به نام خدا

پروژه پنجم درس سیگنال‌ها و سیستم‌ها

فاطمه‌زهرا برومندنیا-۸۱۰۱۰۰۰۹۴

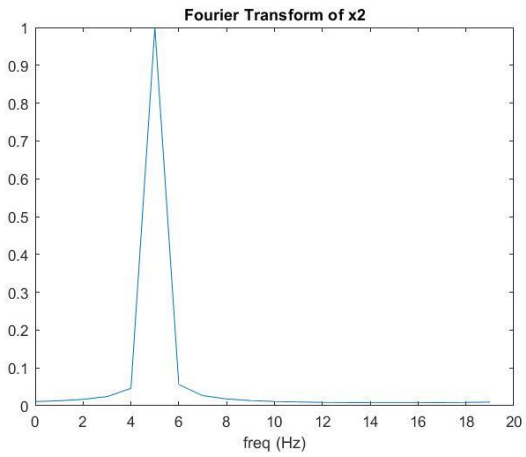
بخش اول

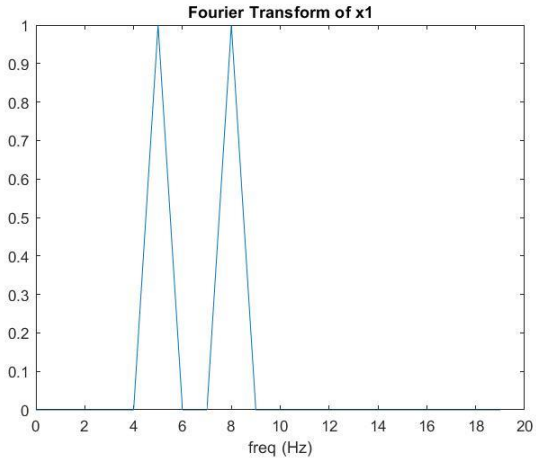
۱-۰)

هنگامی که فرکانس‌های تابع نمایی را به ترتیب برابر با ۵ و ۸ در نظر می‌گیریم، دو پیک به وضوح در فرکانس‌های ۵ و ۸ مشاهده می‌شود. اما وقتی که این مقادیر را به ترتیب ۵ و ۵.۱ در نظر گرفته شده، به دلیل اینکه تفاوت بین آنها کمتر از مقدار رزولوشن فرکانس، به عبارت دیگر ۱ هرتز است، فقط یک پیک با کمی نویز خواهیم دید.

Variables - Threshold_arr

```
fs = 20;
t_start = 0;
t_end = 1;
ts = 1 / fs;
t = t_start:ts:t_end - ts;
N = length(t);
f = 0:(fs / N):((N - 1) * fs / N);
x1 = exp(1j * 2 * pi * 5 * t) + exp(1j * 2 * pi * 8 * t);
figure
plot(f, abs(fft(x1)) / max(abs(fft(x1))))
xlabel('freq (Hz)')
title('Fourier Transform of x1')


x2 = exp(1j * 2 * pi * 5 * t) + exp(1j * 2 * pi * 5.1 * t);
figure
plot(f, abs(fft(x2)) / max(abs(fft(x2))))
xlabel('freq (Hz)')
title('Fourier Transform of x2')
```





۱-۱ الف و ۱- ۲ الف)

