باسمه تعالی

گزارش تمرین کامپیتوری سری اول

پردازش سیگنال EEG

دانشکده مهندسی برق

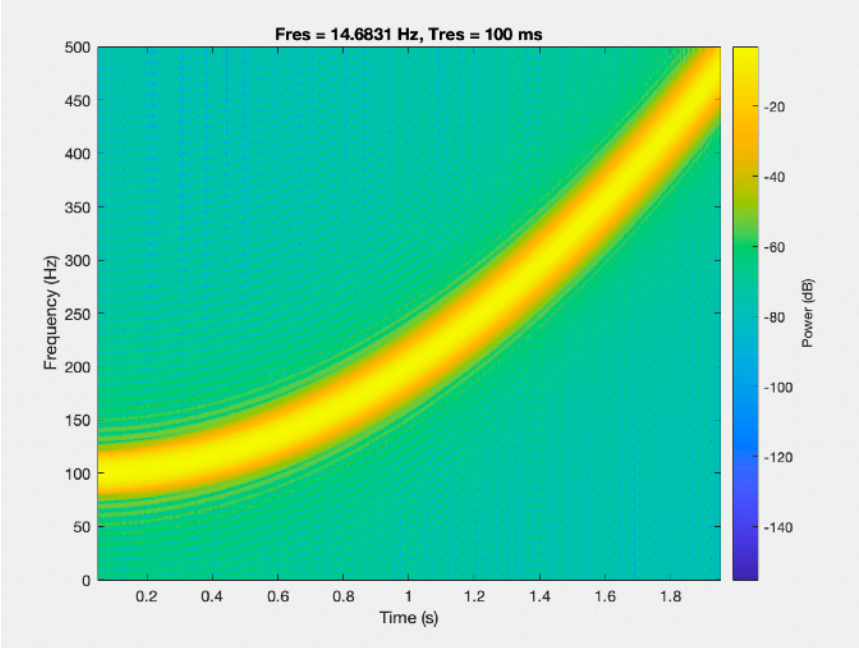
استاد: سپیده حاجی‌پور

سؤال ۱ -

الف)

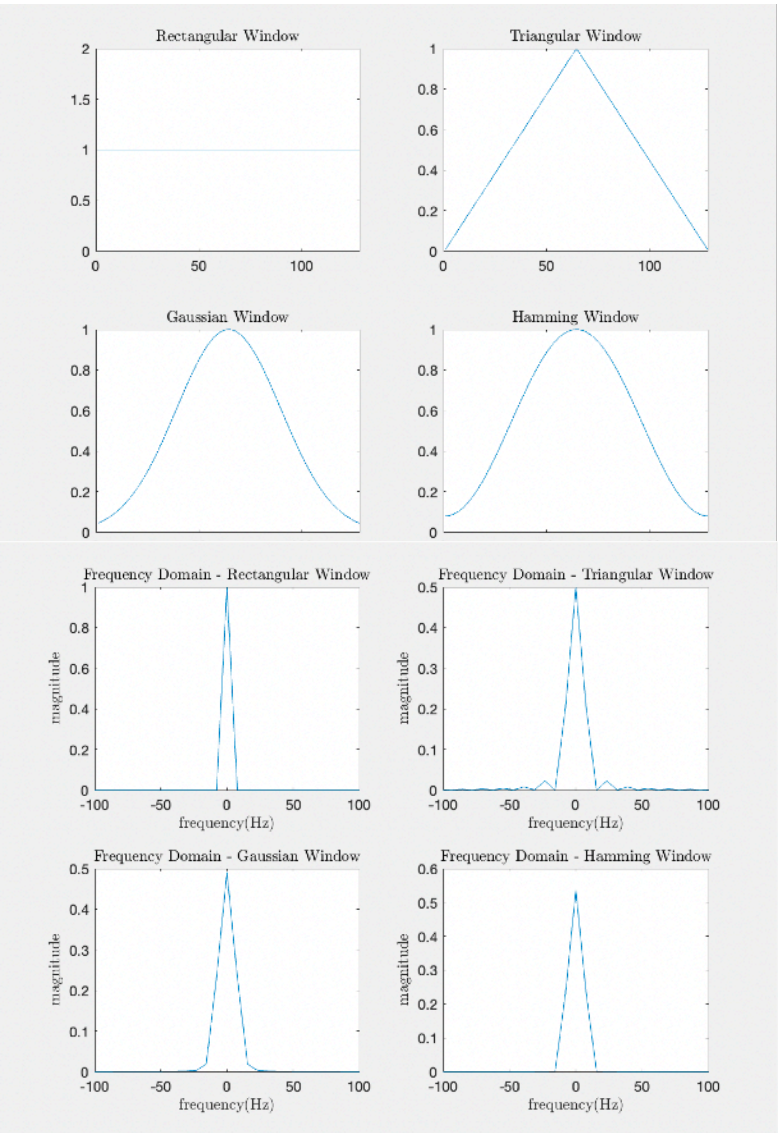
```
%% Q1 - Part A
clc; close all; clear;

fs = 1000;
t = 0:1/fs:2;
f0 = 100;
beta = 100;
x = chirp(t,100,1,f0+beta,'quadratic');
pspectrum(x,1e3,'spectrogram','TimeResolution',0.1, ...
    'OverlapPercent',99,'Leakage',0.85);
```



