به نام خدا

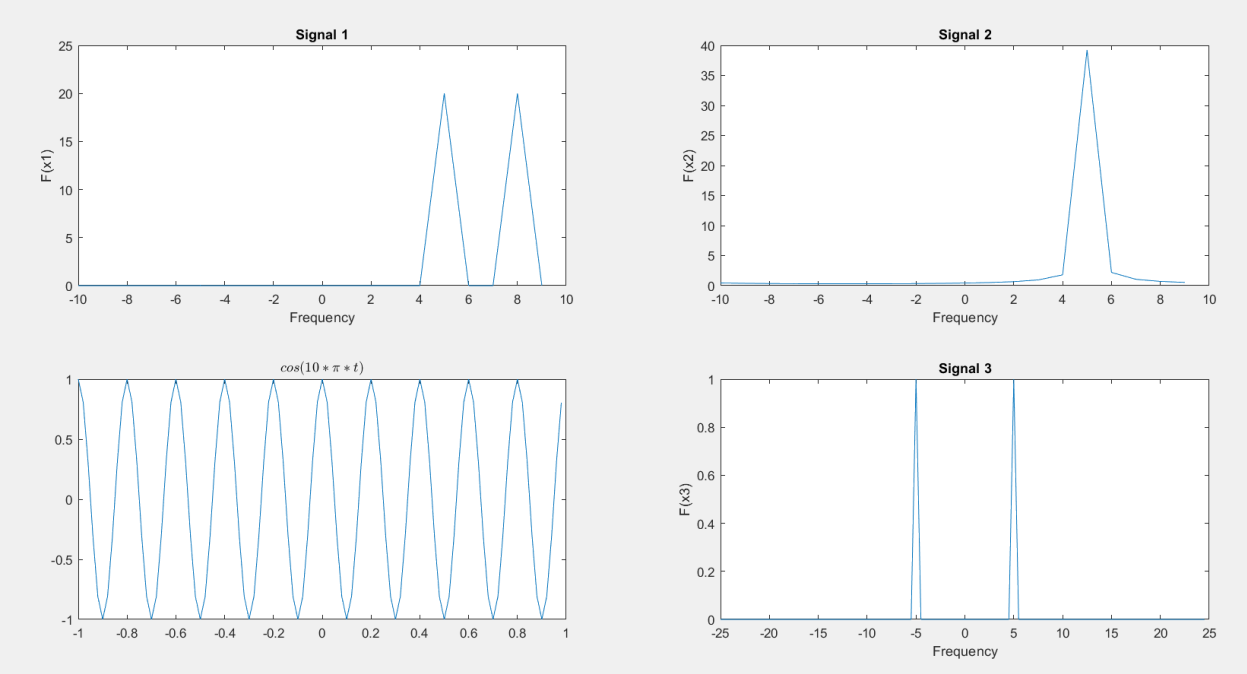
CA 5

امیرعلی شهریاری (۸۱۰۱۰۰۱۷۳)

۱-۱:

رزولوشن فرکانسی قدرت تفکیک فرکانسی در حوزه فوریه را نشان می دهد در نمودار اولیه ، که اختلاف تک تن آن بیش از ۱ است ، در حوزه فوریه دو قله در فرکانس های ۸ و ۵ مشاهده می شود و در قسمت دوم، چون فاصله بین فرکانس دو سیگنال کمتر از T1/=(1) است، پس قدرت تفکیک این دو سیگنال را نداریم و صرفا یک قله در فرکانس ۵ هرتز مشاهده می شود (اختلاف بین دو فرکانس کمتر از ۱ است).