Editor - p1_1.m

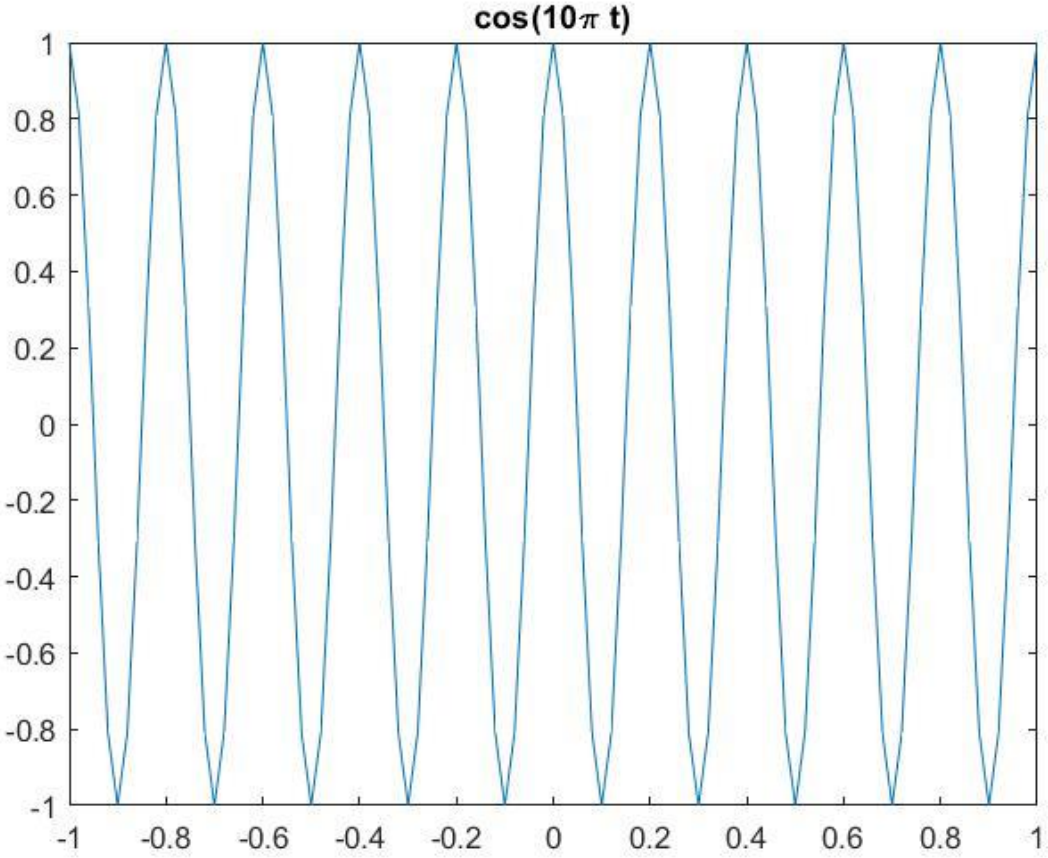
```
clc,close all
fs=50;

tstart=-1;
tend=1;
T=tend-tstart;
N=T*fs;
t=tstart:1/fs:tend;
x1=cos(10*pi*t);

plot(t,x1);

figure

y=fftshift(fft(x1));
F=y/max(abs(y));

f=-(fs/2):(fs/N):(fs/2)-(fs/N)
f=[f f(1)]
plot(f,abs(F));
```

