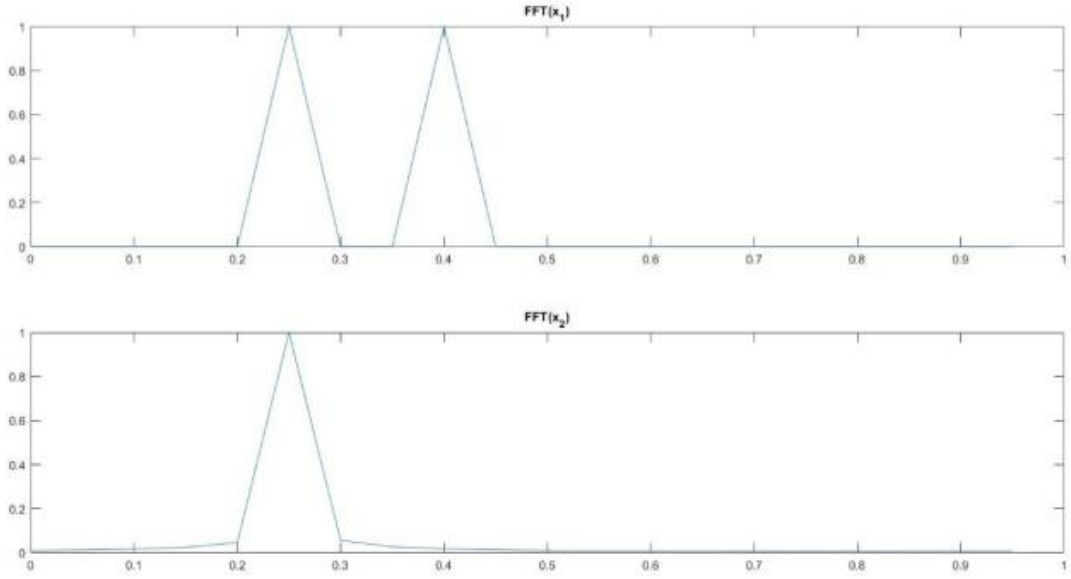
تمرین کامپیوتری 5 سیگنال ها و سیستم ها

مرضیه موسوی 810101526

مهراد لیویان 810101501

:0-1

در حالتی که فرکانسهای تابع را برابر با 5و 8 در نظر بگیریم، دو قله بر روی 5و 8به وضوح قابل مشاهده هستند. اما اگر این مقادیر را 5و 5.1در
نظر بگیریم، چون اختلاف فرکانس تک تن ان کمتر از 1هرتز است این تفاوت قابل مشاهده نیست
و فقط یک قله نشان داده شده است که در تصویر زیر اورده شده است:

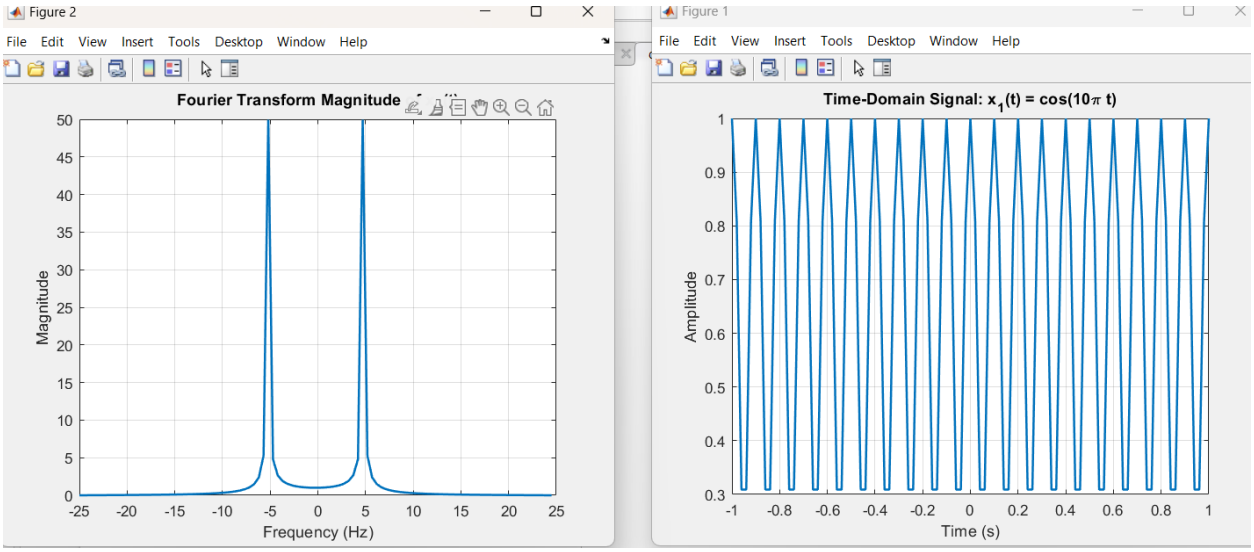


:1-1

اسکریپت زیر برای این تابع نوشته شده است:

```
fs = 50;
t_start = -1;
t_end = 1;
f = 5;


ts = 1 / fs;
t = t_start:ts:t_end;


x1 = cos(10 * pi * t);

figure;
plot(t,abs(x1) , 'LineWidth', 1.5);
xlabel('Time (s)');
ylabel('Amplitude');
title('Time-Domain Signal: x_1(t) = cos(10\pi t)');
grid on;


N = length(x1);
X1 = fft(x1);
f_axis = (-N/2:N/2-1) * (fs/N);
X1_magnitude = (abs(X1));


figure;
plot(f_axis, X1_magnitude, 'LineWidth', 1.5);
xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('Fourier Transform Magnitude of x_1(t)');
grid on;
```

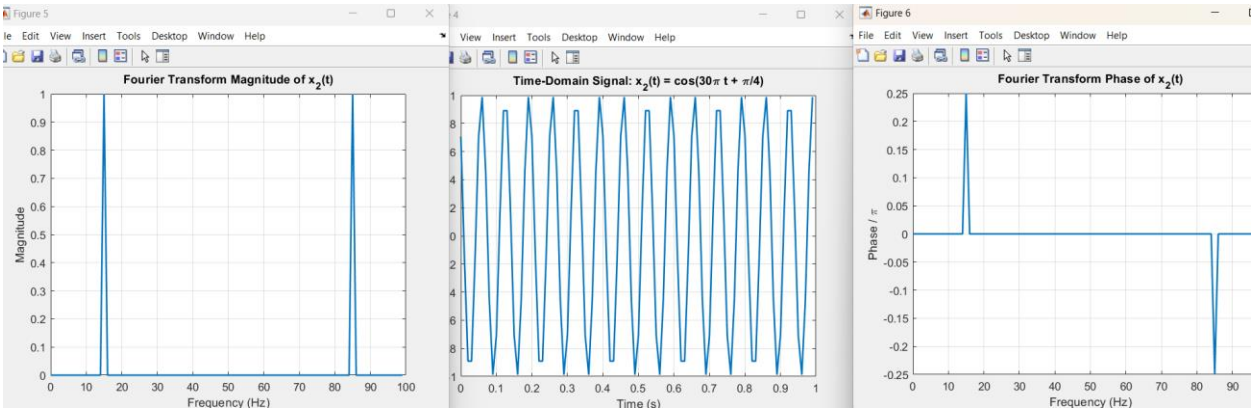
شکل های به دست آمده به صورت زیر است:



همانطور که در شکل مشخص است تبدیل فوریه به صورت مجموع دو ضربه به دست می آید که مطابق دانسته های ماست.

:1-2

```
fs = 100;
t_start = 0;
t_end = 1;
f = 15;
phi = pi / 4;

ts = 1 / fs;
t = t_start:ts:t_end-ts;


x2 = cos(30 * pi * t + phi);

figure;
plot(t, (x2), 'LineWidth', 1.5);
xlabel('Time (s)');
ylabel('Amplitude');
title('Time-Domain Signal: x_2(t) = cos(30\pi t + \pi/4)');
grid on;

N = length(x2);
X2 = fft(x2);
f_axis = 0:(fs/N):((N-1)*fs/N);
X2_shifted =X2;
X2_magnitude = abs(X2_shifted) / max(abs(X2_shifted));
X2_phase = angle(X2_shifted);

figure;
plot(f_axis, X2_magnitude, 'LineWidth', 1.5);
xlabel('Frequency (Hz)');
ylabel('Magnitude');
title('Fourier Transform Magnitude of x_2(t)');
grid on;

tol = 1e-6;
X2_phase(X2_magnitude < tol) = 0;
theta = X2_phase / pi;


figure;
plot(f_axis, theta, 'LineWidth', 1.5);
xlabel('Frequency (Hz)');
ylabel('Phase / \pi');
title('Fourier Transform Phase of x_2(t)');
grid on;
```



فرکانس آن 85 و 15 است که در صورت تبدیل فوریه گرفتن از تابع x به صورت مجموع دو ضربه در 85 و 15 به دست می آید که با دانسته ها مطابقت دارد.

