به نام خدا

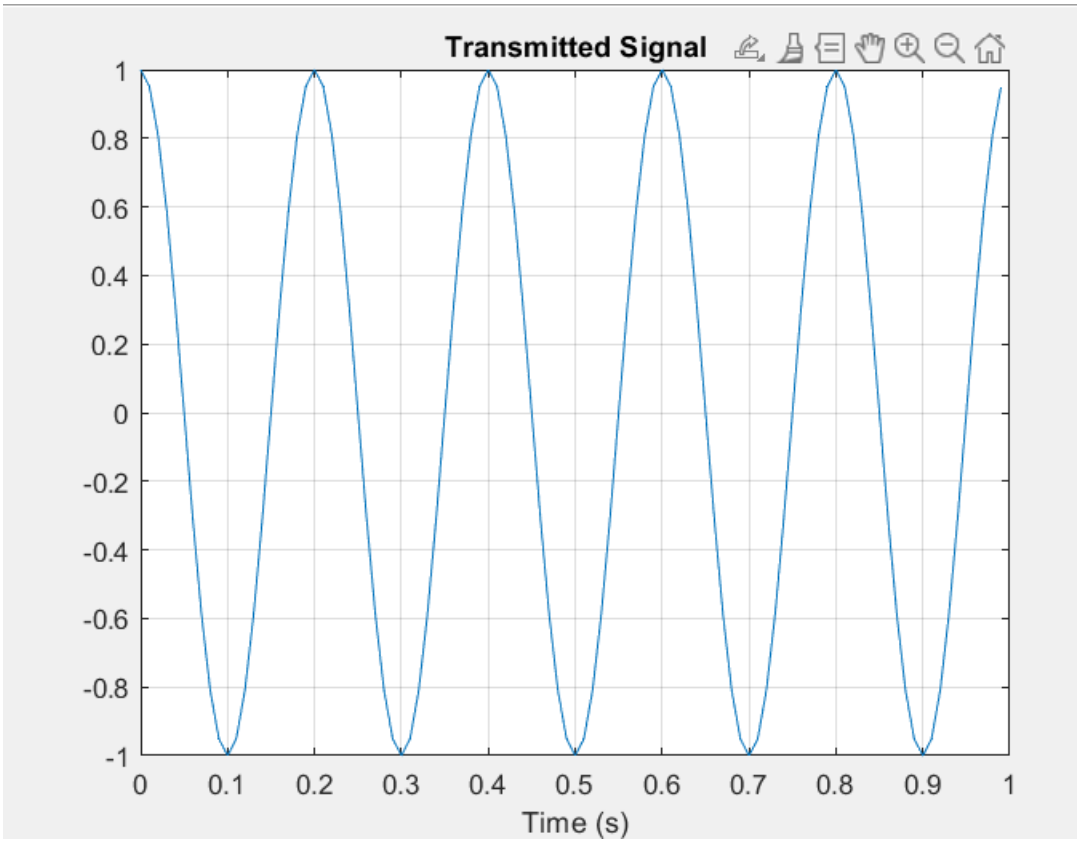
CA 6

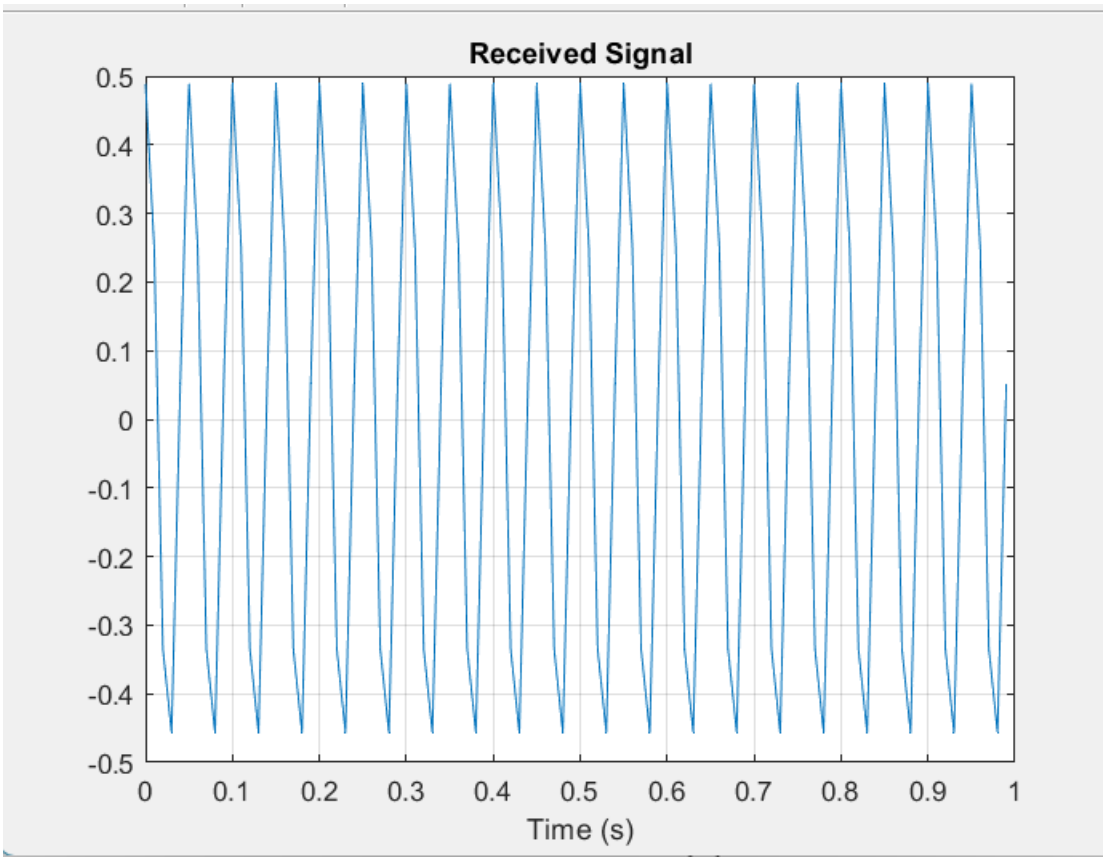
امیرعلی شهریاری (۸۱۰۱۰۰۱۷۳)

۱:

در این تمرین میخواهیم توسط رادار فاصله و سرعت جسم را بر اساس سیگنال ارسالی محاسبه کنیم .

۱-۱ و ۱-۲: در این بخش پارامتر های صورت سوال را ورودی می دهیم و سیگنال ارسالی را ترسیم کرده و با محاسبه متغیر های دیگر مشخص شده در صورت سوال سیگنال دریافتی را نیز محاسبه می کنیم و سرعت و مسافت گزارش می شوند.

