Computer Assignment 6

Seyed Mahdi HajiSeyedHossein

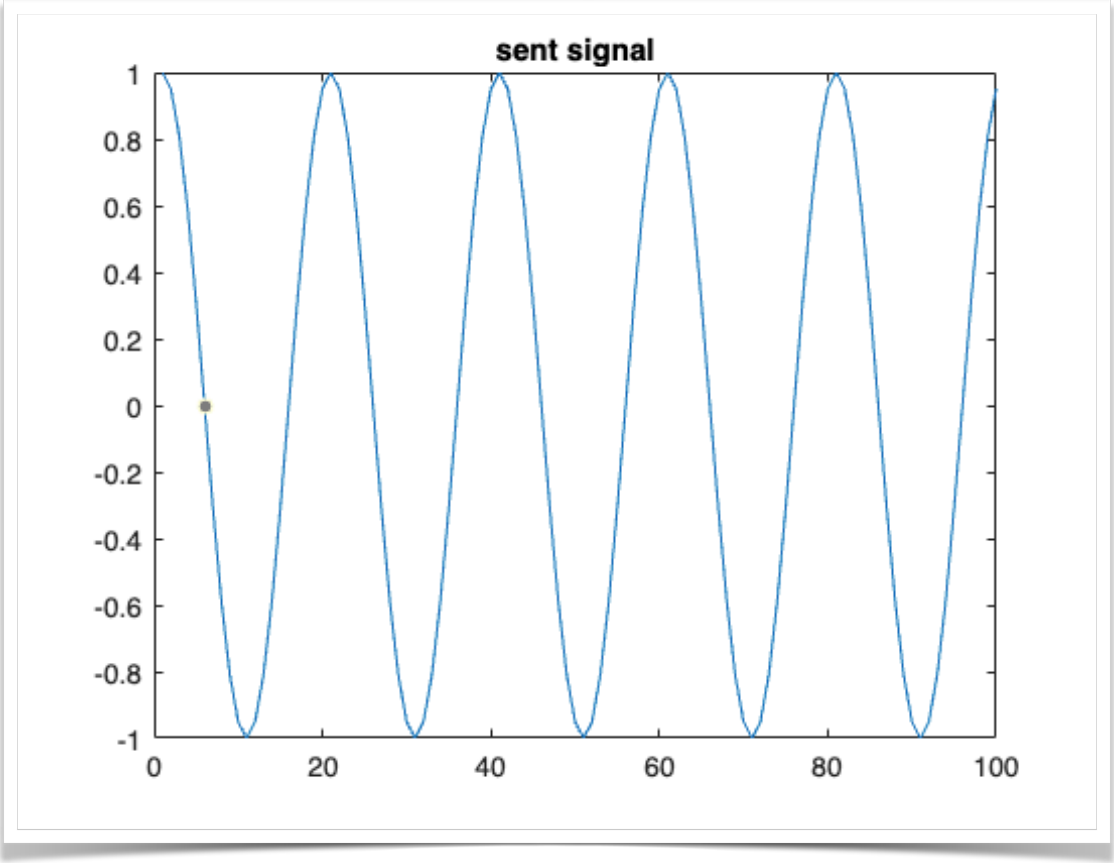
Student Number : 810100118

Dey - 1402

PART 1 :