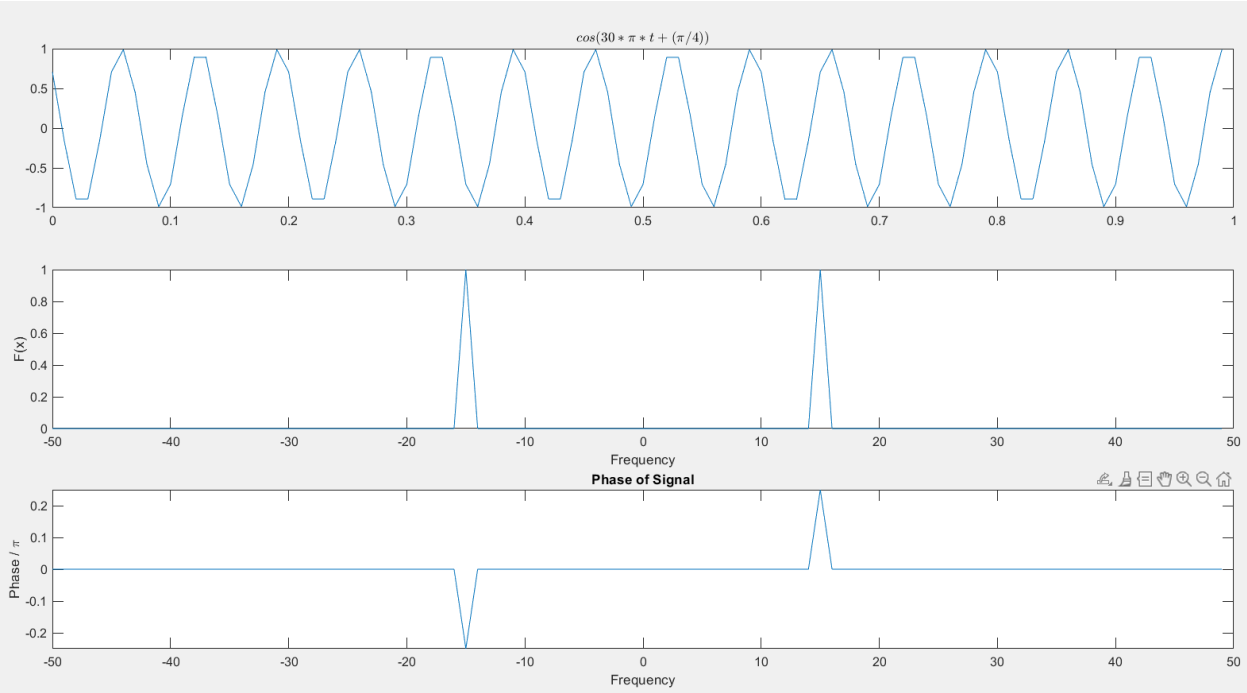
در این بخش می خواهیم سیگنال x1(t)=cos(10t) را به توجه به اطلاعات داده شده ، نمونه برداری کنیم در اینجا فرکانس fs=5 و t3=-1:0.02:0.98 به صورت متغیر تعریف می شود؛ با کمک دستور y3 = fftshift(fft(x3)) سیگنال فوق را به حوزه فوریه می بریم و با متغیر f3 مطابق روابط داده شده تعریف می شود و آنرا در حوزه فوریه ترسیم می کنیم و از آنجایی که مقادیر تبدیل فوریه را به ماکس آن تقسیم کرده ایم مقدار پیک ها ۱ است.



۱-۲:

اینبار می خواهیم سیگنال x2(t)=cos(30 πt+ $\pi/4$) را نمونه برداری کنیم ؛ مجدد دو متغیر t و x را تعریف میکنیم و آنها را پلات کرده . با کمک دستور fftshift(fft(x)) سیگنال را به حوزه فوریه برده وذخیره می کنیم و با کمک متغیر f تعریف

شده ، آنرا در حوزه فوریه ترسیم می کنیم؛ همانگونه که مشاهده می شود دارایی بخش حقیقی و موهومی است و در دو فرکانس - ۱۵ و ۱۵ پیک می زند.



۲:

۲-۱: در قسمت نخست این پروژه به مانند پروژه قبلی ، یک مپ ست برای پیاممان درست می کنیم و در سطر اول آن حروف کوچک انگلیسی ، فاصله ، ویرگول ، نقطه ، ! ، ; ، " است و به کمک جدول اسکی ، حرف هرکدام را به آن وصل می کند؛ همچنین در حلقه فور دیگر اعداد دسیمال ۰ تا ۳۲ را به اعداد باینری ۵ بیتی با دستور dec2bin تبدیل می کند و در سطر دوم مپ ست ذخیره سازی می کند.