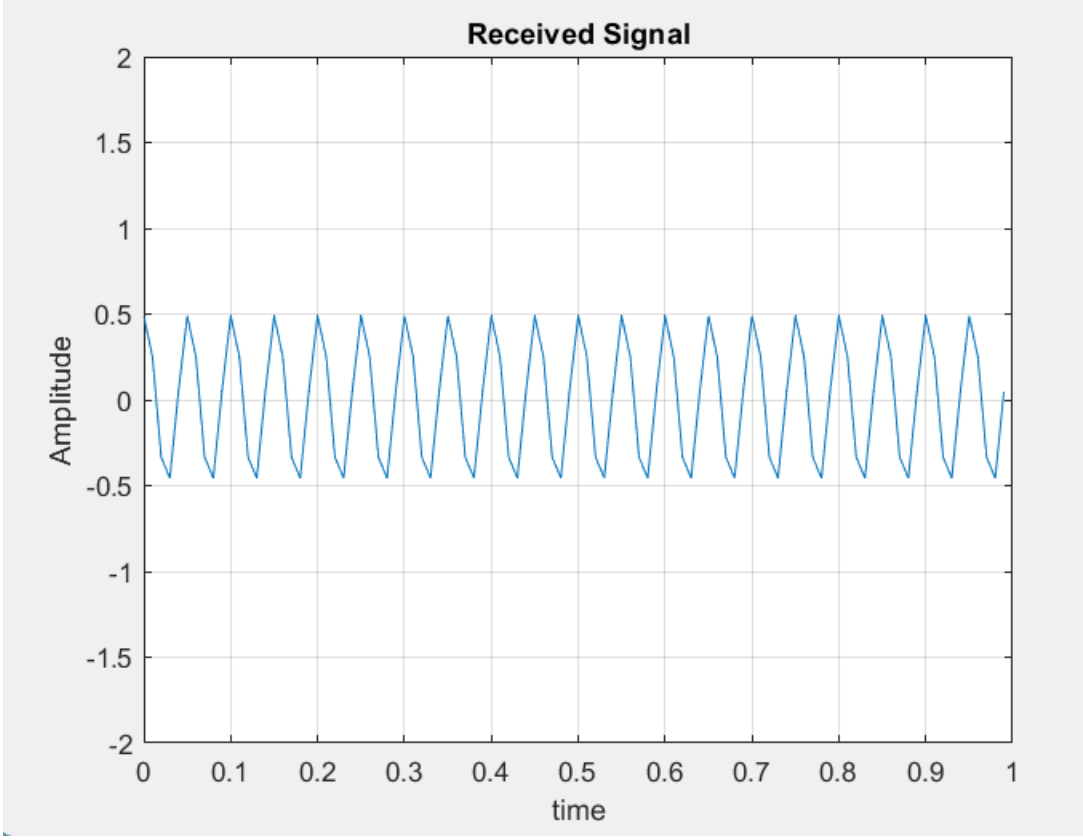




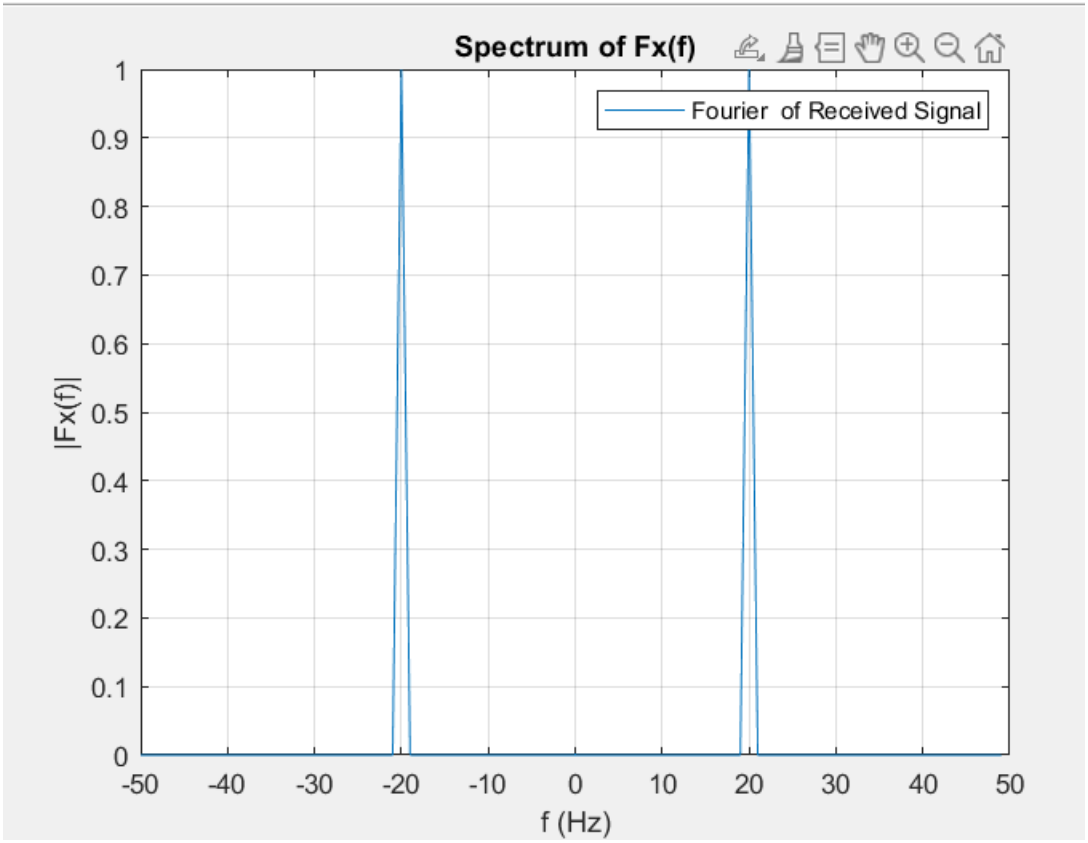
۳-۱:

در این بخش فرکانس دریافتی را به حوزه فوریه میبریم و سیگنال سینوسی شکل داریم که دارای دو فرکانس ضربه است و با ترسیم آن ، پاسخ زوج آن مشخص است و با پیدا کردن فرکانسی که در حوزه فوریه بیشتر است f new را که حاصل جمع fd و fc است را داریم و از آنجایی که مقدار fc مشخص است پس با کمک آن fd را پیدا کرده . با کمک دستورات fft و fftshift فرکانس سیگنال فوق را یافته و به کمک دستور max می توانیم نقطه پیک فرکانسی را بیابیم ؛. همچنین دستور angle را نیز روی سیگنال پیاده میکنیم و مقدار آن را در همان موقعیت پیک فرکانس میابیم. این مقدار همان فاز است. با استفاده از فاز سیگنال می توان مدت زمان تاخیر را یافت و از روی تاخیر می توانیم به فاصله برسیم در آخر سرعت و فاصله را به واحد های قبل برمیگردانیم.

نتایج به شکل زیر خواهد بود :



سیگنال دریافتی به شکل بالا می شود.



همچنین اندازه نرمال شده سیگنال دریافتی در حوزه فوریه به شکل فوق می گردد.

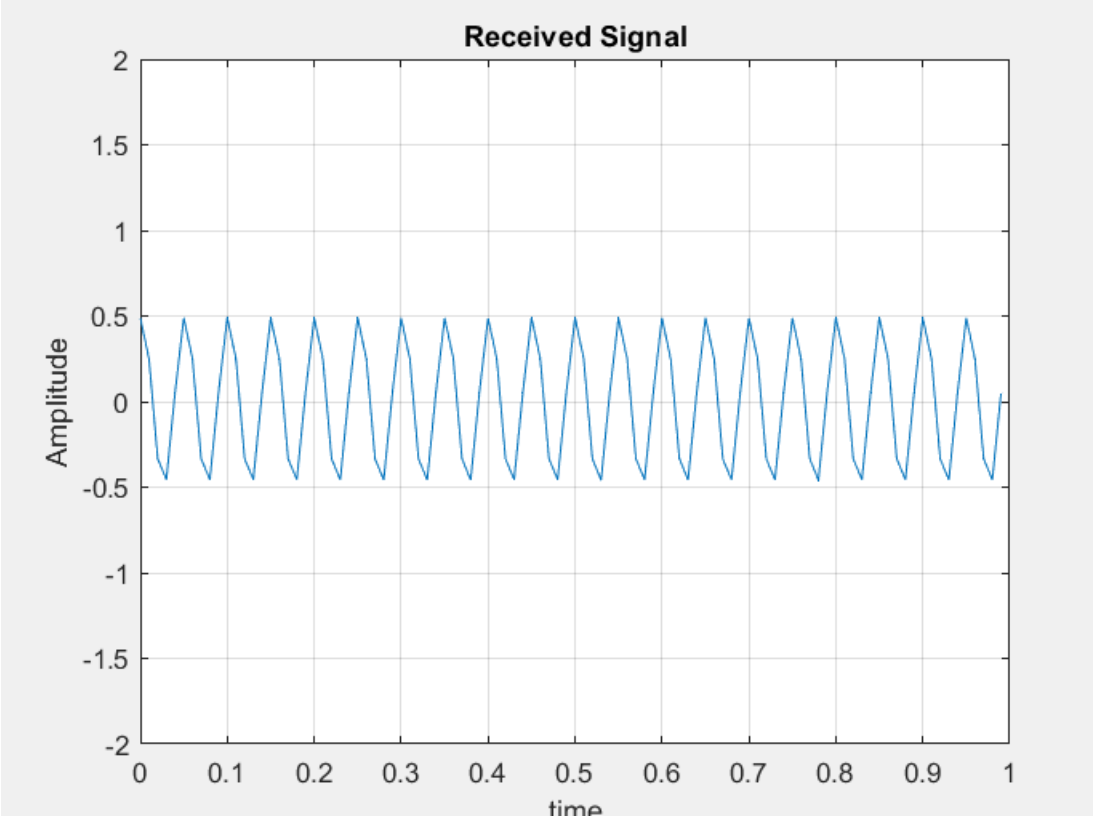
```
ans =

    "distance (R found) = 250.0000 km
      velocity (V found )  = 180.0000 km/h"

fx >>
```