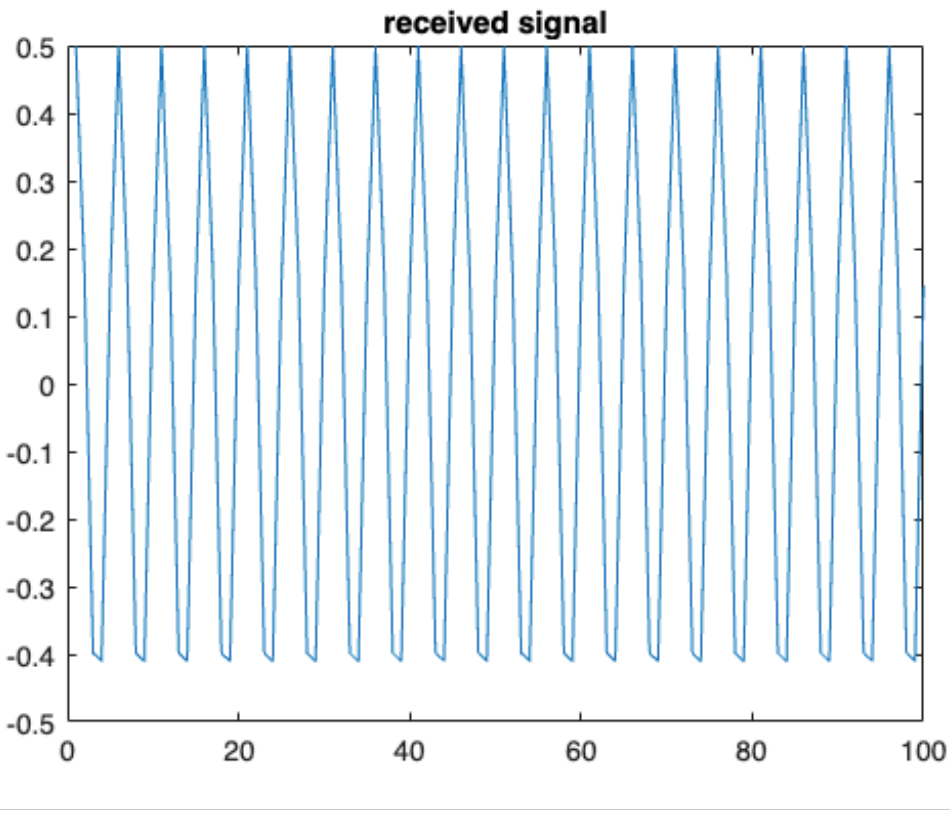
Question 1 :

```
1   clear;clc;
2   tstart=0;
3   tend=1;
4   fs=100;
5   ts=1/fs;
6   t=tstart:ts:tend-ts;
7
8
9   fc=5;
10  sent=cos(2*pi*fc*t);
11  plot(sent)
12  title('sent signal')
13
14
```



Question 2 :

```
tstart=0;
tend=1;
fs=100;
ts=1/fs;
t=tstart:ts:tend-ts;


fc=5;
V=180*10/36;
R=250*1000;
beta=0.3;
alpha=0.5;
fd=beta*V;
c=3*1e9;
ro=2/c;
td=ro*R;
received=alpha*cos(2*pi*(fc+fd)*(t-td));
plot(received)
title('received signal')

```



Question 3 :

برای یافتن سرعت و فاصله جسم، می توانیم فرکانس و فاز سیگنال دریافتی را بیابیم. مقدار فرکانس طبق فرمول برابر مجموع fc و fd است و چون مقدار fc معلوم است، fd و در نتیجه سرعت جسم به دست می آید.

مقدار فاز سیگنال در فرکانس پیک نیز td است ، و در نتیجه فاصله جسم را بدست آورده ایم .

برای یافتن فرکانس دستور های fft , ffshift را روی سیگنال مورد نظر پیاده میکنیم . با استفاده از دستور max ، find نقطه پیک دوم فرکانسی را میابیم .