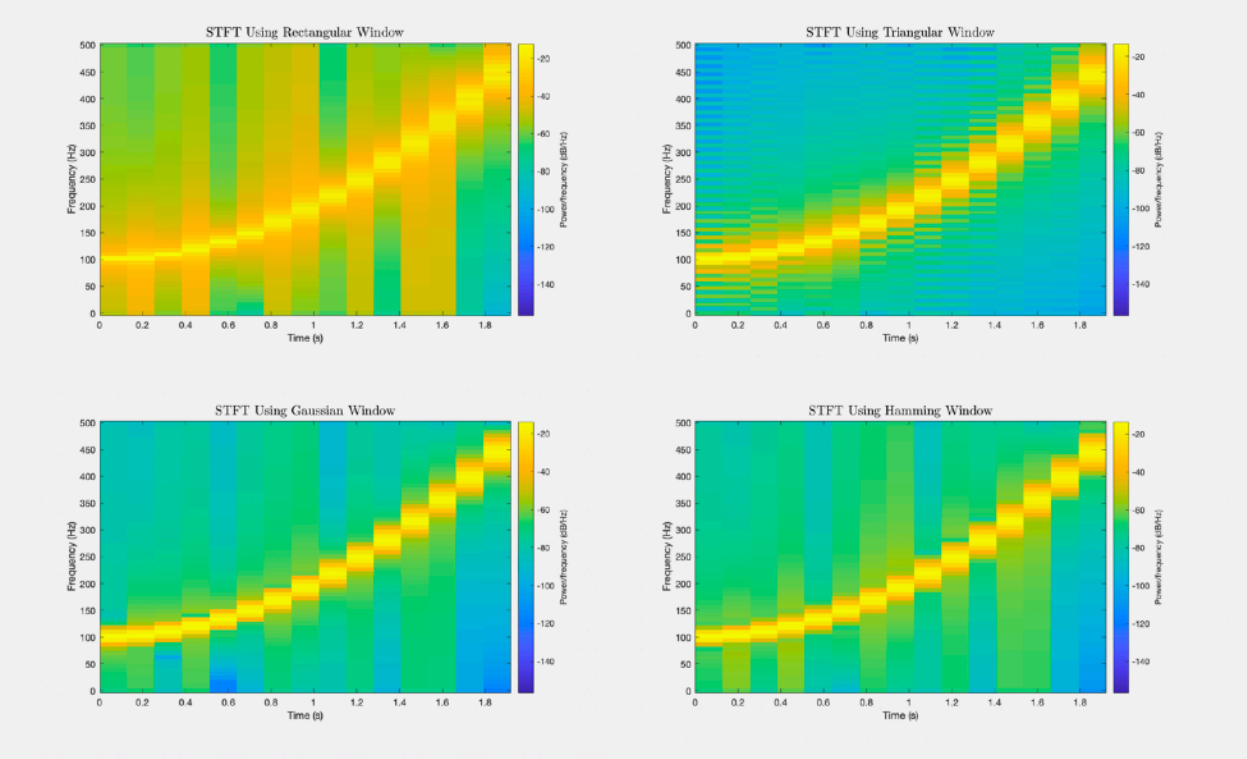
با مشاهده سیگنال در حوزه فرکانس-زمان متوجه می‌شویم که با جلو رفتن زمان فرکانس سیگنال به صورت درجه دو افزایش پیدا می‌کند که این یعنی که سیگنال به درستی ساخته شده است.

ب)



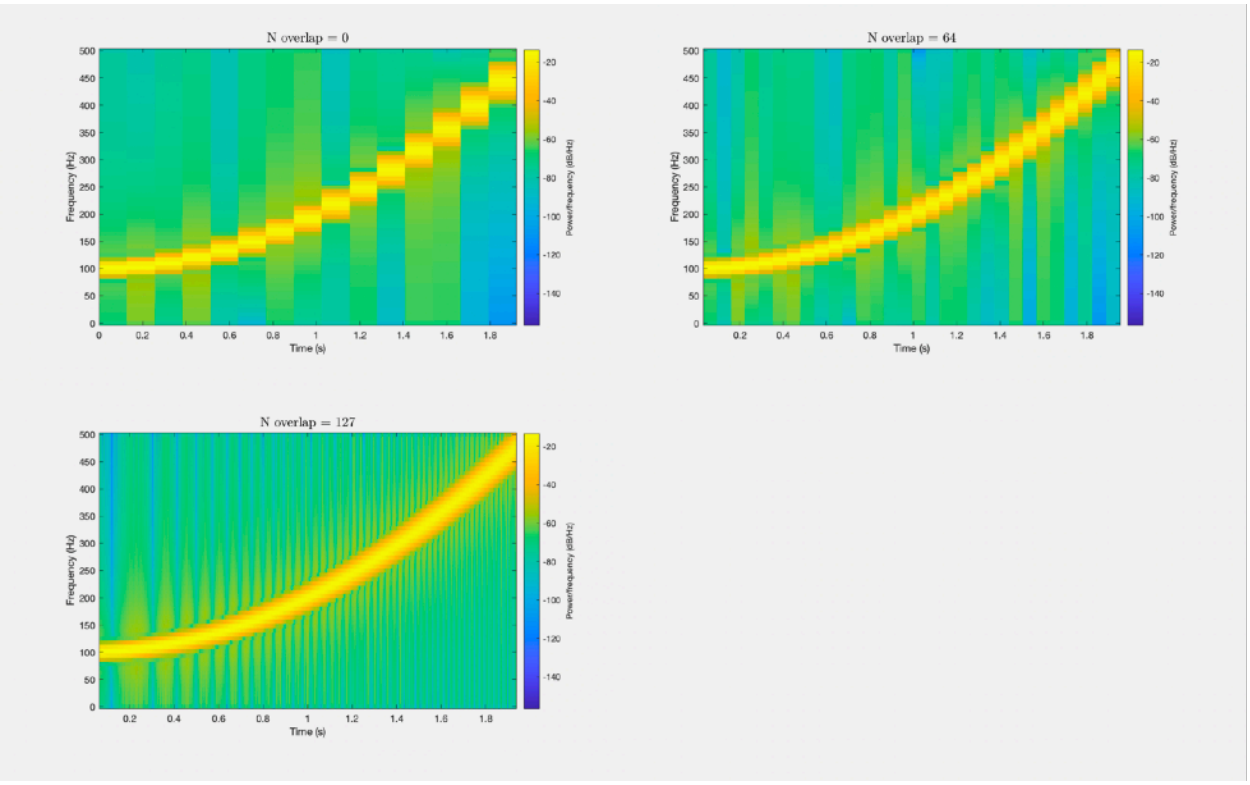
همانطور که مشاهده می‌شود پنجره مستطیلی تغییر ناگهانی از یک به صفر دارد که این تغییر ناگهانی باعث می‌شود که هنگام انجام پنجره گذاری روی سیگنال، اطلاعاتی که در انتهای پنجره وجود دارند دچار مشکل شوند و برای همین ما معمولا از دو پنجره پایینی استفاده می‌کنیم یعنی گاوسی و همینگ؛ چون که این پنجره‌ها خیلی نرم تر هستند و تغییرات ناگهانی ندارند و به عبارتی در همه قسمت مشتق پذیر می‌باشند.