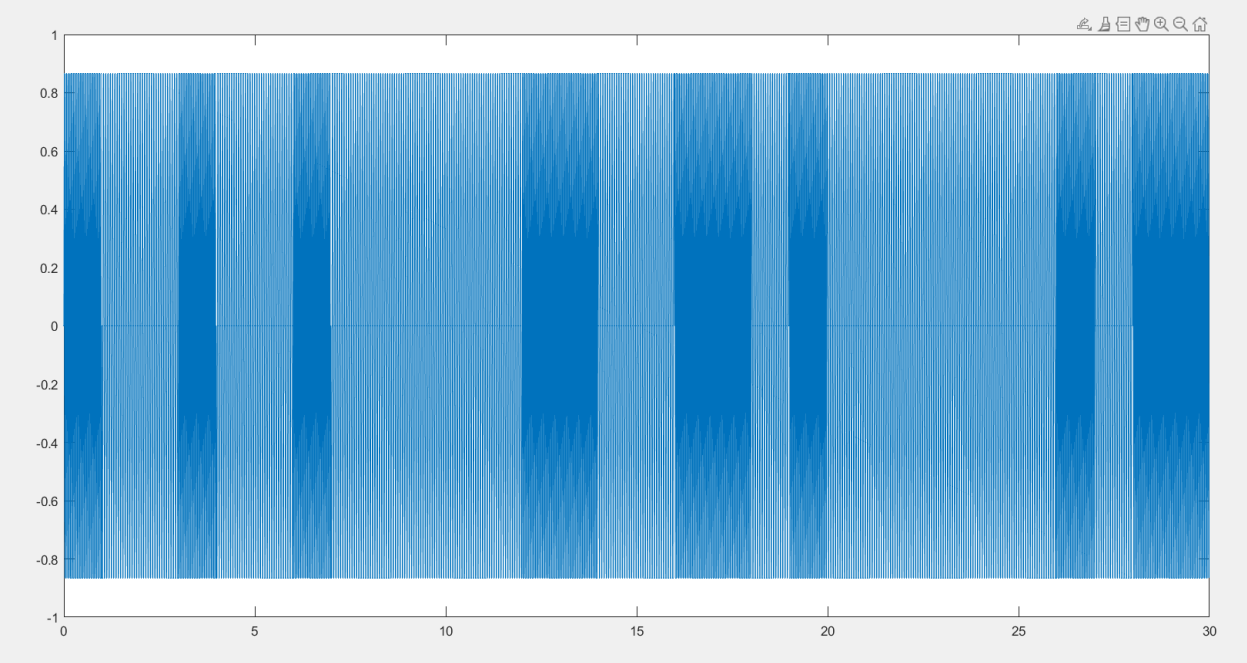
```
function SignalMapping = generate_signal_mapping()
    SignalMapping = cell(2, 32);
    alphabet = char(97:122);

    for idx = 1:26
        SignalMapping(1, idx) = cellstr(alphabet(idx));
    end

    specialChars = {' ', '.', ',', '!', ':'};
    for idx = 1:length(specialChars)
        SignalMapping(1, 26 + idx) = cellstr(specialChars{idx});
    end

    for idx = 1:32
        SignalMapping(2, idx) = cellstr(dec2bin(idx-1, 5));
    end
end
```

۲-۲:

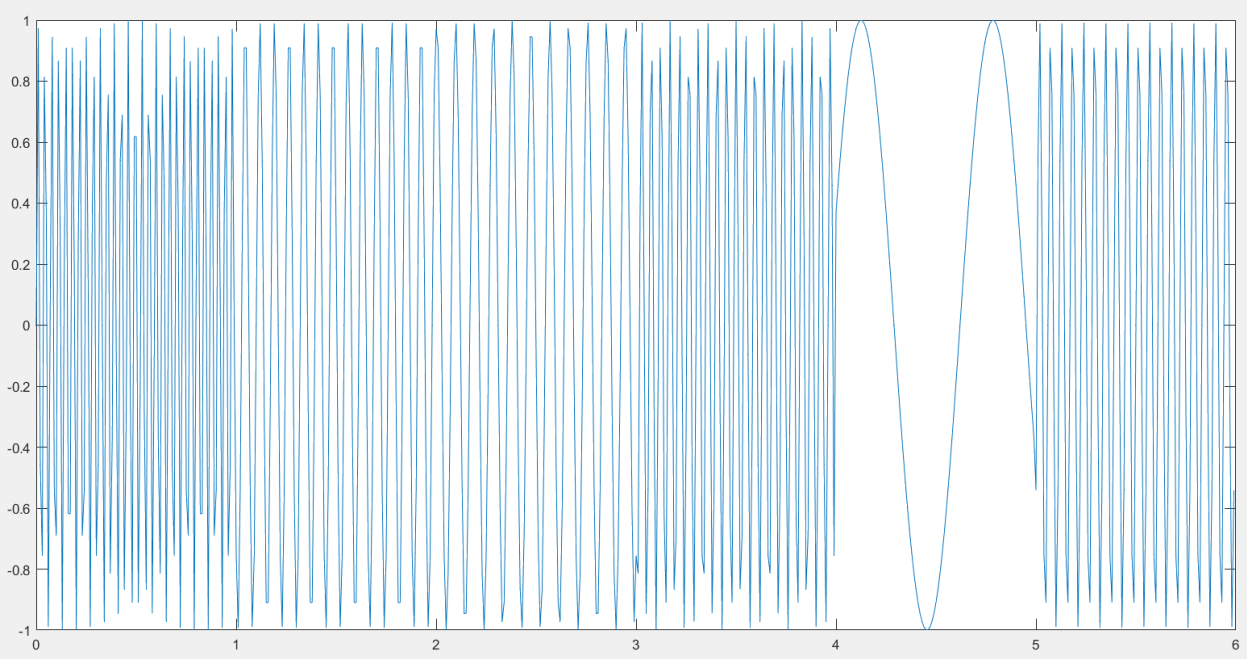
در این بخش میخواهیم تابع انکودر آن را بنویسیم با ورودی پیام و سرعت تنظیم می کنیم. در تابع coding_freq دو ورودی پیام مورد نظر و سرعت ارسال پیام قابل فراخوانی است؛ که در بخش نخست آن باینری استرینگ را ساخته و آنرا کد می کنیم . عملا پیام باینری به تعداد بیت ریت جدا سازی شده و به عدد آن بخش بخش جدا میکند مثلا با bit rate=3 سه تا سه تا جداسازی بیتی انجام می دهد و تمام جایگشت های باینری ممکن ذخیره سازی میگردد و به آن فرکانس هایی نسبت داده می گردد و در ادامه پیام کدگذاری شده را پلات می توانیم بکنیم.

```
function CodedSignal = coding_freq(ClearText, rate)

Mapset = CreateMapset();

BinStr = '';
for charIndex = 1:strlength(ClearText)
    charMappingIndex = find(strcmp(Mapset(1,:), ClearText(charIndex)));
    BinStr = strjoin([BinStr, Mapset(2, charMappingIndex)], '');
end

CodedSignal = zeros(2, strlength(BinStr) * 100 / rate);

section = 50 / (2^rate + 1);

i = 0;

while i + rate <= strlength(BinStr)
    temp = BinStr((i + 1):(i + rate));
    factor = (bin2dec(temp) + 1) * section;

    time_range = (i / rate) : 0.01 : ((i / rate) + 1 - 0.01);
    signal = sin(2 * pi * factor * time_range);

    sample_range = ((i / rate) * 100 + 1):((i / rate) * 100 + 100);
    CodedSignal(1, sample_range) = time_range;
    CodedSignal(2, sample_range) = signal;

    i = i + rate;
end
end
```

خروجی برای پیام signal با سرعت ارسال ۱ و ۵ بدین صورت است :



Bit rate = 1



Bit rate =5

با توجه به اینکه بایستی بیت ریت ها باید بر طول حرف داده شده بخش پذیر باشند برای تست بیت ریت مثلا ۴ دیگر نمیتوانیم کلمه signal را ورودی بدهیم و این متناسب سازی بایستی انجام پذیرد.