و با استفاده از موقعیت پیک ، فرکانس سیگنال مورد نظر را پیدا میکنیم ،

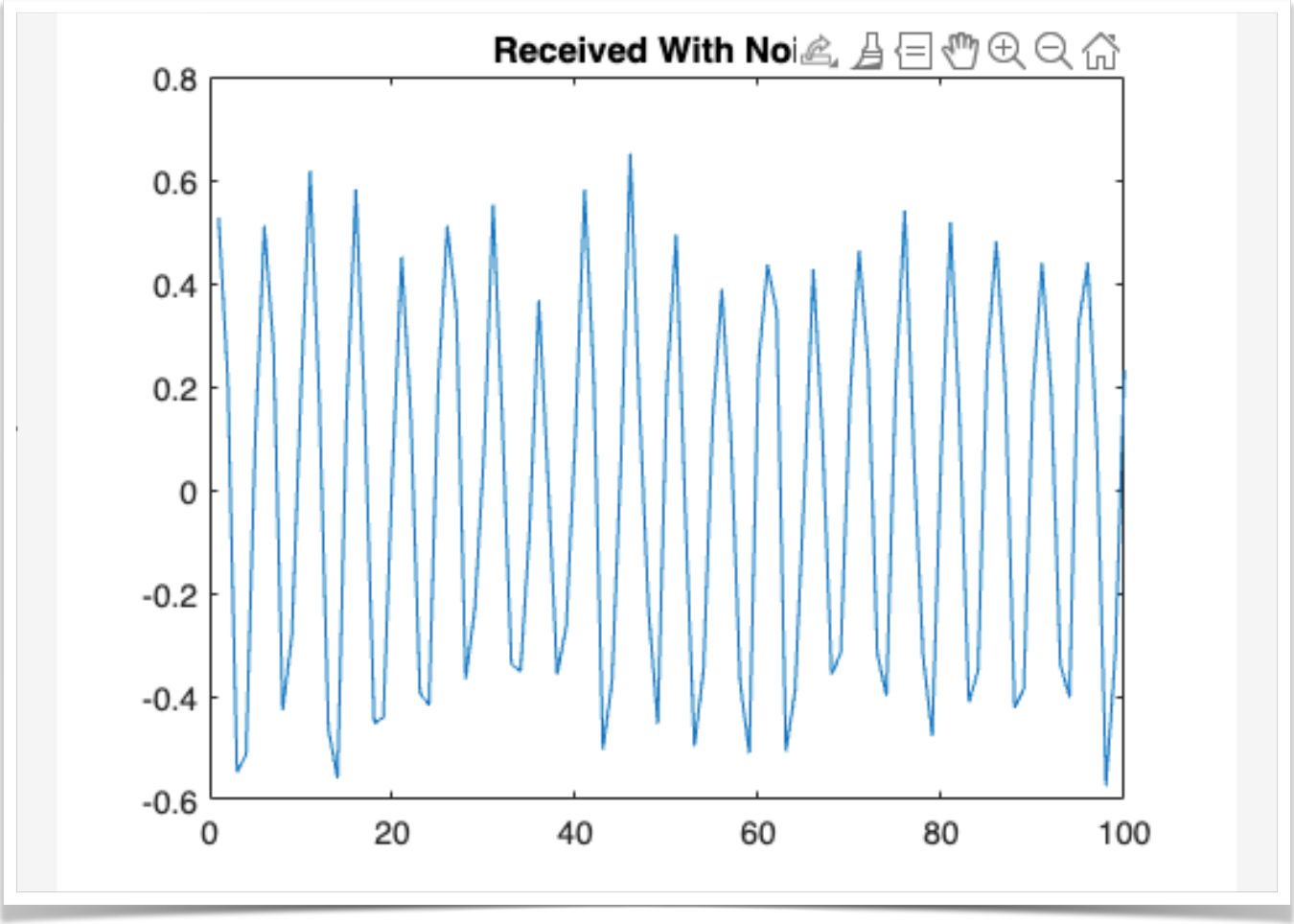
دستور angle را نیز روی سیگنال پیاده سازی میکنم ، و مقدار را در آن قسمت پیک پیدا میکنیم .