بخش دوم:

تمرین 2-1 و 2-2:

دو اسکریپت زیر برای درست کردن mapset و کد کردن ورودی نوشته شده است:

```
function myMapset=MapSet()
    myMapset=cell(2,32);
    alphabet = 'abcdefghijklmnopqrstuvwxyz .,!";';
    for i = 1:32
        myMapset{1,i} = alphabet(i);
        myMapset{2,i} = dec2bin(i-1, 5);
    end

end
```

تمرین 2-3:

```
function coded_message = coding_freq(message, rate)

    Mapset = MapSet;
    binaryMessage = [];
    fs = 100;
    numChunks = ceil((strlength(message) * 5) / rate);


    for charIndex = 1:strlength(message)
        for mapIndex = 1:32
            if strcmp(extract(message, charIndex), Mapset(1, mapIndex))
                binaryMessage = [binaryMessage, Mapset(2, mapIndex)];
            end
        end
    end
    binaryMessage = cell2mat(binaryMessage);


    binaryChunks = mat2cell(binaryMessage, 1, repmat(rate, 1, floor(length(binaryMessage)/rate)));
    if mod(length(binaryMessage), rate) > 0
        binaryChunks{end+1} = binaryMessage(end-mod(length(binaryMessage), rate)+1:end);
    end
```

ابتدا mapset راinitializeمی کنیم.تعداد تکه هایی که می خواهیم کد کنیم را به دست می آوریم.

یک حلقه روی پیام می زنیم.اگر کاراکترش با کاراکتری که در مپ ست است مچ بود کد ان را به binary massage اضافه می کنیم.

بخش باینری را به تکه هایی تقسیم می کنیم.در صورتی که تعداد بیت های کل به ریت بخش پذیر نبود چند کاراکتر در انتها اضافه می کنیم.

```
freqStep = floor(50 / (2^rate));
freqArray = ((0:(2^rate-1)) * freqStep) + floor(freqStep / 2);


binaryRepresentations = arrayfun(@(x) dec2bin(x, rate), 0:(2^rate-1), 'UniformOutput', false);


frequencies = zeros(1, numChunks);
for chunkIndex = 1:numChunks
    binaryChunk = binaryChunks{chunkIndex};
    matchingIndex = find(strcmp(binaryChunk, binaryRepresentations));
    if ~isempty(matchingIndex)
        frequencies(chunkIndex) = freqArray(matchingIndex);
    end
end
```

آرایه ای متشکل از فرکانس هایمان درست می کنیم.

برای هر فرکانس یک نمایش باینری درست می کنیم و سپس به جای هر فرکانسی در binary chunk مان آن را می گذاریم.

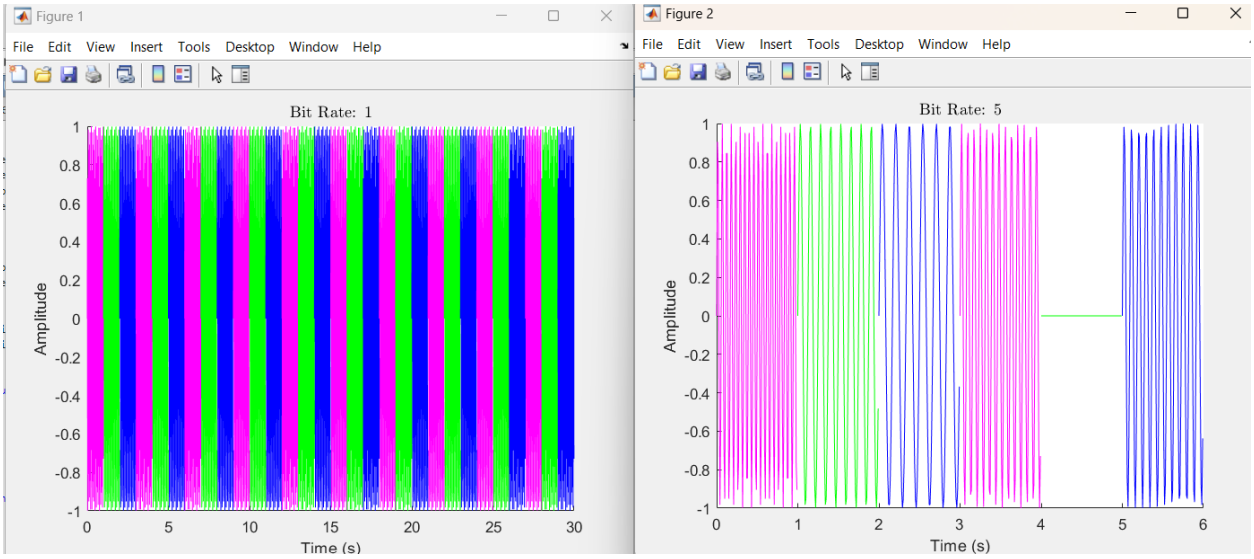
```
timeStep = 1 / fs;
timeVector = 0:timeStep:1-timeStep;
coded_message = arrayfun(@(freq) sin(2 * pi * freq * timeVector), frequencies, 'UniformOutput', false);
coded_message = cell2mat(coded_message');
```