پ)



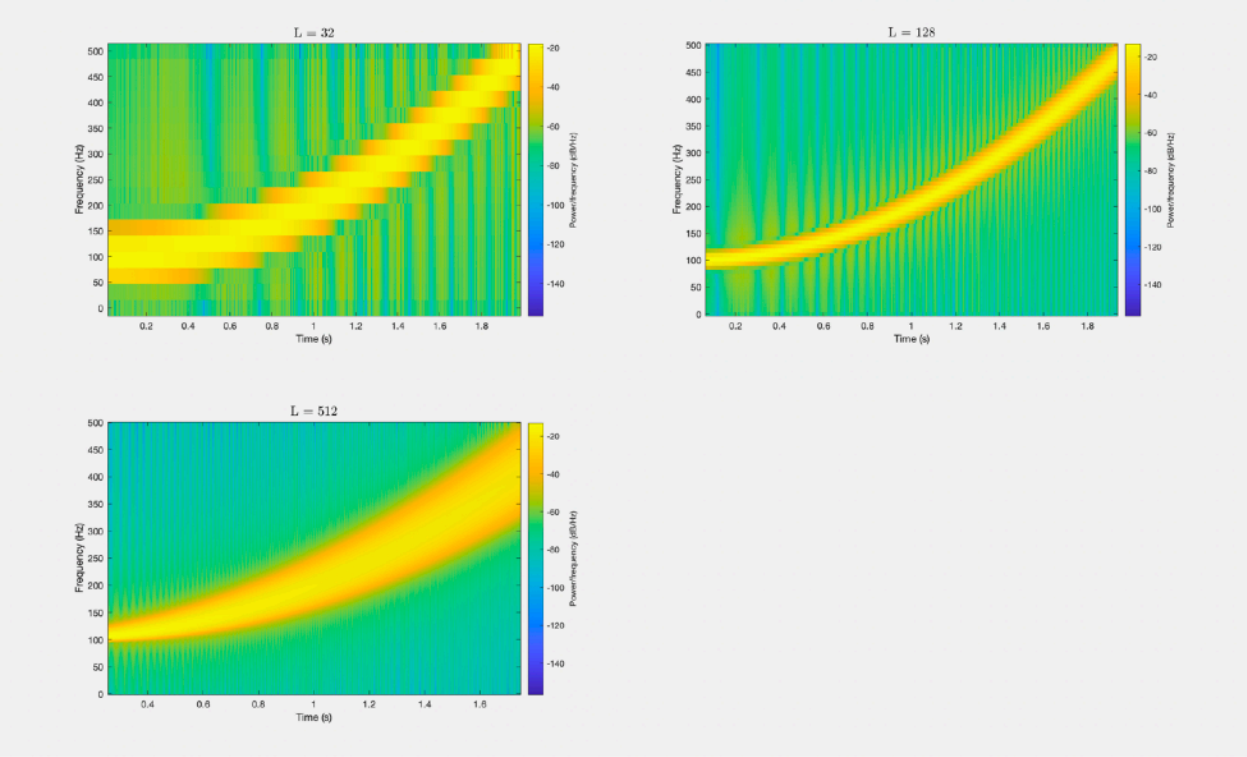
همانطور که دیده می‌شود نتایج پنجره مستطیلی با اختلاف از همه نتایج ضعیف تر هست اما پنجره‌های دیگر عملکرد نسبتا خوبی داشتند. حتی می‌شود گفت که بر روی این سیگنال عملکرد پنجره مثلثی از دو نوع دیگر بهتر بوده چون فرکانس‌های دیگر به جر فرکانس دلخواه در این شکل کمتر زرد شده اند که نشان می‌دهد رزولوشن فرکانسی بهتری داشتیم. البته با تغییر دادن میزان هم پوشانی خواهیم دید که پنجره همینگ هم نتیجه خیلی خوبی می‌تواند داشته باشد.