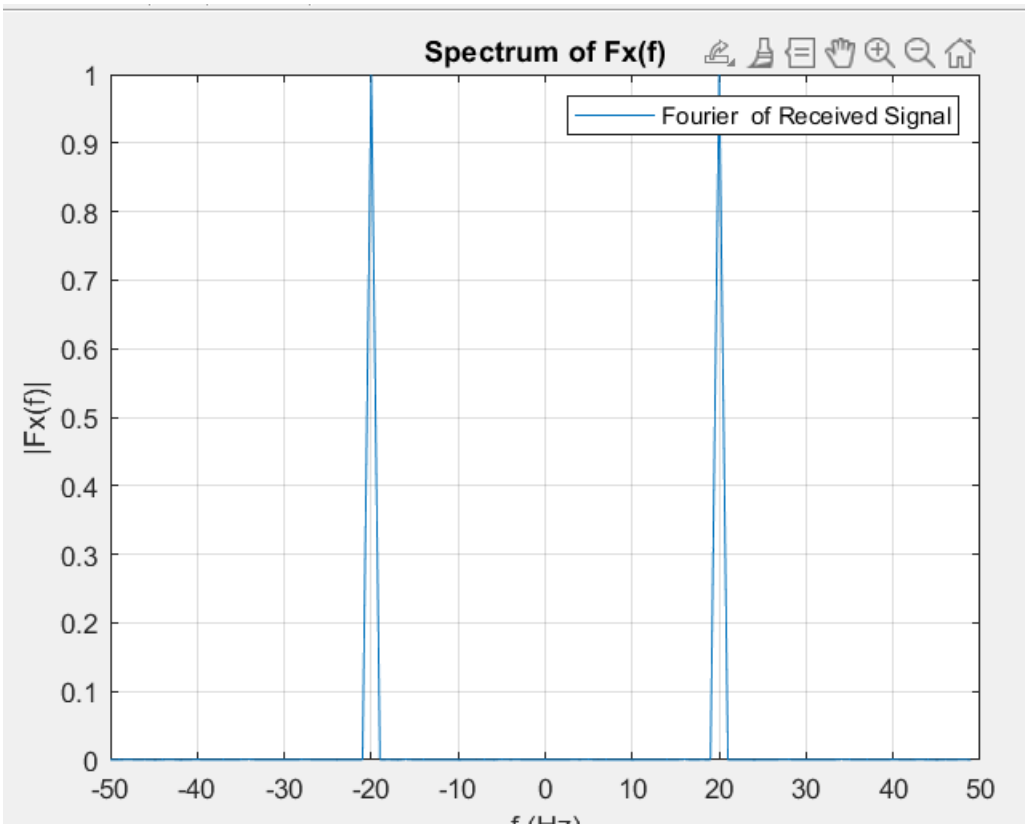
۴-۱:

در این مرحله می خواهیم با افزودن مقداری نویز به آن بخش پیشین را تکرار کنیم و قدرت آن را در چند مرحله افزایش دهیم.

بنا به کد با تغییر عدد نویز در متغیر sigma میتوانیم قدرت نویز را مشخص کنیم.

مثلا با نویز ۰.۰۰۱ نتایج زیر حاصل می شود:


Received Signal
Amplitude
time
2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Spectrum of Fx(f)
Fourier of Received Signal
|Fx(f)|
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-50
-40
-30
-20
-10
0
10
20
30
40
50

```
ans =

    "distance (R found) = 249.9025 km
      velocity (V found )  = 180.0000 km/h"

>>
```

عدد نویز را بیشتر کرده و ۰.۰۰۵ می رسانیم:

```
ans =

    "distance (R found) = 249.1284 km
      velocity (V found )  = 180.0000 km/h"

>>
```

مجدد آنرا بیشتر می کنیم و به ۰.۰۰۹ می رسانیم:

```
ans =

    "distance (R found) = 251.7293 km
      velocity (V found )  = 180.0000 km/h"

>>
```

اینبار با نویز ۰.۰۱ آنرا بررسی می کنیم:

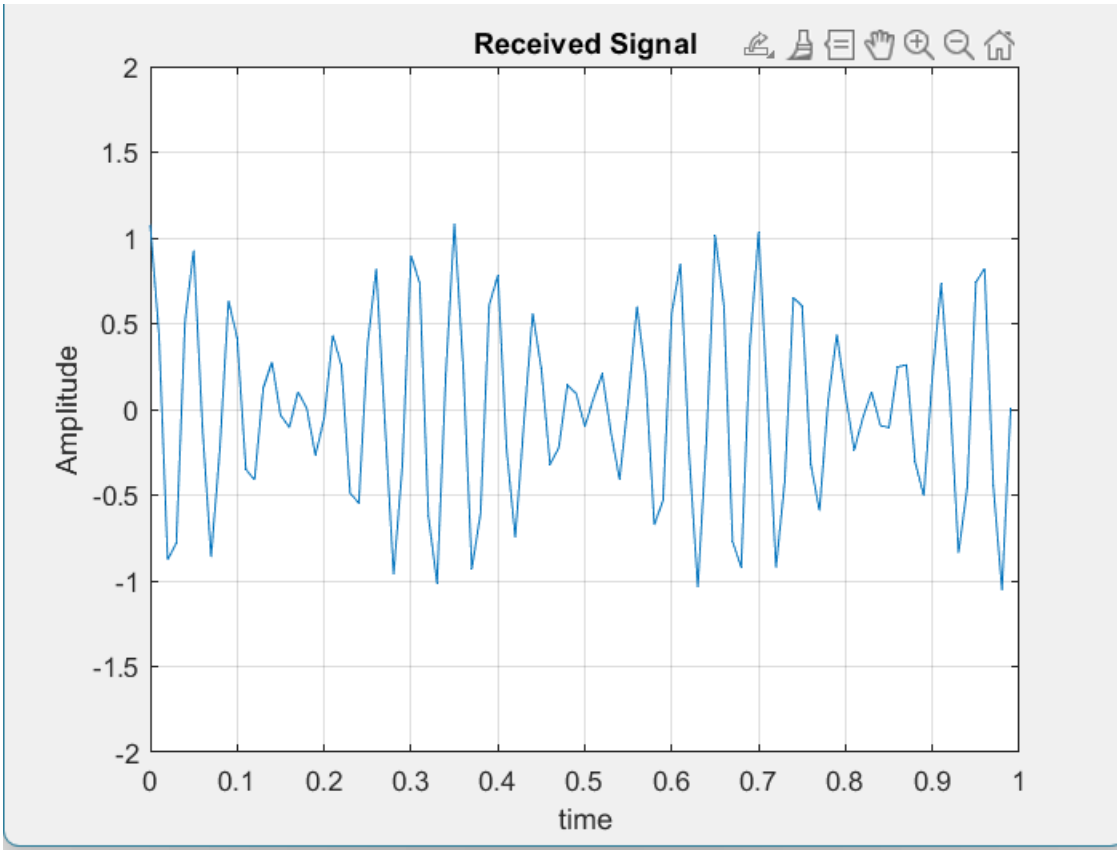
```
ans =

    "distance (R found) = 252.7009 km
      velocity (V found )  = 180.0000 km/h"

>>
```

چیزی که مشخص است سرعت جسم حساسیت زیادی نسبت به نویز ندارد و با اضافه کردن تا حدود یک برابر نویز به سیگنال درست محاسبه می شود. اما فاصله جسم با کوچکترین مقدار نویز با خطا تشخیص داده می شود و در کل پارامتر سرعت که با فرکانس به دست می آید مقاومت بهتری از فاصله – که با زمان حاصل می شود

دارد همچنین بهره گیری از فوریه برای تخمین فاصله کارایی اش را از دست داده و بیانگر قدرت آن در پارامتر های مربوط به فرکانس و ضعف آن در مقابل پارامتر های مرتبط با زمان می باشد.