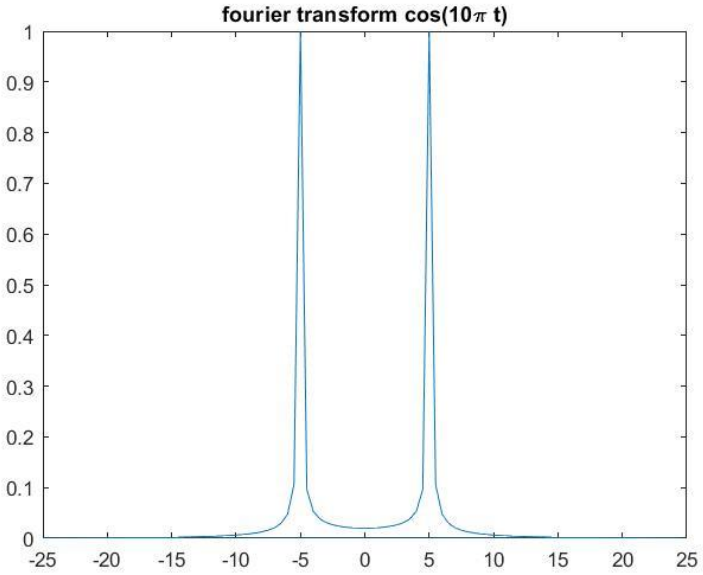
p0.m | p1_1.m | p1_2.m | p2.m

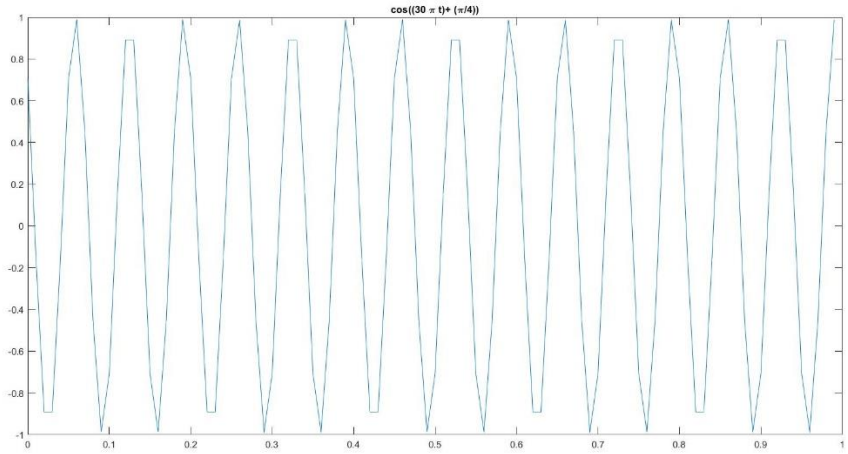
Editor - p1_2.m

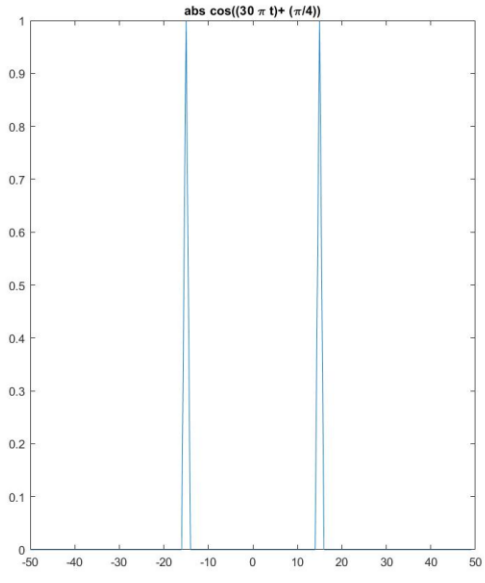
```
fs = 100;
t_start = 0;
t_end = 1;
ts = 1 / fs;
t = t_start:ts:t_end - ts;
N = length(t);
f = (-fs / 2):(fs / N):(fs / 2 - fs / N);
x1=cos((30*pi*t)+(pi/4));

plot(t,x1);

figure

y=fftshift(fft(x1));
F=y/max(abs(y));

subplot(1,2,1);
plot(f,abs(F));
title('abs');
subplot(1,2,2);

tol=1e-6;
F(abs(F)<tol)=0;
theta = angle(F);
plot(f,theta/pi);
title('angle');
xlabel('Frequency (Hz)');
ylabel('Phase/ \pi');
```

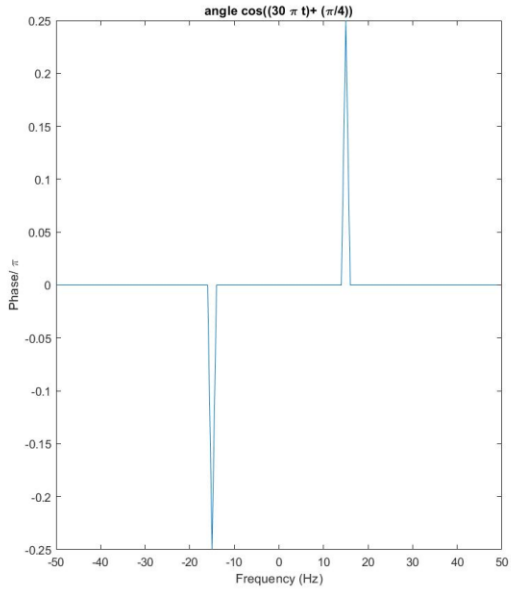
p0.m | p1_1.m | p1_2.m | p2.m | coding_freq.m


cos(10π t)
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1


fourier transform cos(10π t)
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-25
-20
-15
-10
-5
0
5
10
15
20
25







۱-۱-ج)تبدیل فوریه تابع کسینوسی با فرکانس f_0 به صورت زیر است:

$$X(f) = \frac{1}{2} [\delta(f - f_0) + \delta(f + f_0)]$$

حالا، اگر تابع $x(t)$ خود را به صورت $x(t) = \cos(2\pi f_0 t)$ تعریف کنیم، تبدیل فوریه آن به صورت زیر خواهد بود:

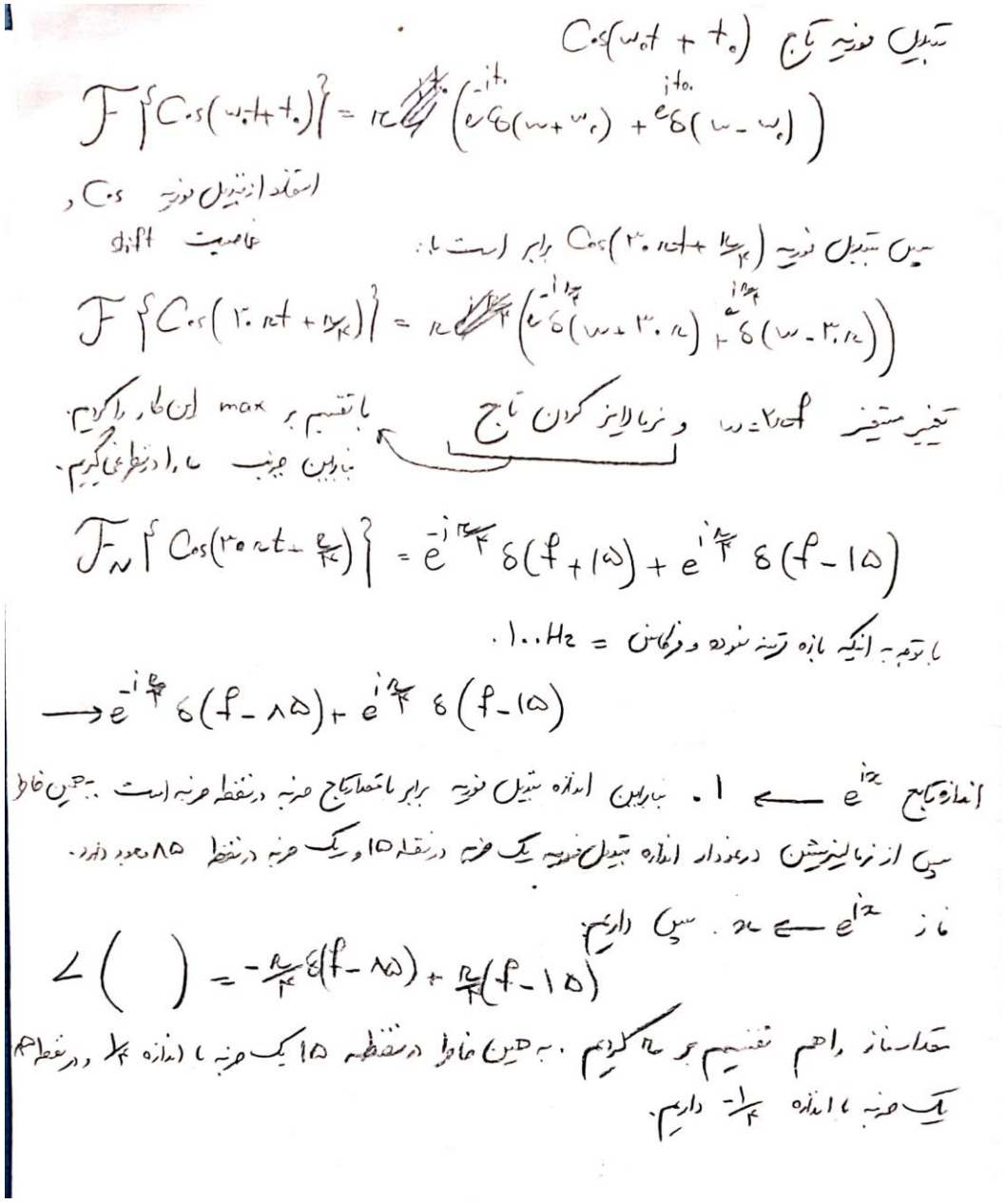
$$X(f) = \frac{1}{2} [\delta(f - f_0) + \delta(f + f_0)]$$

از طرفی نمودارها به جای اینکه بر اساس ωرسم شده باشند، بر حسب ffرسم شدهاند. در نتیجه باید این تغییر متغیر را نیز لحاظ کنیم:

$$\omega = 2\pi ff \longrightarrow \mathcal{FNN}\{\cos(10\pi tt)\} = \delta(ff - 5) + \delta(ff + 5)$$

در این کد، یک سیگنال کسینوسی با فرکانس $f_0 = 5$ هرتز ایجاد شده. با توجه به تبدیل فوریه کسینوسی که توضیح داده شد، انتظار داریم در نمودار خروجی، دو نقطه با فرکانس مثبت و منفی حول $f_0 = 5$ هرتز داشته باشیم.

۱-۲-ج)



بخش دوم

۲-۱ و ۲-۲)

Editor - p2.m

```
1  clear, close all
2  Alphabet='abcdefghijklmnopqrstuvwxyz .,!";';
3
4  numChar=length(Alphabet);
5  mapset=cell(2,numChar);
6  for i=1:numChar
7      mapset{1,i}=Alphabet(i);
8      mapset{2,i}=dec2bin(i-1,5);
9  end
10 msg='signal';
11 bitRate=5;
12 outSig=coding_freq(msg,mapset,bitRate);
13 figure
14 plot(1:length(outSig),outSig);
15 title1=sprintf('bitrate=%d', bitRate);
16 title(title1);
17 decodedMsg=decoding_freq(outSig,mapset,bitRate);
18
```