ت)



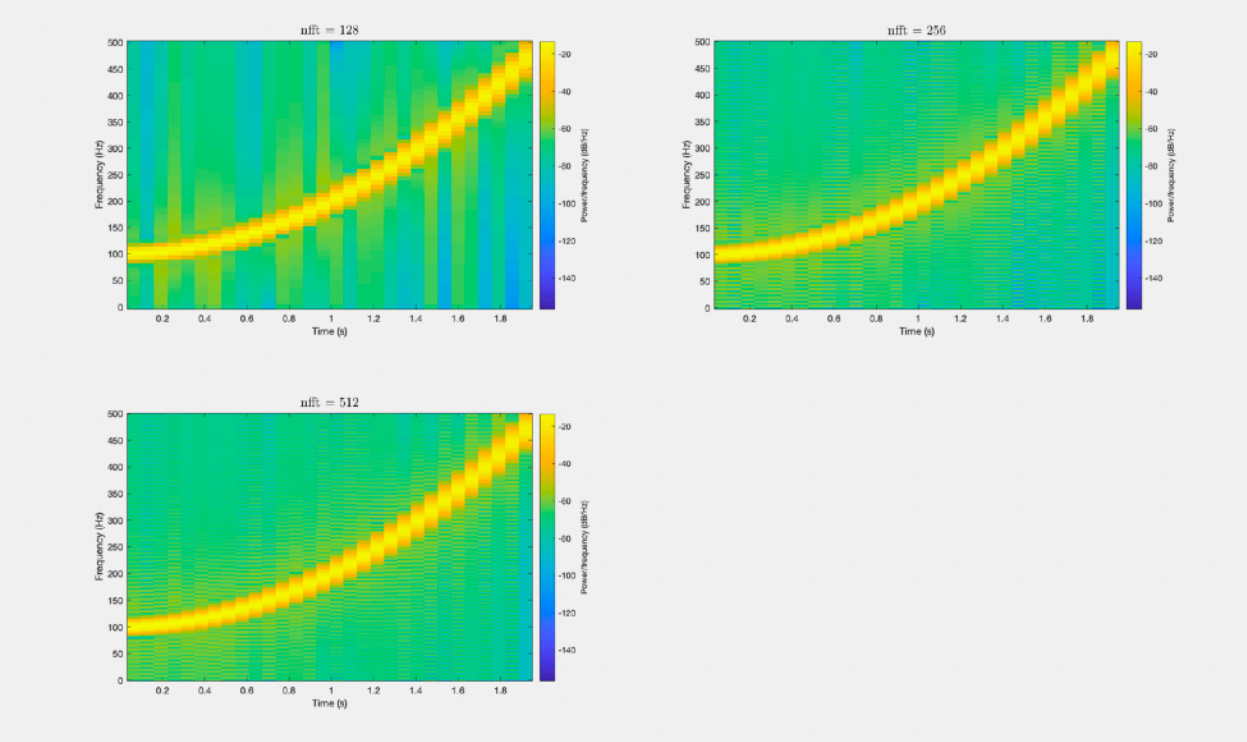
ما پنجره همینگ را استفاده کرده ایم و مشاهده می‌کنیم که با افزایش نقاط هم پوشان به نتایج دقیق تری رسیده ایم. به طور کلی دلیل استفاده از نقاط هم پوشان این هست که همانطور که در قسمت ب دیدیم، شکل پنجره همینگ در گوشه‌ها نسبت به وسط پنجره دامنه کمتری دارد و برای همین ما از هم پوشانی استفاده می‌کنیم که مطمئن باشیم تمام نقاط سیگنال در نهایت وزن مشابهی دریافت کرده باشند. به صورت یک قاعده کلی معمولی نصف طول پنجره را به عنوان نقاط هم پوشان استفاده می‌کنند و در اکثر سیگنال‌ها همین مقدار می‌تواند نتایج خوبی به ما بدهد.

ث)



اثر تغییر طول پنجره به این صورت است که هر چقدر طول پنجره بیشتر باشد، رزولوشن زمانی ما کاهش پیدا می‌کند چون نمی‌دانیم که در کدام قسمت این پنجره چنین فرکانس‌هایی داشتیم اما از آنجایی که همیشه ترید آفی بین رزولوشن زمانی و فرکانسی برقرار هست، رزولوشن فرکانسی ما بالاتر می‌رود و بهتر می‌توانیم تشخیص دهیم که چه فرکانسی‌هایی در سیگنال وجود دارند. شکل‌های کشیده شده هم این گزاره را تایید می‌کنند.

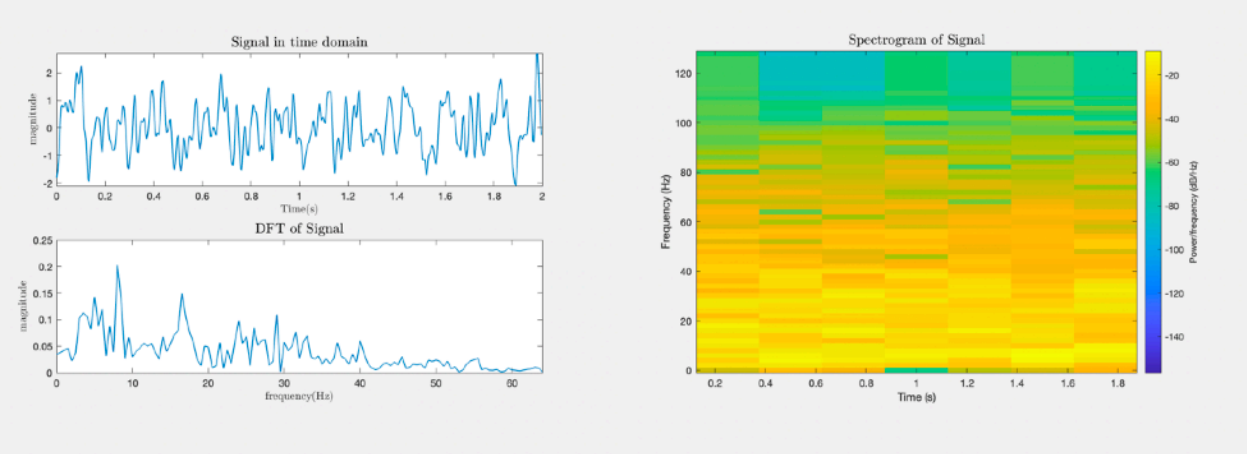
ج)