این مقدار همون مقدار فاز است .

```
>> Q3

Vfound =

   180


Rfound =

  250.0000

>>
```

```
1
2   tstart=0;
3   tend=1;
4   fs=100;
5   ts=1/fs;
6   t=tstart:ts:tend-ts;
7
8
9   fc=5;
10  V=180*10/36;
11  R=250*1000;
12  beta=0.3;
13  alpha=0.5;
14  fd=beta*V;
15  c=3*1e9;
16  ro=2/c;
17  td=ro*R;
18  received=alpha*cos(2*pi*(fc+fd)*(t-td));
19
20
21
22
23  N=length(received);
24  m=fftshift(fft(received));
25  theta = angle(m);
26  m=abs(m);
27  [,col]=find(m==max(m));
28  pha=abs(theta(col(2)));
29  fnew=(col(2)-N/2-1)*fs/N;
30  fdfound=fnew-fc;
31  Vfound=fdfound/beta*36/10
32  Rfound=pha/(2*pi*(fdfound+fc)*ro)*0.001
33
34
35
```

Question 4 :



خروجی فاصله و سرعت جسم با نویز به صورت زیر است:

```
>> Q4

Vfound_withnoise =

   180

Rfound_withnoise =

  261.7777

>>
```

```
1
2   tstart=0;
3   tend=1;
4   fs=100;
5   ts=1/fs;
6   t=tstart:ts:tend-ts;
7
8
9   fc=5;
10  V=180*10/36;
11  R=250*1000;
12  beta=0.3;
13  alpha=0.5;
14  fd=beta*V;
15  c=3*1e9;
16  ro=2/c;
17  td=ro*R;
18  received=alpha*cos(2*pi*(fc+fd)*(t-td));
19
20
21  received=received+0.01*randn(size(received));
22  plot(received);
23  title("Received With Noise")
24  N=length(received);
25  m=fftshift(fft(received));
26  theta = angle(m);
27  m=abs(m);
28  [,col]=find(m==max(m));
29  pha=abs(theta(col(2)));
30  fnew=(col(2)-N/2-1)*fs/N;
31  fdfound=fnew-fc;
32  Vfound_withnoise=fdfound/beta*36/10
33  Rfound_withnoise=pha/(2*pi*(fdfound+fc)*ro)*0.001
34
```

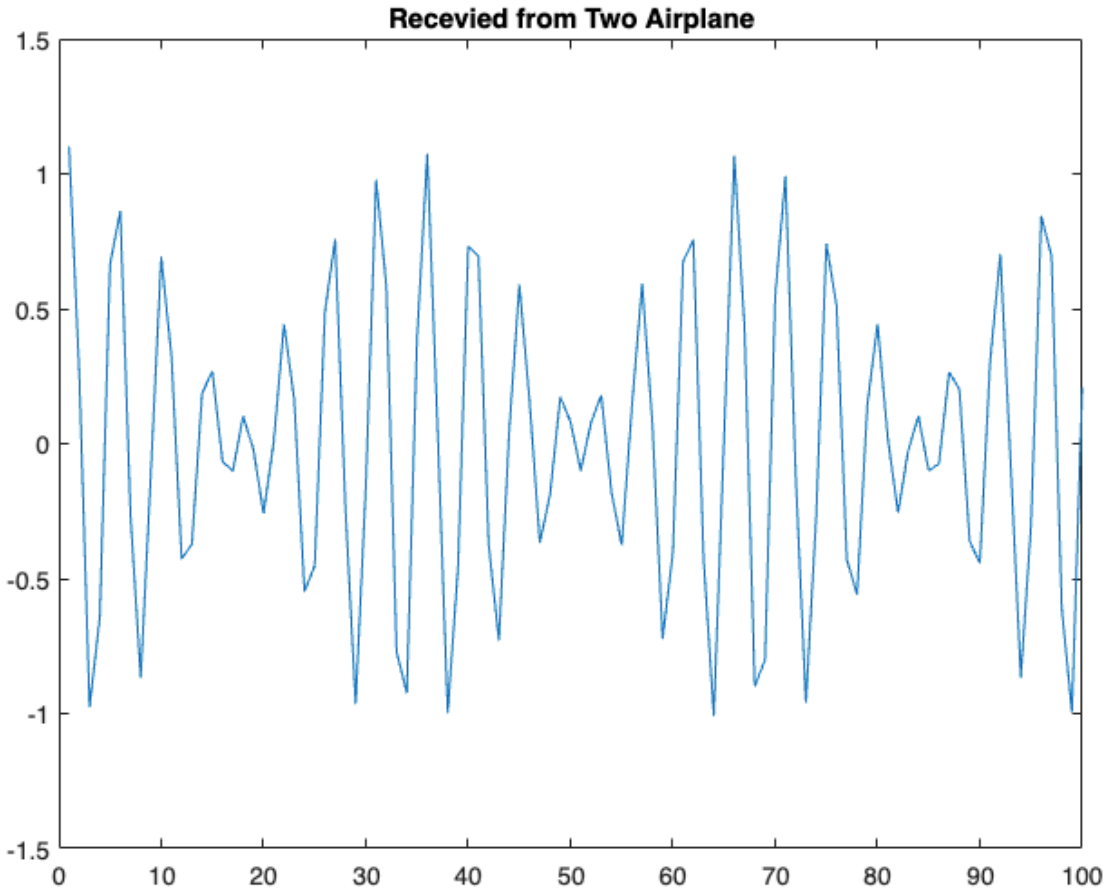
سرعت جسم حساسیت زیادی نسبت به نویز ندارد و با اضافه کردن تا حدود یک برابر نویز به سیگنال درست محاسبه می شود. (ضریب پشت نویز 1 باشد)