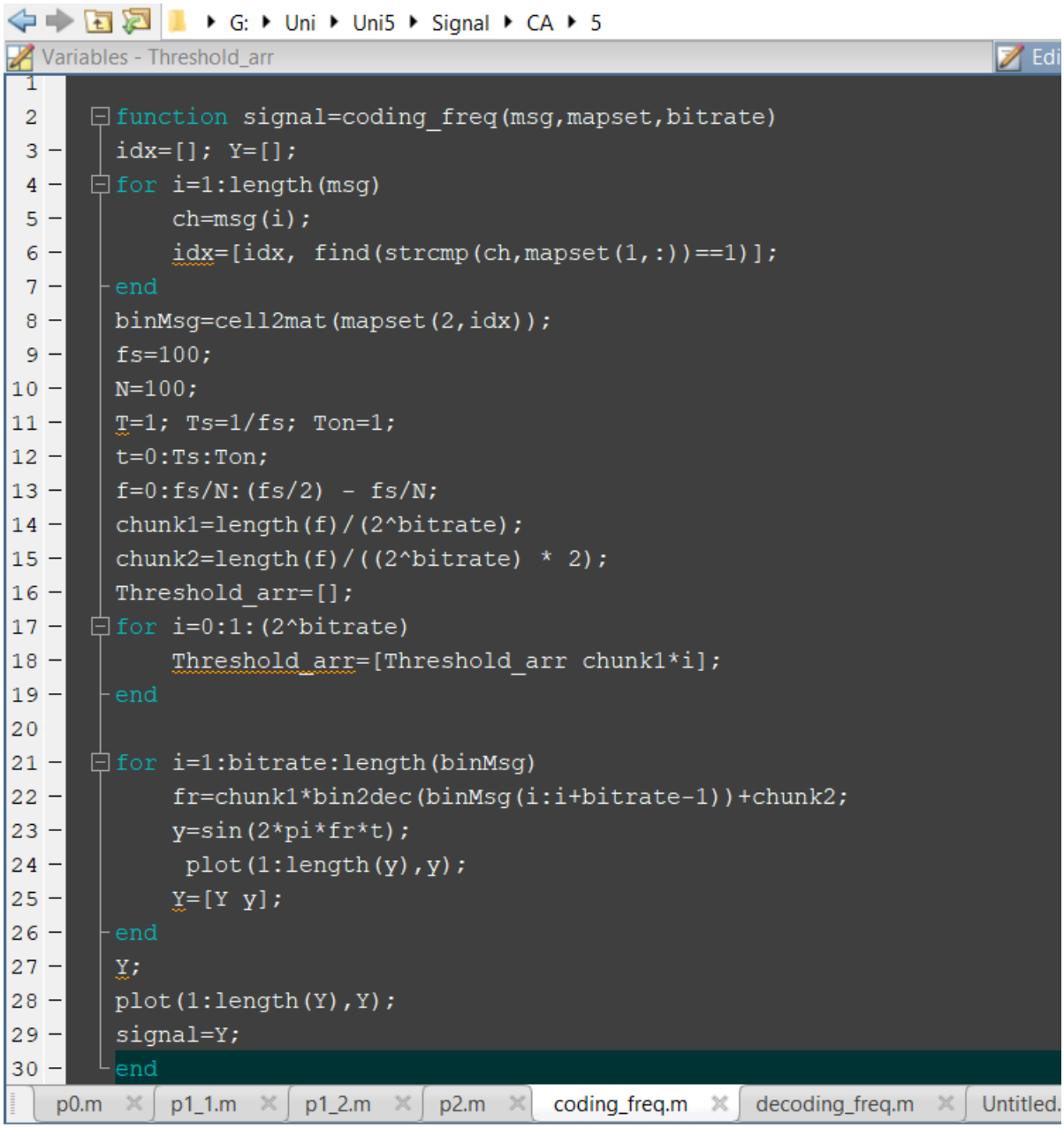
۲-۳) این تابع ابتدا ایندکس‌های مربوط به هر کاراکتر را در `mapset` پیدا می‌کند و آنها را در `idx` ذخیره می‌کند. سپس پیام را به صورت دودویی در `binMsg` ذخیره می‌کند.

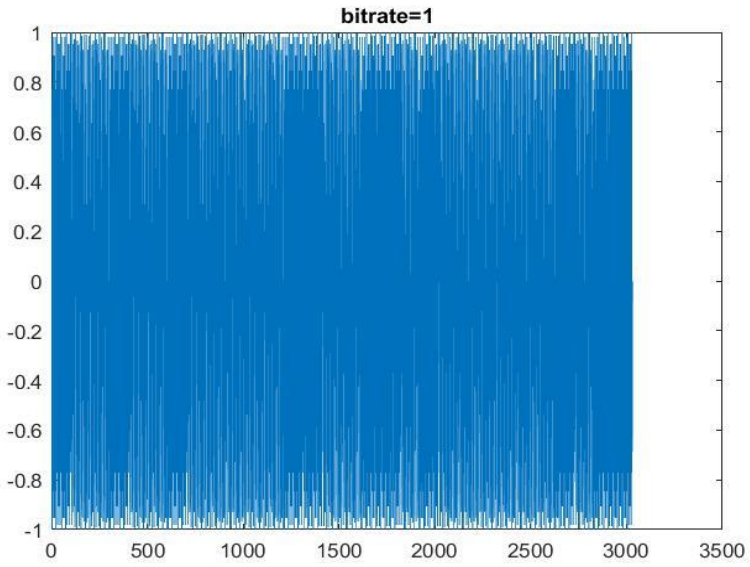
سپس، تعداد نمونه‌های سیگنال را با استفاده از `fs` (فرکانس نمونه‌برداری) و `Ton` (مدت زمان سیگنال) تعیین می‌کند. سپس یک حلقه بر روی بیت‌های پیام ایجاد می‌کند و باید فرکانس متناظر با آن بیت/بیت‌ها را بسازیم، سپس سیگنال سینوسی را با استفاده از آن فرکانس تولید کنیم و به سیگنال تولیدی کانکت کنیم.