به طور کلی افزایش تعداد نقاط DFT متناظر به این هست که ما تعداد بیشتری سمپل فرکانسی از روی تبدیل فوریه اصلی برداشته‌ایم و این یعنی که رزولوشن فرکانسی افزایش پیدا می‌کند. همانطور که در شکل‌ها دیده می‌شود با افزایش تعداد نقاط، خطای اسپکتوگرام ما کم می‌شود و فرکانس‌های موجود را با دقت بیشتری پیدا می‌کند که این معادل همان افزایش رزولوشن فرکانسی می‌باشد.

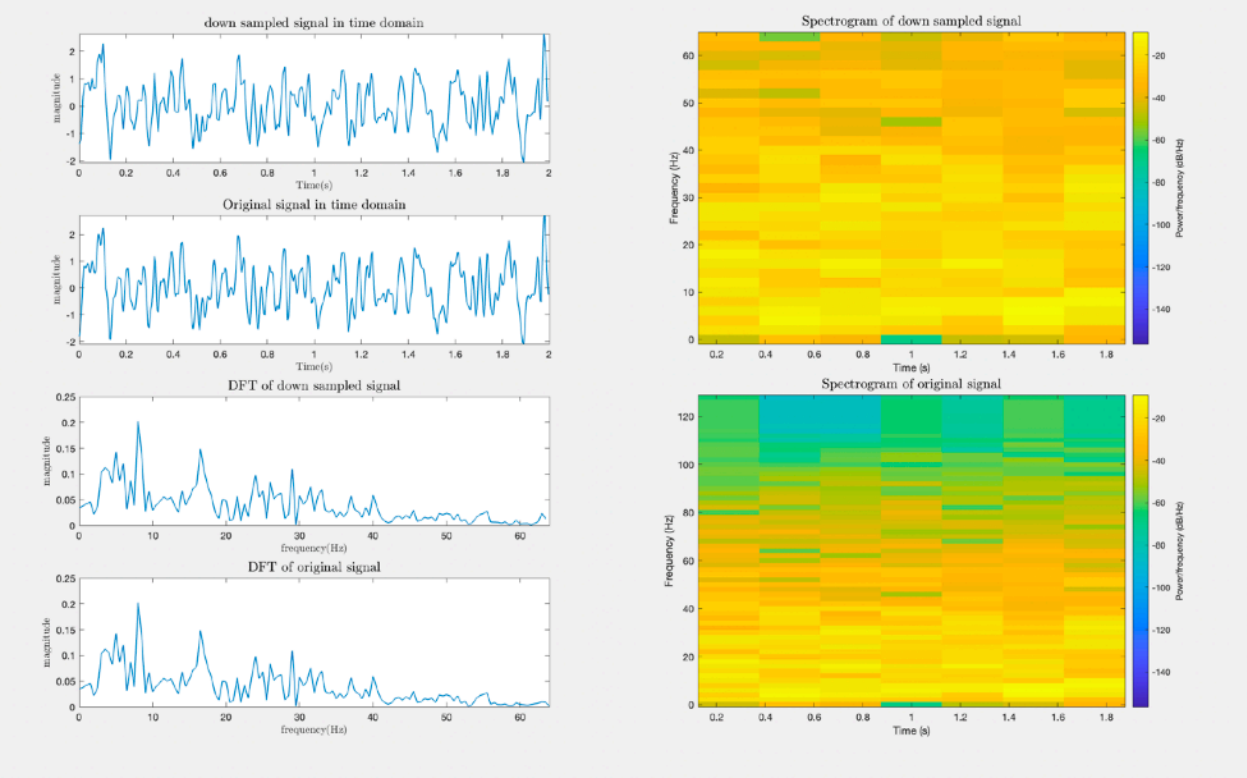
سؤال ۲ -

الف)



خواسته‌های مسئله در شکل بالا آورده شده است، تنها موضوعی که مهم بود این هست که در بخش اسپکتروگرام، طول پنجره را 128 انتخاب کرده ایم که دلیل این موضوع این هست که بتوانیم رزولوشن فرکانسی بالاتری داشته باشیم و متوجه شویم که به صورت کلی (بدون دانستن زمان دقیق) چه فرکانس‌هایی بیشتر در این سیگنال حضور دارند که همانطور که دیده می‌شود تقریبا فرکانس‌های بالای ۶۰ هرتز دامنه بسیار کمی دارند.