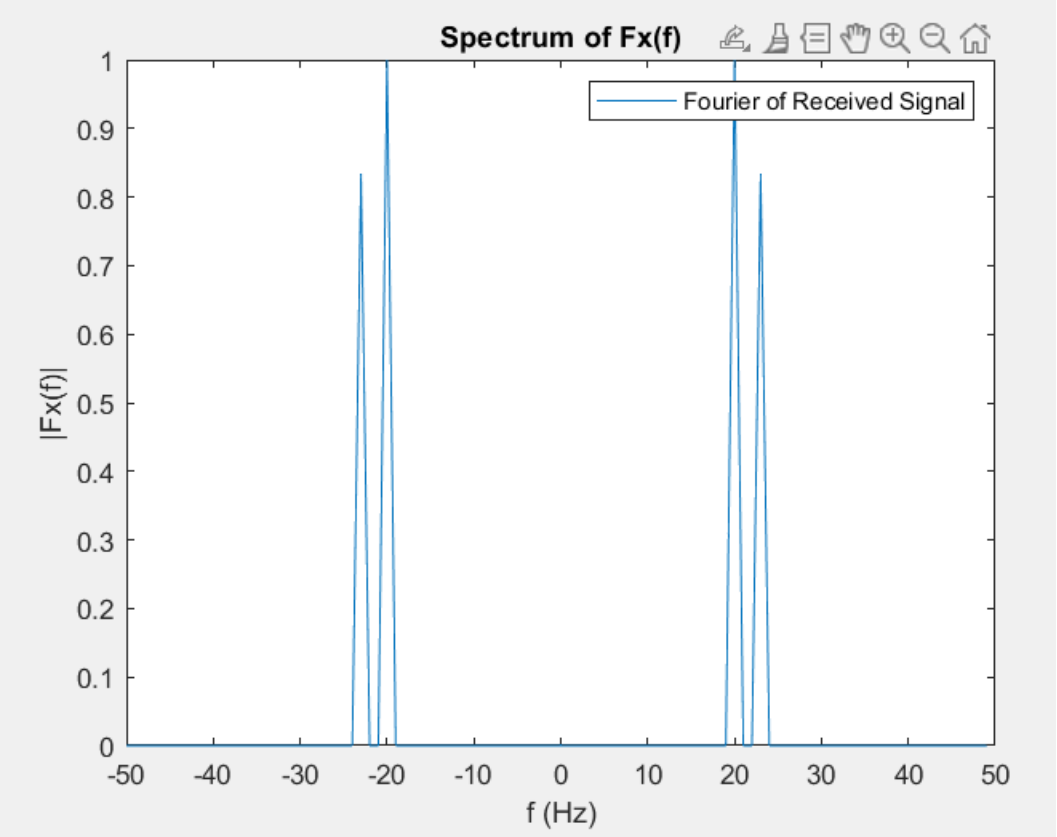
۵-۱:

در این بخش مشابه قسمت اول عمل می کنیم. سیگنال دریافتی مجموع سیگنال دریافتی از هریک از دو جسم می باشد.



۶-۱:

در این بخش با دید برعکس به آن مثل آنچه پیشتر انجام دادیم ، با گرفتن فوریه و محاسبه فرکانس متناظر با دو قله فاصله و سرعت را می توان به دست آورد . از threshold برای یافتن قله ها استفاده می کنیم.از آنجایی که محاسبات متلب تقریبی است به عدد تجربی ۰.۷۵ ماکسیمم دامنه قرار می دهیم .




```
1:
Estimated distance = 200.0000 km
Estimated velocity = 180.0000 km/h
2:
Estimated distance = 200.0000 km
Estimated velocity = 216.0000 km/h
```

۱-۷:

خیر از آنجایی که هم فرکانس میشوند روش فوریه دیگر کارایی نداشته پس میتواند به درستی سرعت را تشخیص داده و اما فاصله صحت عملکردش را از دست می دهد.

چنانچه حداقل قادر به تشخیص ۲ جسم باشد بایستی حداقل اختلاف سرعت ها به گونه ای باشد فرکانس ها ۱ هرتز با هم متفاوت باشند پس ۱۰۰/۳ =min (v1-v2) = 1/b min (f1-f2)

۱-۸:

بلی میتوانیم ، روش فوریه به خوبی کار میکند چونکه پارامتر وابسته به فرکانس تغییری نکرده است .

۹-۱:

پس از اعمال دستور fft و fftshift بردار تعدادی پیک می زند که ما تعداد آنها را نمیدانیم. 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 آن مقایسه کرد. اگر نمودار پیک زده باشد این مقدار بزرگ است. پس با تعیین یک آستانه و مقایسه اختالف دو نقطه با آن، می توان هر تعداد پیک را پیدا کرد و فرکانس و فاز آنها را یافت.