برای این کار، ابتدا بازه فرکانسی مثبت را، به تعداد عددهای مختلفی که با آن بیت‌ریت تولید می‌شود می‌شکنیم (bitrate^2)، متغیر chunk1 به همین منظور تولید شده. این‌ها در دیکودکردن به عنوان آستانه تشخیص به کار می‌روند. حالا برای تخصیص فرکانس به هر عدد، از این بازه‌ها وسط‌شان را انتخاب کرده و اختصاص می‌دهیم، که هنگام اضافه‌شدن نویز اگر لب‌مرزی باشند تشخیص سخت می‌شود. به خاطر همین در متغیر chunk2 نصف chunk1 را نگه می‌داریم.حالا فرکانس متناظر با هر عدد، برابر می‌شود با آن عدد در chunk1 به علاوه chunk2.

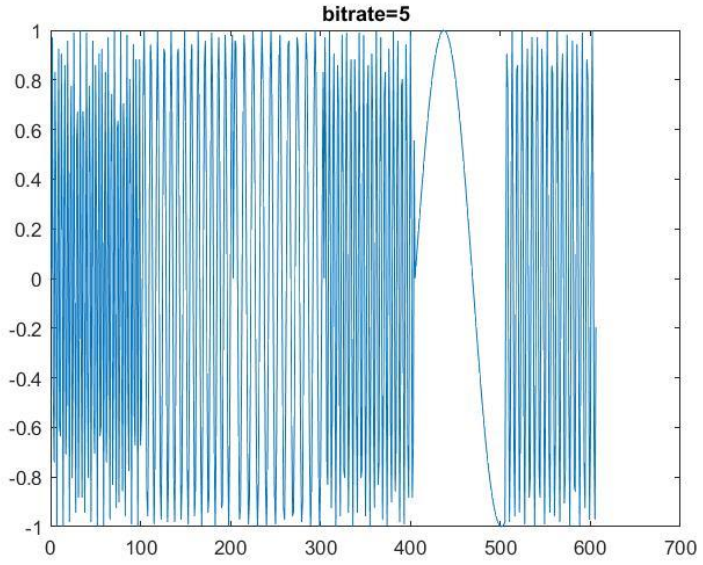
سیگنال نهایی با ادغام سیگنال‌های تولید شده در هر بیت در `Y` ساخته می‌شود.

در نهایت، سیگنال نهایی در `signal` ذخیره می‌شود و به عنوان خروجی تابع برگردانده می‌شود.


G: ▸ Uni ▸ Uni5 ▸ Signal ▸ CA ▸ 5
Variables - Threshold_arr
function signal=coding_freq(msg,mapset,bitrate)
idx=[]; Y=[];
for i=1:length(msg)
ch=msg(i);
idx=[idx, find(strcmp(ch,mapset(1,:))==1)];
end
binMsg=cell2mat(mapset(2,idx));
fs=100;
N=100;
T=1; Ts=1/fs; Ton=1;
t=0:Ts:Ton;
f=0:fs/N:(fs/2) - fs/N;
chunk1=length(f)/(2^bitrate);
chunk2=length(f)/((2^bitrate) * 2);
Threshold_arr=[];
for i=0:1:(2^bitrate)
Threshold_arr=[Threshold_arr chunk1*i];
end
for i=1:bitrate:length(binMsg)
fr=chunk1*bin2dec(binMsg(i:i+bitrate-1))+chunk2;
y=sin(2*pi*fr*t);
plot(1:length(y),y);
Y=[Y y];
end
Y;
plot(1:length(Y),Y);
signal=Y;
end
p0.m
p1_1.m
p1_2.m
p2.m
coding_freq.m
decoding_freq.m
Untitled.



نرخ بیت ۱



نرخ بیت 5

۲-۴) سیگنال، سیگنال ورودی است که باید رمزگشایی شود.