اما فاصله جسم با کوچکترین مقدار نویز با خطا تشخیص داده می شود. و مقدار آن تقریبی خواهد بود .

Question 5 :

سیگنال دریافتی مجموع سیگنال دریافتی از هریک از دو جسم می باشد.

```
tstart=0;
tend=1;
fs=100;
ts=1/fs;
t=tstart:ts:tend-ts;


fc=5;
V=180*10/36;
R=250*1000;
beta=0.3;
alpha=0.5;


R1=250*1000;R2=200*1000;
V1=180*10/36;V2=216*10/36;
alpha1=0.5;alpha2=0.6;
fd1=round(beta*V1);
fd2=round(beta*V2);
td1=ro*R1;
td2=ro*R2;
received1=alpha1*cos(2*pi*(fc+fd1)*(t-td1));
received2=alpha2*cos(2*pi*(fc+fd2)*(t-td2));
final=received1+received2;
figure()
plot(final)
title('Recevied from Two Airplane')
```


Recevied from Two Airplane
1.5
1
0.5
0
-0.5
-1
-1.5
0
10
20
30
40
50
60
70
80
90
100

Question 6 :

باید دقیقا همانند سوال ۳ عمل کنیم . فقط باید به ازای هر هواپیما هر کدام یک fft , shift استفاده کنیم .

محل هردو پیک را با ددستور max بدست میاَوردم . که در واقع از آن میتوان فرکانس را در اورد.

به ازای هر فرکانس نیز فاز را مثل سوال سه حساب میکنم .

برای اینکه fs برای تعییر فرکانس دقت را کم میکند ، در قسمت قبلی فرکانس را با دستور round رند میکنیم تا مسافت محاسبه شده نیز دور نباشد .

```
>> Q6

Vfound1 =

   216

Vfound2 =

   180

Rfound1 =

  200.0000

Rfound2 =

  250.0000

>>
```