ب)

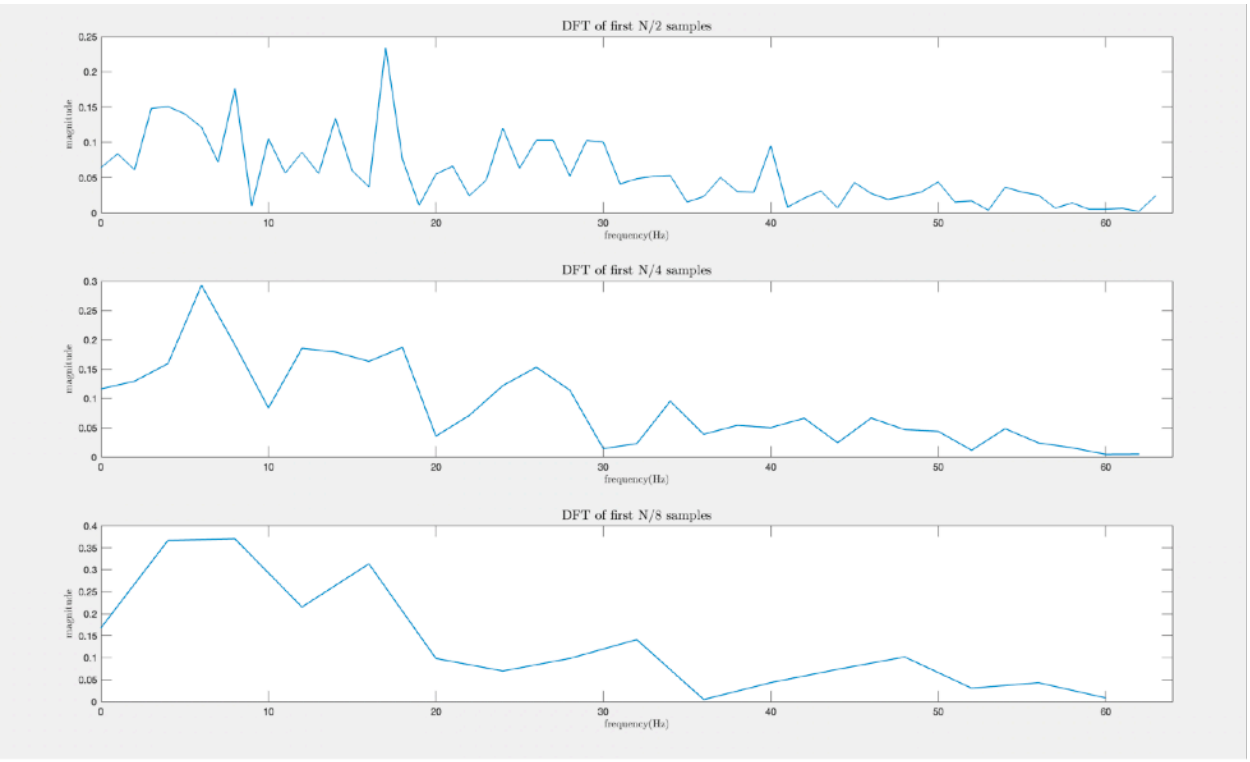


در این بخش با توجه به اینکه در قسمت الف اشاره کردیم که عمده محتوای فرکانسی این سیگنال در فرکانس‌های زیر ۶۰ هرتز است، در ابتدا یک فیلتر پایین گذر ۶۴ هرتز استفاده کردیم و سپس سیگنال را با نرخ ۲ داون سمپل کردیم.

همانطور که در شکل‌ها دیده می‌شود هیچ تغییری در طیف فرکانسی ایجاد نشده است که این موضوع نشان دهنده این هست که ما اطلاعات زیادی را از دست نداده ایم. دقت شود که تفاوت دو اسپکتروگرام به دلیل تغییر محور فرکانس از ۱۲۸ هرتز به ۶۴ هرتز است.

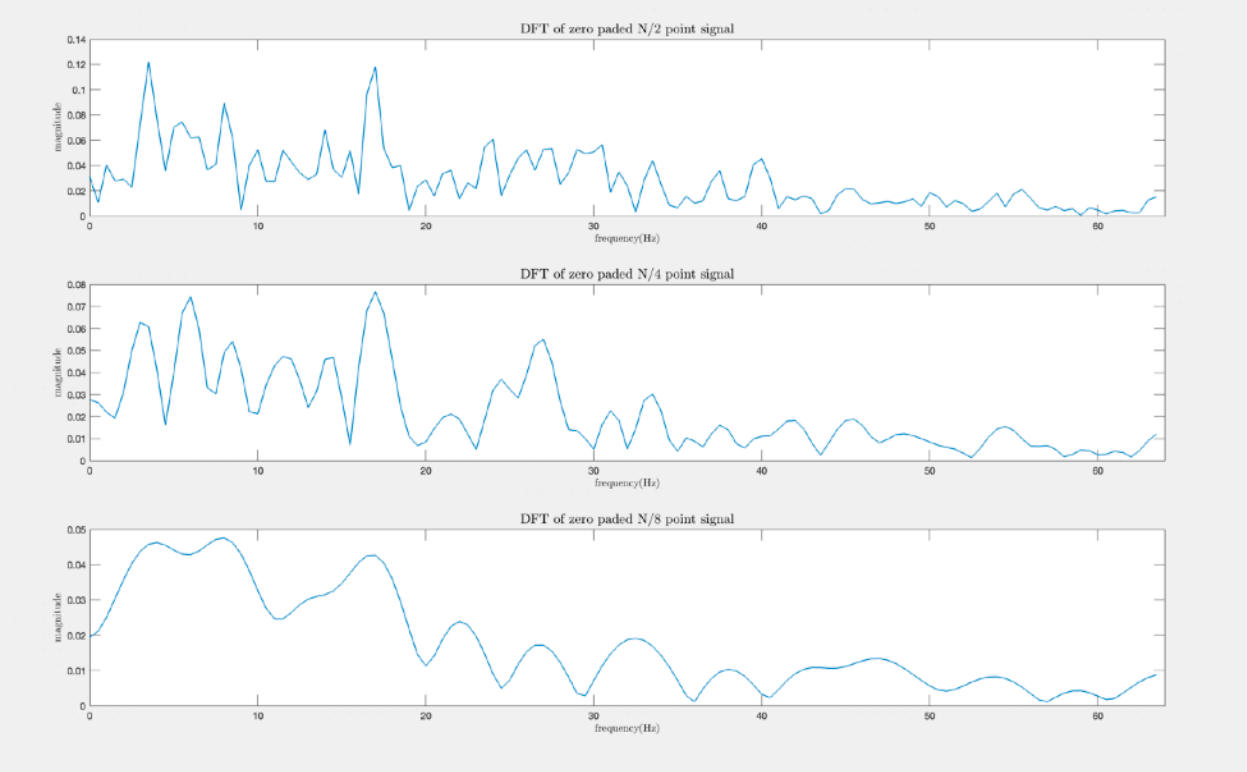
```
filtered_signal = lowpass(signal,64,fs);
downsampled_signal = downsample(filtered_signal,2);
```

