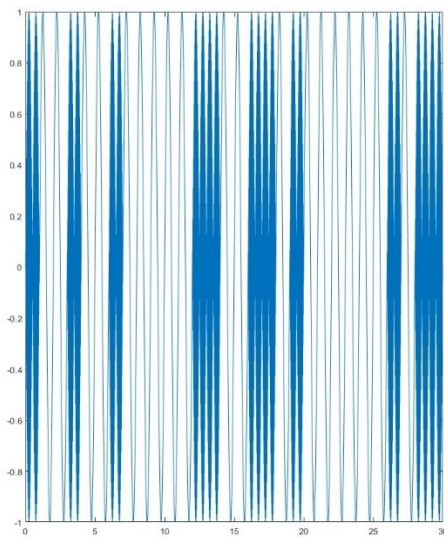
---

```

function coded_msg=coding_freq(msg,bit_rate)
    load Mapset.mat Mapset
    fs=100;
    ts=1/fs;
    bin='';
    for i=1:strlength(msg)
        c=extract(msg,i);
        bin=strcat(bin,Mapset{2,strcmp(Mapset(1,:), c)});
    end
    t=0:ts:strlength(bin)/bit_rate-ts;
    coded_msg=zeros(size(t));
    f=(fs/2-1)/(2^bit_rate-1);
    i=1;
    tsin=0:ts:1-ts;
    for j=1:bit_rate:length(bin)-bit_rate+1
        val=bin2dec(bin(j:j+bit_rate-1));
        fi=floor(val*f);
        if(fi==0)
            fi=fi+1;
        end
        coded_msg((i-1)*fs+1:i*fs)=sin(2*pi*tsin*fi);
        i=i+1;
    end
    % plot(t,coded_msg);
end

```

۳- می‌توان در تصویر خروجی سیگنال کد شده برای کلمه **signal**، اختلاف فرکانس قسمت‌های مختلف سیگنال را مشاهده کرد که به دلیل تفاوت انتخاب فرکانس‌های ۰ و ۴۹ و اختلاف زیاد این دو، در تصویر کاملاً مشخص است.



تصویر اسکرپت:

```
clc;clearvars;
msg='signal';
ts=1/100;
coded=coding_freq(msg,1);
subplot(1,2,1);
t=0:ts:strlen(msg)*5-ts;
plot(t,coded);
subplot(1,2,2);
coded=coding_freq(msg,5);
t=0:ts:strlen(msg)-ts;
plot(t,coded);
```

۴- از تابع دیکد دامنه تمرین قبلی استفاده می‌کنیم و آن را برای دیکد کردن فرکانس به جای دامنه تغییر می‌دهیم. این تابع با بررسی حداکثر فرکانس‌های تبدیل فوریه سیگنال، فرکانسش را پیدا کرده و تبدیل به رشته باینری و سپس رشته باینری را تبدیل به پیام می‌کند.

تصویر اسکرپت:

```

function decoded_msg=decoding_freq(coded,bit_rate)
    load Mapset.mat Mapset
    bit_len_map=strlength(Mapset{2, 1});
    decoded_msg='';
    fs=100;
    % ts=1/fs;
    % tsin=0:ts:1-ts;
    f=(fs/2-1)/(2^bit_rate-1);
    bin='';
    for i=1:fs:length(coded)-fs+1
        fft_coded=fftshift(fft(coded(i:i+fs-1)));
        max_f_val=0;
        max_f_idx=0;
        for j=0:(2^bit_rate-1)
            fi=floor(j*f);
            if(fi==0)
                fi=fi+1;
            end
            if(max_f_val<abs(fft_coded(fi+fs/2+1)))
                max_f_val=abs(fft_coded(fi+fs/2+1));
                max_f_idx=j;
            end
        end
        bin=strcat(bin,dec2bin(max_f_idx,bit_rate));
    end
    index=1;
    for i=1:bit_len_map:strlength(bin)-bit_len_map+1
        decoded_msg=strcat(decoded_msg,Mapset{1,strcmp(Mapset(2, :), bin(i:i+bit_len_map-1))});
        index=index+1;
    end
end

```

می‌توان دید در هر دو حالت **bit\_rate** یک و پنج پیام به درستی رمزگشایی می‌شود.

```

1      clc;clearvars;
2      msg='signal';
3      coded=coding_freq(msg,1);
4      decoding_freq(coded,1)
5      coded=coding_freq(msg,5);
6      decoding_freq(coded,5)

```

Command Window

ans =

'signal'

ans =

'signal'



۵- با نویز با واریانس  $0.0001$  همچنان در هر دو `bit_rate` تابع پیام را به درستی رمزگشایی میکند.

```
1  clc;clearvars;
2  msg='signal';
3  coded=coding_freq(msg,1);
4  noise_std=0.01;
5  noise=noise_std*randn(1,length(coded));
6  decoding_freq(coded+noise,1)
7  coded=coding_freq(msg,5);
8  noise=noise_std*randn(1,length(coded));
9  decoding_freq(coded+noise,5)
```