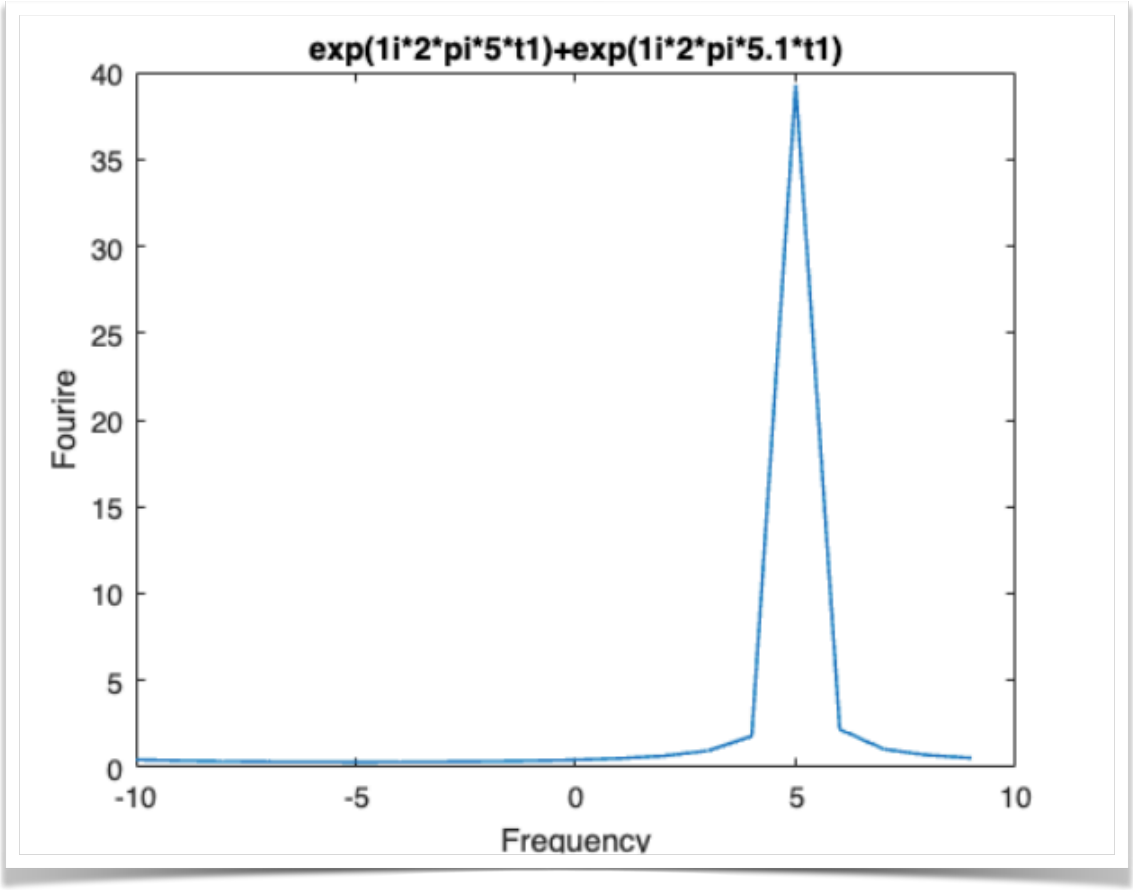
```
N=length(final);
m2=fftshift(fft(final));
theta2 = angle(m2);
m2=abs(m2);
[mx,col]=maxk(m2,4);
fnew1=(col(2)-N/2-1)*fs/N;
pha1=abs(theta2(col(2)));
fnew2=(col(4)-N/2-1)*fs/N;
pha2=abs(theta2(col(4)));
fdfound1=fnew1-fc;
fdfound2=fnew2-fc;

Vfound1=fdfound1/beta*36/10
Vfound2=fdfound2/beta*36/10
Rfound1=pha1/(2*pi*(fdfound1+fc)*ro)*0.001
Rfound2=pha2/(2*pi*(fdfound2+fc)*ro)*0.001
```

Question 7 :

در این حالت امکان تفکیک دو سنگنال را در حوزه فوریه نداریم . چرا ؟ برمیگردیم به پروژه قبلی :

چون اختلاف فرکانس دو سیگنال تک تن کمتر از 1Hz است در صورتی که اختلاف دو سرعت بر حسب کیلومتر بر ساعت برابر 12 باشد اختلاف فرکانس ها به یک هرتز میرسد ، و پارامتر ها درست استخراج میشوند .