پ)



در این قسمت تنها بخشی از ابتدای سیگنال را نگه داشته‌ایم و از آن DFT گرفته ایم. و میبینم که هر چقدر مقدار کمتری از سیگنال را نگه داشته‌ایم، طیف فرکانسی ما بیشتر دچار تغییر شده است اما همچنان با سیگنال اصلی شباهت دارد. چنانچه یک سیگنال کاملا ایستا داشتیم انتظار داشتیم که با برش زدن هر قسمت از آن به یک طیف فرکانسی مشابه برسیم.

ت)



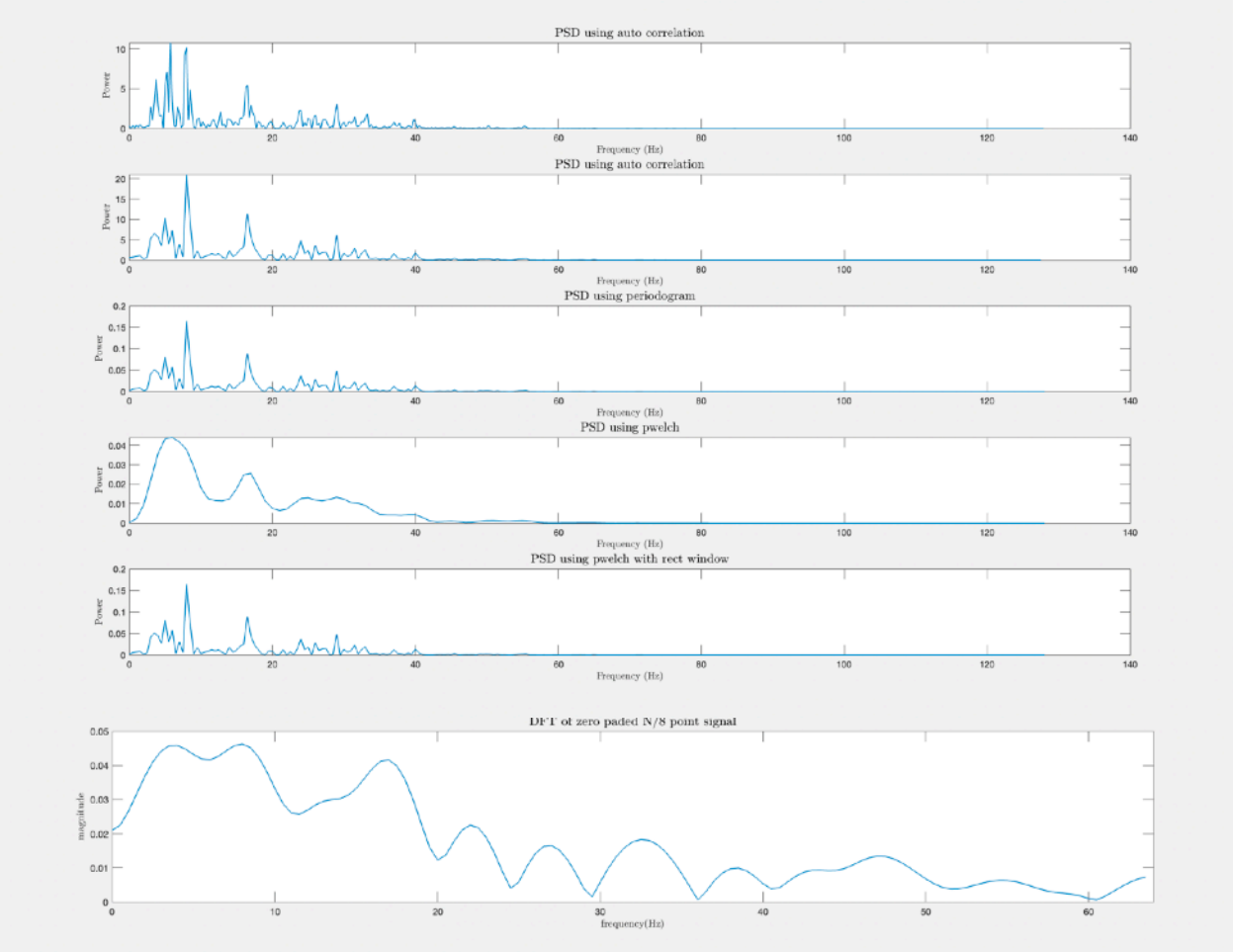
همانطور که دیده می‌شود با انجام دادن این فرآیند به تخمین بهتری از طیف فرکانسی اصلی می‌رسیم. به طور کلی زیرو پدینگ به دو علت اصلی انجام می‌شود، دلیل اول همین هست که با انجام دادن زیرو پدینگ و گرفتن DFT با تعداد نقاط بیشتر، می‌تواند تعداد سمپل بیشتری از تبدیل فوریه اصلی برداشته باشیم و به همین دلیل تخمین بهتری از طیف فرکانسی خواهیم داشت. دلیل دوم هم این هست که الگوریتم fft برای تعداد نقاطی که توان دو باشند خیلی سریعتر هست ، در همین راستا اگر یک سیگنال۱۰۰۰ نقطه داشته باشد ترجیح بر این هست که آن را با زیرو پدینگ ۱۰۲۴ نقطه‌ای کنیم و سپس DFT بگیریم.البته به صورت دیفالت در متلب اگر عدد تعداد نقاط DFT بیشتر از طول سیگنال باشد، خودش عملیات زیرو پدینگ را انجام می‌دهد.