به ازای هر فرکانس یک موج سینوسی تولید می کنیم.

```
    colors = {'m', 'g', 'b'};
    figure;
    hold on;
    for chunkIndex = 1:numChunks
        timeOffset = (chunkIndex - 1);
        plot(timeVector + timeOffset, coded_message(chunkIndex, :), 'Color', colors{mod(chunkIndex - 1, length(colors)) + 1});
    end
    hold off;


    title(['Bit Rate: ', num2str(rate)], 'Interpreter', 'latex');
    xlabel('Time (s)');
    ylabel('Amplitude');
end
```

هر بخش از موج سینوسی با استفاده از آفستی مپ شده و کشیده می شود.

نتیجه ی تست:



اسکریپت زیر برای تست نوشته شده است:

```
message1 = coding_freq('signal', 1);  % Encode the message with rate 1
message2 = coding_freq('signal', 5);  % Encode the message with rate 5
```

تمرین 2-4:

اسکریپت زیر مربوط به decode فانکشن است:

```
function decodedMessage = decoding_freq(codedmessage, rate)

    binary_num = 2^rate;
    binary_num = arrayfun(@(x) dec2bin(x, rate), 0:binary_num-1, 'UniformOutput', false);
    Mapset = MapSet();
    messageSize = size(codedmessage);
    fs = 100;

    numSegments = 2^rate;
    segmentwidth = fix((fs/2) / numSegments + 1);
    Increment = fix(segmentwidth / 2);
    segment1 = Increment + (0:numSegments-1) * segmentwidth;
```

ابتدا تمام مقادیری را که براساس بیت ریت می توانیم داشته باشیم را بدست می آوریم.

سپس width هر بخش فرکانس را بدست می آوریم.فرکانس مرکز هر بخش را بدست می آوریم تا برای بدست آوردن مقادیر باینری استفاده کنیم.

```
frequencyRange = -(fs/2):1:(fs/2) - 1;
dominantFrequencies = [];


frequencySpectrum = abs(fftshift(fft(codedmessage, [], 2), 2));
[~, maxIndices] = max(frequencySpectrum, [], 2);
dominantFrequencies = abs(frequencyRange(maxIndices));
```

حال تبدیل فوریه مسیج را بدست می آوریم.ایندکس ماکس تبدیل فوریه را بدست آورده و مقدار آن را در dominant frequencies ذخیره می کنیم.

```
segmentthreshold = zeros(1, 2^rate);
for i= 1 : 2^rate - 1
    segmentthreshold(1, i + 1) = (segment1(1, i + 1) - segment1(1, i)) / 2 + segment1(1, i);
end
binarySequence = [];

segmentThreshold = (segment1(1, 2:end) + segment1(1, 1:end-1)) / 2;
binarySequence = cell(1, messageSize(1));
```

آستانه های بین های فرکانس را بدست می آوریم که بفهمیم هر فرکانس dominant در چه محدوده ای می باشد.

```
for i = 1:messageSize(1)
% finding the segment index for the current dominant frequency
segmentIndex = find(dominantFrequencies(1, i) < segmentThreshold, 1);
if isempty(segmentIndex)
    segmentIndex = 2^rate;
end
% appending the corresponding binary value to the binary sequence
binarySequence{i} = binary_num{segmentIndex};
end
% concatenating the binary sequence into a single array
   binarySequence = [binarySequence{:}];
```

با استفاده از آستانه های بدست آمده بخش مربوط به هر فرکانس را پیدا می کنیم.سپس به مقادیر باینری مپ می کنیم و مقادیر باینری را concat می کنیم.

```
bitsforchar = 5;
binarySequenceLength = length(binarySequence);
count = 1;

for k = 1 : bitsforchar : binarySequenceLength
    startIdx = k;
    endIdx = min(k + bitsforchar - 1, binarySequenceLength);

    groupedBinary{count} = binarySequence(startIdx : endIdx);
    count = count + 1;
end
```

حال برایdecoding سیکوینس بدست آمده را 5 تا 5 تا جدا می کنیم.