عکسی از پروژه قبل :



Question 8 :

بله سرعت به درستی استخراج میشوند چون قابلیت تفکیک فرکانس های آنها را داریم .

Question 9 :

بعد از اعمال دستور fftshift , fft ، سیگنال حوزه فوریه به تعداد هواپیما ها پیک دارد ،

برای یافتن هر پیک می توان در هر نقطه از نیمه دوم نمودار(قسمت مربوط به فرکانس های مثبت) مقدار نمودار را با نقطه قبل ان مقایسه کرد. اگر نمودار پیک زده باشد این مقدار بزرگ است.

پس با تعیین یک آستانه و مقایسه اختلاف دو نقطه با آن، می توان هر تعداد پیک را پیدا کرد و فرکانس و فاز آنها را یافت.

PART 2 :

Question1 :

در ابتدا همان ثابت هایی که در صورت پروژه آمده است را مینویسم :

```
tstart=0;

fs=8000;
ts=1/fs;
tend=0.5;
tau=0.025;

t=tstart:ts:tend-ts;
t2=tstart:ts:tau-ts;


silence=zeros(size(t2));

```

سپس به هر کاراکتر موجود در پیانومان را به همراه فرکانس هایشان پیاده سازی میکنیم :

```
characters=["B" , "A#" , "A" , "G#" , "G" , "F#" , "F" , "E" , "D#" , "D" , "C#" , "C"];
characters_freq=[987.77 , 932.33 , 880 ,830.61 ,783.99,739.99,698.46 , 659.25 , 622.25 , 587.33 , 554.37 , 523.25]
```

حالا در قسمت بعد آن موسیقی داده شده را به نوت های موجود تبدیل میکنم و آن را در یک سلول دو بعدی که بعد اول آن حروف نوت های موسیقی و بُعد دوم آن را به duration هر کاراکتر اختصاص میدهیم .

```
given_song={'D','D','G','F#','D','D','E','E','D','F#','D','E','D','E','F#','E','D','E','E','D','F#','D','E','D','E','D','F#','E','
;0.5,0.5,1,1,1,0.5,0.5,0.5,0.5,0.5,1,1,1,1,0.5,0.5,0.5,0.5,0.5,0.5,1,1,0.5,0.5,1,1,1,0.5,0.5,1,1,0.5,0.5,1,0.5,0
```

حالا باید یک حلقه لوپ تشکیل دهیم و هر کاراکتر نوت موسیقی را در کاراکتر پیدا کنیم و سپس فرکانس آن را دربیاوریم ، حالا باید به مقدار duration داده شده برای آن کاراکتر ، یک سیگنال از نوع سینوسی به شکل sin(2*pi*f*t) پیاده سازی کنیم که f مقدار فرکانس یافته شده است . و به اندازه ان duration این سیگنال را در آهنگ قرار دهیم .

و حالا باید به اندازه زمان استراحت بین دو انگشت سیگنال را صفر کنیم .

```
23  song=[];
24  for i=1:length(given_song)
25
26      [,num]=find(characters==given_song{1,i});
27      y=sin(2*pi*characters_freq(num)*t);
28
29
30      dur=given_song{2,i}*size(t,2);
31      song=[song y(1:dur) silence];
32
33  end
34  %sound(song)
35  audiowrite('given_song.wav',song,fs)
36
```

Question2 :

حالا کد قسمت یک را به یک function تبدیل میکنیم و کافی است یک آهنگ را به صورت گفته شده به عنوان ورودی تابع به تابع بدهیم :