ابتدا در وکتور threshold_arr به نسبت بیت‌ریت داده شده آستانه تقسیم‌بندی بازه ۰تا۵۰ را می‌سازیم و ذخیره می‌کنیم. در فور لوپ اصلی این تابع، ابتدا پارت‌های یک ثانیه‌ای را از سیگنال جدا می‌کنیم تا روی آن‌ها عملیات دیکود انجام شود. بعد از تبدیل فوریه گرفتن روی آن پارت، چون

سیگنال ارسالی سیگنال سینوسی با فرکانس‌های مشخصی بوده، تبدیل فوریه آن در دو نقطه پیک خواهد زد که محل آن به ما نشان می‌دهد فرکانس کد شده چه بوده. بنابراین با کمک تابع max خود متلب، ایندکس‌هایی که در آن تبدیل فوریه پیک زده را پیدا می‌کنیم. حالا فقط مقدار مثبت را می‌خواهیم، به کمک تابع find، از آخر به اولین نتیجه‌ای که رسیده‌ایم و مکس بوده است را برداشته‌ایم تا فرکانس مثبتی که در آن پیک خورده را پیدا کنیم.حالا باید این فرکانس به دست آمده، که نامش found_freq است را با مقادیر Threshold_arr بسنجیم تا ببینیم بین کدام بازه‌ها قرار می‌گیرد و آن‌گاه نتیجه را، به عنوان بیت نتیجه‌شده از آن سیگنال، به رشته binMsg کانکت کنیم.

پس از پایان حلقه، طول رشته binMsg به ۵ تقسیم شده و حروف معادل آن محاسبه می شود. با استفاده از تابع find، شاخص نماد متناظر در آرایه mapset بر اساس مقایسه با ردیف دوم آرایه به دست می آید.

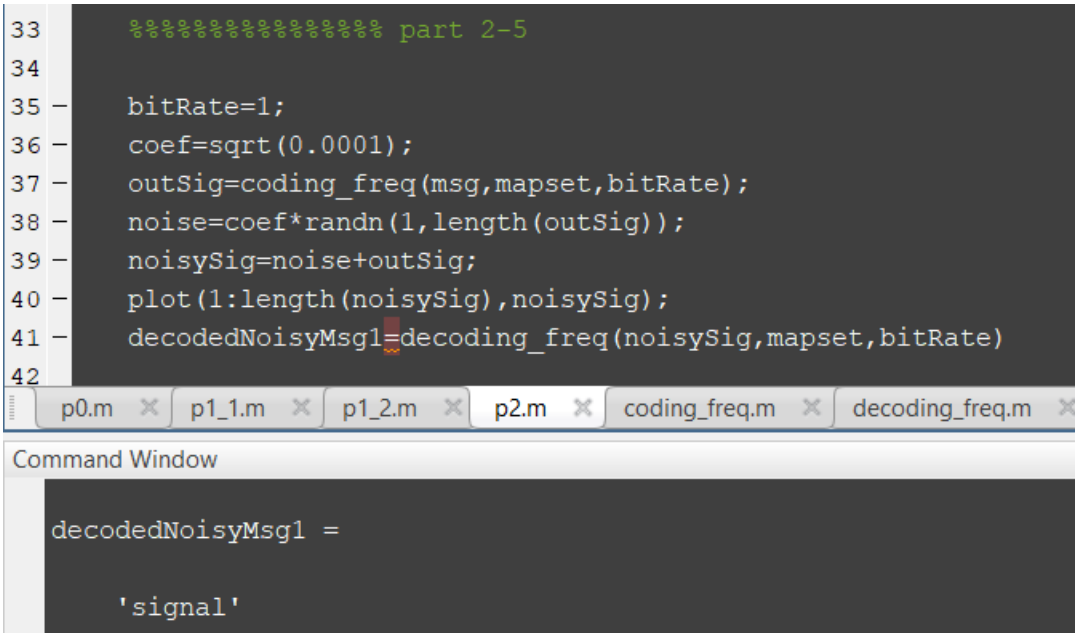
در نهایت، با ادغام مقادیر متناظر از ردیف اول آرایه mapset، پیام رمزگشایی شده به متغیر msg اختصاص داده می شود.

به طور خلاصه، این کد با استفاده از الگوی مطابقت الگو و کورلیشن، سیگنال را رمزگشایی کرده و پیام رمزگشایی شده را بر اساس مجموعه نگاشت بازمی گرداند.