ث)

در هر یک از قسمت‌ها توضیحات مربوطه داده شده است.

سؤال ۳-

من به پنج صورت چگالی طیف توان را بدست آوردم که به ترتیب توضیح خواهم داد:



روش اول:

```
% Using correlation
fs = 256;
signal_acorr = xcorr(signal);
L = length(signal_acorr);
f = (0:L/2)*(fs/L);
m = fft(signal_acorr);
m = m(1:round(L/2));
fft_signal_acorr = abs(m)/L;
```

در این روش در ابتدا کرولیشن گرفتیم و سپس از آن DFT گرفتیم. انتظار داریم که به جواب درست رسیده باشیم اما همانطور که در شکل بالا دیده می‌شود جواب درستی حاصل نشده است. دلیل این موضوع این هست که طبق تعریف ما باید از اتو کرولیشن تبدیل فوریه بگیریم تا به چگالی طیف توان برسیم ولی ما اینجا DFT میگیریم که این باعث تغییر جواب می‌شود، به خاطر اینکه ما به جای کرولیشن معمولی اینجا باید برای اینکه به جواب درست

برسیم، کرولیشن دایروی بگیریم، یعنی وقتی سیگنال را شیفت می‌دهیم آن قسمت آخرش که از پنجره خارج می‌شود را دوباره برگردانیم و به اول سیگنال اضافه کنیم. چون که در خواص DFT متناظر با ضرب در حوزه فرکانس، کانولوشن عادی نیست بلکه کانولوشن دایروی هست.

روش دوم:

```
% Using Circular Correlation
signal_acorr = ifft(fft(signal).*conj(fft(signal)));
L = length(signal_acorr);
f = (0:L/2-1)*(fs/L);
m = fft(signal_acorr);
m = m(1:round(L/2));
fft_signal_acorr = abs(m)/L;
```

در این بخش ما می‌خواهیم که کورلیشن عادی را به کورلیشن دایروی تغییر بدهیم. بدین منظور می‌توانستیم این کار را به صورت دستی انجام بدهیم، اما ما کار بهتری انجام داده‌ایم. اتو کورلیشن عادی برابر است با کانولوشن سیگنال با مزدوج خودش، حال برای اینکه کرولیشن دایروی بگیریم کافی است که به جا کانولوشن عادی، کانولوشن دایروی بگیریم. برای اینکار ما از خواص DFT استفاده کرده ایم، بدین صورت که ما میدانیم متناظر ضرب در حوزه فرکانس می‌شود کانولوشن دایروی در حوزه زمان؛ به همین خاطر ما DFT سیگنال و مزدوج تبدیل فوریه آن را در هم ضرب کرده ایم و سپس DFT معکوس گرفته ایم تا به کرولیشن دایروی برسیم. همانطور که دیده می‌شود با انجام اینکار چگالی طیف توان بدست آمده مشابه با حالت‌های دیگر می‌شود.

روش سوم:

```
[pxx,f] = periodogram(signal,[],[],fs);
plot(f,pxx,'Linewidth',1);
```

در این بخش از دستور periodogram استفاده کردیم که نتیجه درستی بدست آمد.

روش چهارم:

```
[pxx,f] = pwelch(signal,[],[],[],fs);
subplot(5,1,4);
plot(f,pxx,'Linewidth',1);
```

در این روش از pwelch استفاده کردیم که می‌بینیم نتیجه با حالات قبل متفاوت است. دلیل این تفاوت این هست که اگر به این تابع نوع پنجره مد نظر را ندهیم به صورت دیفالت آن را hamming در نظر می‌گیرد که با تابع periodogram که به صورت دیفالت پنجره را rect می‌گیرد تفاوت دارد؛ برای همین نتایج متفاوت شده است.

روش پنجم:

```
L = length(signal);
[pxx,f] = pwelch(signal,rectwin(L),[],[],fs);
subplot(5,1,5);
plot(f,pxx,'Linewidth',1);
```

در این بخش تنها تفاوتی که ایجاد کردیم این هست که پنجره مستطیلی به طول سیگنال را به عنوان ورودی به تابع pwelch دادیم و همانطور که دیده می‌شود نتایج مشابه حالت‌های قبل شد.