که ورودی مثال ما به صورت زیر است :

```
% Input must be like this :
% {'NOTEs' ; theirDuration}

simple_song_notes = {'C', 'D', 'E', 'C', 'E', 'E', 'E' ;0.5, 0.5, 1, 0.5, 0.5, 0.5, 1};


makeMusic(simple_song_notes);
```

```
function makeMusic(given_song)
    %MAKEMUSIC Summary of this function goes here
    %   Detailed explanation goes here
    tstart=0;

    fs=8000;
    ts=1/fs;
    tend=0.5;
    tau=0.025;

    t=tstart:ts:tend-ts;
    t2=tstart:ts:tau-ts;
    silence=zeros(size(t2));


    characters=["B" , "A#" , "A" , "G#" , "G" , "F#" , "F" , "E" , "D#" , "D" , "C#" , "C"];
    characters_freq=[987.77 , 932.33 , 880 ,830.61 ,783.99,739.99,698.46 , 659.25 , 622.25 , 587.33 , 554.37 , 523.25]

    song=[];
    for i=1:length(given_song)

        [,num]=find(characters==given_song{1,i});
        y=sin(2*pi*characters_freq(num)*t);


        dur=given_song{2,i}*size(t,2);
        song=[song y(1:dur) silence];

    end
    sound(song)
    audiowrite('mysong.wav',song,fs)

end
```

Question3 :

این سوال دقیقا مشابه پروژه قبلی است که میتوان با استفاده از shfit , fft دقیقا نوت های موسیقی یک صوت را تشخیص داد .

یک حلقه while دارد برنامه که در آن میاییم تا زمانی که amplitute سیگنال به صفر نرسیده است ، سیگنال را قاچ میکنیم ، حالا باید به به دنبال انهای آن نوت باشیم که کافی است یک آرایه که که طول آن بیشتر از یک است و شامل صفر است را پیدا کنیم (چون که ممکن است در سینگال سینیوسی صفر داشته باشیم اما هیچ گاه دو صفر درون یک نوت کنار هم نمی آیند و فقط حداکثر یک صفر داریم)
پس حالا توانسته ایم که موج یک سینوس را پیدا کنیم .

و چون دیگر از اینجا به بعد طول سکوت ها را میدانیم که .200 هستند دیگر از نوت اول به بعد تمام نوت ها را داریم ،

```
23
24      notes={};
25      en=length(amp);
26      n=1;
27      while en~=0
28          for i=2:en
29              if amp(i)==0 && amp(i+1)==0
30                  break
31              end
32          end
33          y=amp(1:i-1);
34          notes(n)={y};
35          n=n+1;
36          amp=amp(i+200:en);
37          en=length(amp);
38
39      end
```

حالا برای اینکه بفهمیم هر سیگنال چه فرکانسی دارد ، مشابه تمرین کامپیتوری قبل از , fftshift fft استفاده میکنیم ، و فرکانس هر یک را میابیم ،

چون که فرکانس کمی خطا دارد ، برای هریک یک threashhold تعریف میکنیم ، که بفهمیم احتمالا بین کدام دو فرکانس قرار دارد ،

```
    for i=1:length(notes)

        y=cell2mat(notes(i));
        N=length(y);
        m=abs(fftshift(fft(y)));
        [row,col]=find(m==max(m));
        frq=(row(2)-N/2-1)*fs/N;
        for n=1:length(characters_freq)
            trsh=2;
            if abs(characters_freq(n)-frq)<trsh
                output(1,i)=characters(n);
                output(2,i)=N/4000;
            end
        end
    end

end
```

حالا به عنوان نمونه همان موسیقی که در بخش دو داریم را به decodeMusix میدهیم :

```
4   music = "mysong.wav";
5   output = decodeMusic(music)
```

و خروجی کاملا مشابه خود نوت های داده شده است :

```
4
5       % Input must be like this :
6       % {'NOTEs' ; theirDuration}
7
8       simple_song_notes = {'C', 'D', 'E', 'C', 'E', 'E', 'E' ;0.5, 0.5, 1, 0.5, 0.5, 0.5, 1};
9
10
```

```
output =

  2x7 string array

    "C"      "D"      "E"      "C"      "E"      "E"      "E"
    "0.5"    "0.5"    "1"      "0.5"    "0.5"    "0.5"    "1"

>>
```