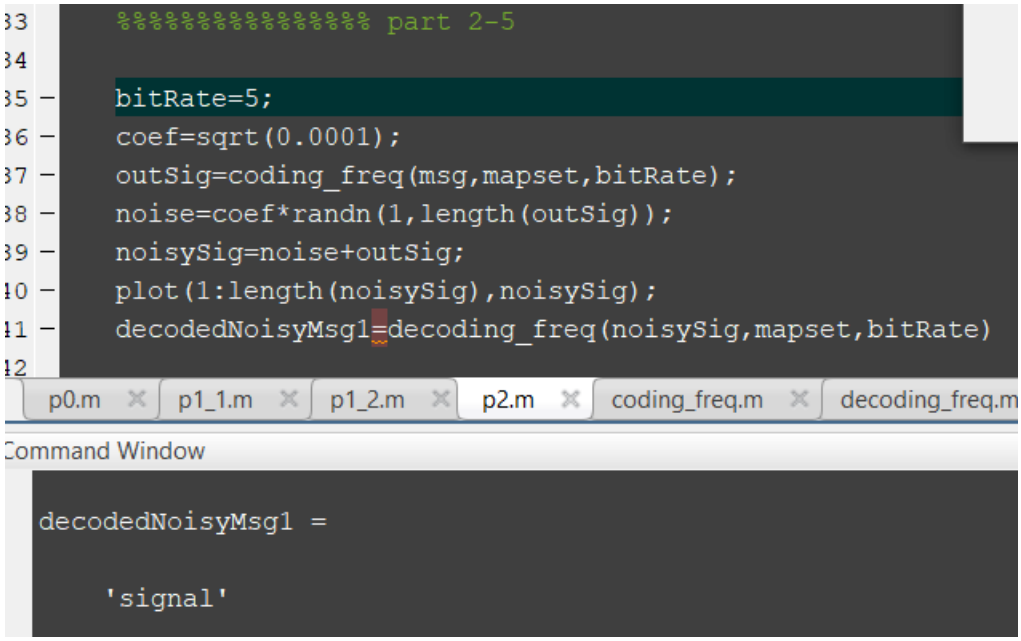
Editor - decoding_amp.m

```
function msg=decoding_amp(signal,mapset,bitrate)
Fs=100;
Ts=1/Fs;
Ton=1;
chunk = round(Ton * Fs);
t=0:Ts:Ton;

binMsg='';
threshold = 2*(2^bitrate-1);

fraction_arr=[];
for i=0:1:(2^bitrate-1)
    fraction=i/(2^bitrate-1);
    fraction_arr = [fraction_arr fraction];
end

for i = 1:chunk:(length(signal) - chunk)
    sig_part = signal(i:i+chunk-1);

        template = 2*sin(2*pi*t);
        correlation = max(0.01*xcorr(sig_part, template));
    m=abs(fraction_arr - correlation);
    [u, idx] = min(abs(fraction_arr - correlation));

     if threshold > u
        binMsg=strcat(binMsg,dec2bin(idx-1,bitrate));
     else
         binMsg=strcat(binMsg,'?');
     end
end
```

۲-۶) برای هر دو نرخ درست تشخیص داده شد. در تصاویر زیر خروجی هرکدام آمده است:



نرخ بیت ۱



نرخ بیت 5

۲-۶) طبق جدول گزارش زیر، به ترتیب بیت‌ریت ۵ و سپس ۱ مقاوم بودند. نه نتایج با آنچه در مقدمه گفته شد همخوانی ندارد،

واریانس/بیت ریت	بیت ریت ۱	بیت ریت ۵
۰.۰۰۱	صحیح	صحیح
۰.۰۱	صحیح	صحیح

۱.۵	صحیح	صحیح
1.7	غلط	صحیح
1.9	غلط	صحیح
2.1	غلط	غلط

۲-۷) برای بیت ریت پنج، تقریبا تا ۱.۹ مقاوم بود و پس از آن نتیجه غلط داد. برای بیت ریت ۱، تا ۱.۷ مقاوم بود و پس از آن با ۱.۸ نتیجه غلط داد. به علت رندوم بودن نویز این موضوع کمی غیرقابل پیش بینی است.

۲-۹)

در کدگذاری فرکانسی، نرخ نمونه‌برداری (fs) و پهنای باند مصرفی (بازه فرکانسی مورد استفاده) دو پارامتر مهم هستند. این دو پارامتر به طور مستقیم با یکدیگر در ارتباط هستند. اگر نرخ نمونه‌برداری افزایش داده شود و پهنای باند مصرفی ثابت بماند، این به معنای افزایش تعداد نمونه‌ها در یک دوره زمانی است که می‌تواند به افزایش دقت در بازسازی سیگنال منجر شود. حالا سناریوهای زیر بررسی می‌شود:

۱. **افزایش نرخ نمونه‌برداری (fs):** افزایش نرخ نمونه‌برداری به وضوح می‌تواند کیفیت بازسازی سیگنال را بهبود بخشد، زیرا اطلاعات بیشتری در دسترس قرار می‌گیرد و جزئیات دقیق‌تری از سیگنال را تشخیص می‌دهد. این امکان داده می‌شود که سیگنال‌های با فرکانس‌های بالا و جزئیات کوچک‌تر را نیز نمونه‌برداری کنید.

۲. **حفظ پهنای باند مصرفی:** اگر پهنای باند مصرفی ثابت بماند، به این معنی است که فرکانس نمونه‌برداری بیشتر برای بازسازی سیگنال استفاده می‌شود اما اطلاعات بیشتری در مورد فرکانس‌های بالا به دست نمی‌آید. این می‌تواند باعث شود که برخی از جزئیات فرکانسی از دست رفته و نتیجه نهایی کیفیت کمتری داشته باشد.

پس در مورد نویز، افزایش نرخ نمونه‌برداری به طور کلی می‌تواند به مقاومت بیشتر در برابر نویز منجر شود.