---

Command Window

```
ans =  
  
    'signal'  
  
ans =  
  
    'signal'
```

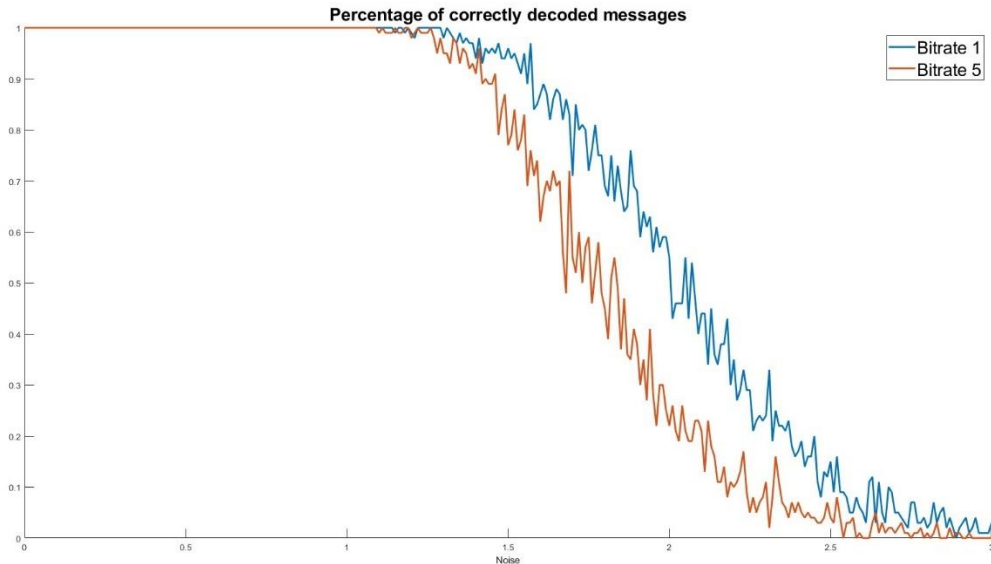
۶- حال اسکرپتی مینویسیم که نویزهای مختلف با واریانس  $0$  تا  $9$  را به سیگنال اضافه کند تا مقاوم بودن تابع رمزگشایی نسبت به نویز با قدرت مختلف در دو حالت چقدر است  
تصویر اسکرپت:

```

clc;clearvars;
msg='signal';
t=0:0.01:3;
n_test=100;
correct_bit_rate1=zeros(size(t));
correct_bit_rate5=zeros(size(t));
idx=0;
for noise_std=t
    idx=idx+1;
    for j=1:n_test
        coded=coding_freq(msg,1);
        noise=noise_std*randn(1,length(coded));
        decoded_msg=decoding_freq(coded+noise,1);
        if(strcmp(msg,decoded_msg))
            correct_bit_rate1(idx)=correct_bit_rate1(idx)+1;
        end
        coded=coding_freq(msg,5);
        noise=noise_std*randn(1,length(coded));
        decoded_msg=decoding_freq(coded+noise,5);
        if(strcmp(msg,decoded_msg))
            correct_bit_rate5(idx)=correct_bit_rate5(idx)+1;
        end
    end
end
correct_bit_rate1=correct_bit_rate1/n_test;
correct_bit_rate5=correct_bit_rate5/n_test;
figure;
hold on;
plot(t,correct_bit_rate1,'LineWidth',2);
plot(t,correct_bit_rate5,'LineWidth',2);
lgd=legend('Bitrate 1','Bitrate 5');
lgd.FontSize = 20;
tle=title('Percentage of correctly decoded messages');
tle.FontSize=20;
xlabel('Noise');
hold off;

```

تصویر خروجی:



۷- می‌توان دید همان طور که انتظار داشتیم عملکرد تابع رمزگشایی در `bit_rate` یک بهتر از پنج است که این به دلیل قابل تشخیص بودن فرکانس‌های مختلف به دلیل تنوع کمتر در خوزه فرکانس است. می‌توان دید که در `bit_rate` پنج تا نویز با انحراف معیار  $1/68$  همچنان بیشتر از نیمی از اوقات پیام به درستی رمزگشایی میشود در حالیکه در `bit_rate` یک با نویز با انحراف معیار  $2$  بیش از نیمی از اوقات رمزگشایی درست صورت میگیرد.

۹- خیر. زیرا با افزایش نرخ نمونه برداری، تاثیر نویز روی فرکانس‌هایی که نمونه برمی‌داریم تغییر نمی‌کند ولی با افزایش پهنای باند فرکانس‌های مورد نظر قابل تشخیص می‌شوند و درصد خطا کاهش میابد.