کاراکتر مربوط به هر بخش باینری را در مپست می یابیم و مسیج را دیکود می کنیم.

```
    binarySequence = groupedBinary;
    decodedMessage = [];
    mapLength = size(Mapset);

    decodedMessageCell = {};
for m = 1 : messageSize(1) * rate / bitsforchar
    for n = 1 : mapLength(1, 2)

        if all(binarySequence{1, m} == Mapset{2, n})
            decodedMessageCell{end+1} = Mapset{1, n};
        end
    end
end
decodedMessage = strjoin(decodedMessageCell);
```

اسکریپت زیر برای تست نوشته شده است:

```
message1 = coding_freq('signal', 1);
message2 = coding_freq('signal', 5);
decoded2 = decoding_freq(message2, 5);
decoded1 = decoding_freq(message1, 1);
```

نتیجه ی تست به صورت زیر است:

```
Decoded message for rate 1: s i g n a l
Decoded message for rate 5: j e d g z f
```

:2-5

تابع اضافه کردن noise به صورت زیر است:

```
function noisyMessage = withNoise(message, noiseLevel)


    noise = noiseLevel * randn(size(message));
    noisyMessage = message + noise;
end
```

برای جنریت کردن نویز ابتدا سایز مسیج کد شده را بدست می آوریم.با استفاده از تابع randn اعداد رندوم را جنریت می کنیم و سیگما را در آن ضرب می کنیم.

```
message1 = coding_freq('signal', 1);
message2 = coding_freq('signal', 5);
message1 = withNoise(message1, 0.0001);
message2 = withNoise(message2, 0.0001);
decoded2 = decoding_freq(message2, 5);
decoded1 = decoding_freq(message1, 1);
```

اسکریپت بالا برای تست نوشته شده است.

```
Decoded message for rate 1: s i g n a l
Decoded message for rate 5: j e d g i f
```

همانطور که در خروجی مشخص است بیت ریت یک تفاوتی ندارد ولی بیت ریت 5 نتایج مرحله ی قبل را نمی دهد.

2-6

```
Decoded message for rate 1: s a g n a l
Decoded message for rate 5: j e d g b f
    0.9000

Decoded message for rate 1: s i h n q l
Decoded message for rate 5: j e d g y f
     1

Decoded message for rate 1: s a h e e k
Decoded message for rate 5: o e d g d f
    1.1000
```

```
    0.1000

Decoded message for rate 1: s i g n a l
Decoded message for rate 5: j e d g a f
    0.2000

Decoded message for rate 1: s i g n a l
Decoded message for rate 5: j e d g e f
    0.3000

Decoded message for rate 1: s i g n a l
Decoded message for rate 5: j e d g y f
    0.4000

Decoded message for rate 1: s i g n a l
Decoded message for rate 5: j e d g f f
    0.5000

Decoded message for rate 1: s i g n a l
Decoded message for rate 5: j e d g g f
    0.6000

Decoded message for rate 1: s i g n a l
Decoded message for rate 5: j e d g g f
    0.7000

Decoded message for rate 1: s i g n a l
Decoded message for rate 5: j e d g x f
    0.8000
```

همانطور که از نتایج مشخص است بیت ریت 1 به نویز مقاوم تر بود.

2-7:

بیت ریت 5 با کوچکترین نویزی نتایج اشتباه می داد درحالی که بیت ریت یک تا حدود نویز 0.9 مقاوم است.

:2-8

اگر پهنای باند را بیشتر کنیم فاصله ی فرکانس هایی که وجود دارد را می توانیم بیشتر در نظر بگیریم و نسبت به نویز مقاوم تر است.

:9-2

وقتی نرخ نمونه‌برداری افزایش پیدا می‌کند، تعداد نمونه‌های سیگنال در هر بازه زمانی بیشتر می‌شود .این امر باعث می‌شود سیگنال بازسازی شده دقیق‌تر باشد و بتوان اطلاعات اصلی را با دقت بیشتری تشخیص داد .حتی اگر نویز حضور داشته باشد، شانس استخراج اطلاعات درست بیشتر می‌شود.