۲:

۲-۱:

در این بخش دو آرایه تعریف می کنیم که در آنها نام نت ها و همچنین فرکانس آنها را ذخیره می کنیم و فرکانس هارا به همان ترتیبی داریم قرار داده و آهنگ مورد نظر را به صورت سل ذخیره سازی کرده در ردیف نخست آن نام هر نت و در ردیف دوم آن طول را به ۱ یا ۰.۵ نوشته و هر نت را می یابیم.

همچنین هر سمپل bitspersampleآن به کمک دستور audioinfo به شکل زیر می شود.

```
>> audioinfo('mysong.wav')

ans =

  struct with fields:

            Filename: 'C:\Users\Amirali\Desktop\UT\سیگنال\CA\CA6\p2\mysong.wav'
    CompressionMethod: 'Uncompressed'
          NumChannels: 1
           SampleRate: 8000
         TotalSamples: 70000
             Duration: 8.7500
                Title: []
              Comment: []
               Artist: []
        BitsPerSample: 16
```

۳-۲:

فایل مورد نظر را خوانده و ذخیره سازی می کنیم و دو آرایه مشابه قبل تعریف می کنیم .

در حلقه تا زمانی که طول آن به ۰ نرسیده است ادامه داده و به دنبال سکوت که در اینجا به صورت دو صفر پشت سر هم تعریف شده است می گردیم و آ را از ابتدا تا آن نقطه قطع کرده که این نت است و در سلولی که تعبیه شده ذخیره می کنیم ۲۰۰ خانه که اندازه طول سکوت است آ را ذخیره کرده و با اجرای مجدد حلقه و تمام نت ها خارج می شوند. برای پیدا کردن فرکانس به ازای هر یک fft و fftshift اجرا کرده و با پیک آنرا یافت کرده و و فرکانس مورد نظر بدست می آید از آنجایی که فرکانس حاصل نویز دارد برای اینکه بفهمیم مربوط به کدام نت است با تعریف آستانه اختلاف را بررسی کرده و نت به هر یک نسبت داده می شود در حلقه بعدی با مقادیر بدست امده هر یک مقایسه شده و چنانچه شرط صدق کند نام نت مورد نظر را استخراج کرده و ذخیره سازی می کنیم. در نهایت ارایه ای دو بعدی از نام و طول نت های موسیقی خواهیم داشت.

خروجی آن به شکل فوق خواهد شد :

```
final_result =

  2×44 cell array

  Columns 1 through 9

    {["D"   ]}    {["D"   ]}    {["G"]}    {["F#"]}    {["D"]}    {["D"   ]}    {["E"   ]}    {["E"   ]}    {["D"   ]}
    {[0.5000]}    {[0.5000]}    {[  1]}    {[   1]}    {[  1]}    {[0.5000]}    {[0.5000]}    {[0.5000]}    {[0.5000]}

  Columns 10 through 19

    {["F#"  ]}    {["D"   ]}    {["E"]}    {["D"]}    {["E"]}    {["F#"]}    {["E"]}    {["D"   ]}    {["E"   ]}    {["E"   ]}
    {[0.5000]}    {[0.5000]}    {[  1]}    {[  1]}    {[  1]}    {[   1]}    {[  1]}    {[0.5000]}    {[0.5000]}    {[0.5000]}

  Columns 20 through 29

    {["D"   ]}    {["F#"  ]}    {["D"   ]}    {["E"]}    {["D"]}    {["E"   ]}    {["D"   ]}    {["F#"]}    {["E"]}    {["D"]}
    {[0.5000]}    {[0.5000]}    {[0.5000]}    {[  1]}    {[  1]}    {[0.5000]}    {[0.5000]}    {[   1]}    {[  1]}    {[  1]}

  Columns 30 through 38

    {["E"   ]}    {["D"   ]}    {["F#"]}    {["E"]}    {["D"   ]}    {["D"   ]}    {["E"]}    {["F#"  ]}    {["E"   ]}
    {[0.5000]}    {[0.5000]}    {[   1]}    {[  1]}    {[0.5000]}    {[0.5000]}    {[  1]}    {[0.5000]}    {[0.5000]}

  Columns 39 through 44
```