اکنون برای پیاده سازی تابع decoding که دارای دو ورودی پیام کدگذاری شده و سرعت ارسال پیام است و خروجی آن مپ ست می باشد به فرمت فوق تبدیل فوریه هر پارت را محاسبه کرده و در ادامه آنرا به رشته باینری تبدیل نموده و متن انکود را پیدا می کنیم و در نهایت به کمک دستور strjoin کارکتر های انتخاب شده را به یکدیگر می چسبانیم.

```
function DecodedText = decoding_freq(CodedSignal, rate)
    FourArr = [];
    for startIndex = 0:100:(length(CodedSignal)-1)
        segment = CodedSignal(2, startIndex + 1:(startIndex + 100));
        fourier = fftshift(fft(segment));

        [~, idx] = max(abs(fourier));
        idx = idx - 51;

        FourArr = [FourArr, -1 * idx];
    end

    section = 50 / (2^rate + 1);
    BinArr = dec2bin(uint8(FourArr / section - 1));

    BinArr = BinArr';

    BinStr = '';

    for charIndex = 1:numel(BinArr)
        BinStr = strcat(BinStr, BinArr(charIndex));
    end

    Mapset = CreateMapset();

    DecodedText = '';

    for i = 1:5:strlength(BinStr)
        binSubstring = BinStr(i:i+4);
        index = find(strcmp(Mapset(2,:), binSubstring));
        DecodedText = strjoin([DecodedText, Mapset(1, index)], '');
    end
end
```

۵-۲:

```
clc
clear all
close all
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
text = 'signal';
rate = 1;
noise_variance =1.25;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

Coded = coding_freq(text, rate);
plot (Coded(1,:) ,Coded(2,:));
Noise = noise_variance * randn(1, length(Coded(2,:)));
Coded(2,:) = Coded(2,:) + Noise;
output = decoding_freq(Coded, rate);
output
```

در این بخش می خواهیم اثر نویز را در گیرنده اضافه کنیم و نویز گوسی با واریانس ۰.۰۰۰۱ و میانگین صفر را به آن اعمال می کنیم

مثلا با نویز با واریانس ۰.۰۰۰۱و سرعت ۵ مجدد می توانیم آنرا دیکود کنیم :

```
output =

    'signal'
```

همچنین در سرعت ۱ و این واریانس :

```
output =

    'signal'
```

و مجدد مشاهده می شود که توانست متن خواسته شده را دیکود نماید. از آنجایی که مقدار نویز کم است نتایج تغییری نمیکند و خروجی همچنان همان کلمه است.

۲-۶:

نمونه های بررسی شده با bit rate = 1 :

```
output =
    'signal'
```

واریانس ۰.۲۵

```
output =
    'signal'
```

واریانس ۱.۱

```
output =
    'sihnal'
```

واریاس ۱.۱۵

```
output =
    'ifcceacraqab'
```

واریانس ۲

نمونه های بررسی شده با bit rate =5 :

```
output =
    'signal'
```

واریانس ۰.۲۵

```
output =
    'rignal'
```

واریانس ۱.۱

```
output =
    'jmunal'
```

واریانس ۱.۷

همانطور که مشاهده می گردد پیام با بیت ریت ۱ مقاو م تر بود چرا که فاصله بین فرکانس های بیشتر و خطا کمتر است و با بحث مطرح شده در مقدمه همگام است.

۲-۷:

در سرعت ۵ بیشینه واریانس ۱.۰۵ و در سرعت ۱ برابر ۱.۲۵ می باشد .

۲-۹:

هنگامی که پهنای باند را تغییر نمی دهیم ، افزایش فرکانس نمونه برداری تغییری در فاصله بین فرکانس ها ایجاد نمیشود به همین علت تاثیری در مقاوم بودن نسبت به نویز ندارد.

جمع بندی : با افزایش سیگنال ها ، احتمال خطا کاهش یافته و بیت ریت نسبت به نویز مقاومتش بیشتر می گردد و با افزایش پهنای باند این فاصله به اندازه خوبی افزایش می یابد و مقاومت بیت ریت ها بیشتر می گردد.