به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

پردازش سیگنال‌های زمان–گسسته

تمرین سری 3

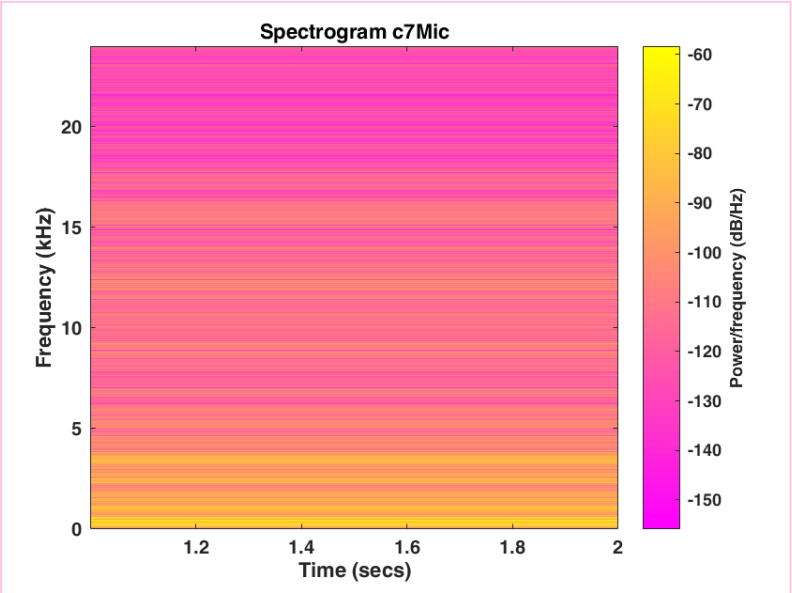
نام و نام خانوادگی	نرجس نورزاد
شماره دانشجویی	810196626
تاریخ ارسال گزارش	4مرداد 1399

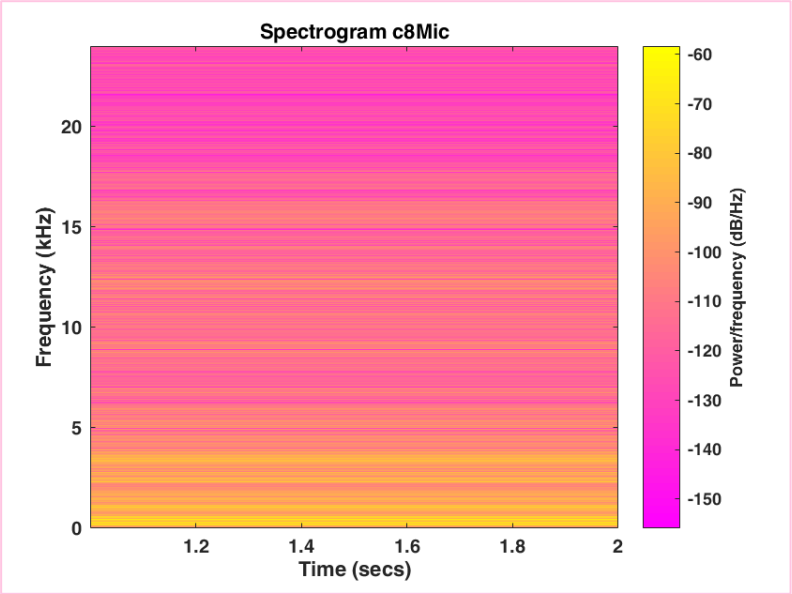
فهرست گزارش سوالات

.1

در این قسمت ابتدا 2 فایل صوتی موجود در Intro خوانده شده و یک سیگنال و یک فرکانس نمونه برداری از آن استخراج شده است .

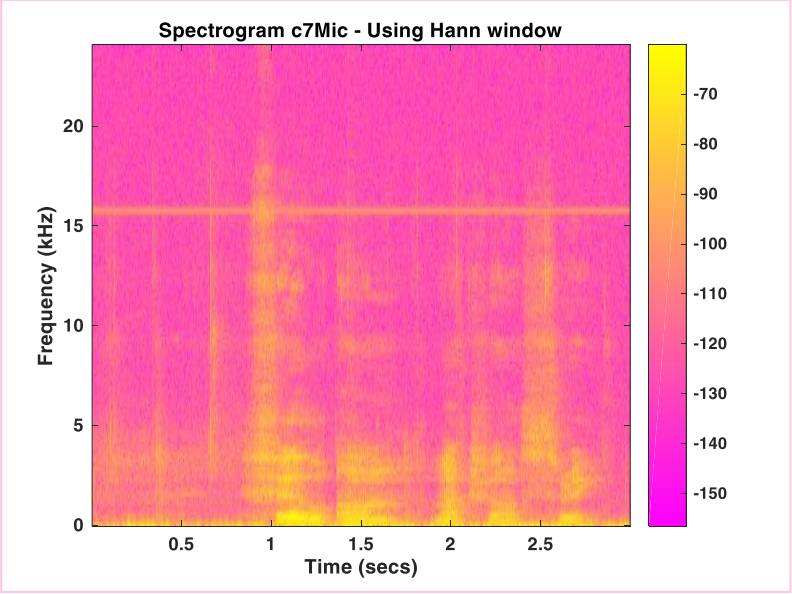
سپس ، با استفاده از تابع spectrogram متلب ، شکل های زیر رسم شده اند.

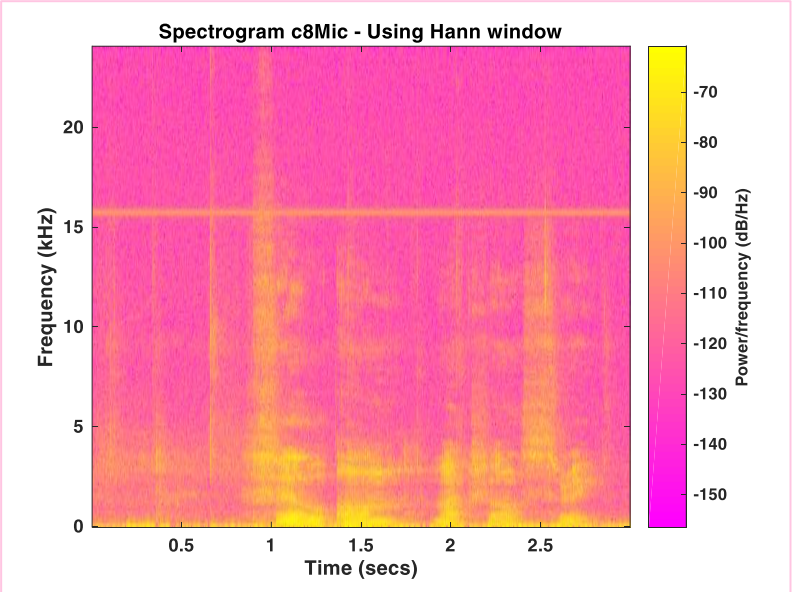




دو شکل زیر به کمک spectrogram و به کمک پنجره گذاری رسم شده است . اندازه ی پنجره کنترلیست بر روی resolutionتصویری که به ما میدهد، هرچه این عدد کوچک تر باشد ، resolution تصویر نیز بالاتر خواهد بود .

(البته باید در نظر داشت مینیمم آن مقدار مشخصی دارد چون ممکن است باعث بهم ریختگی شکل شود .)




Spectrogram c8Mic - Using Hann window
Frequency (kHz)
20
15
10
5
0
0.5
1
1.5
2
2.5
Time (secs)
-70
-80
-90
-100
-110
-120
-130
-140
-150
Power/frequency (dB/Hz)

.2

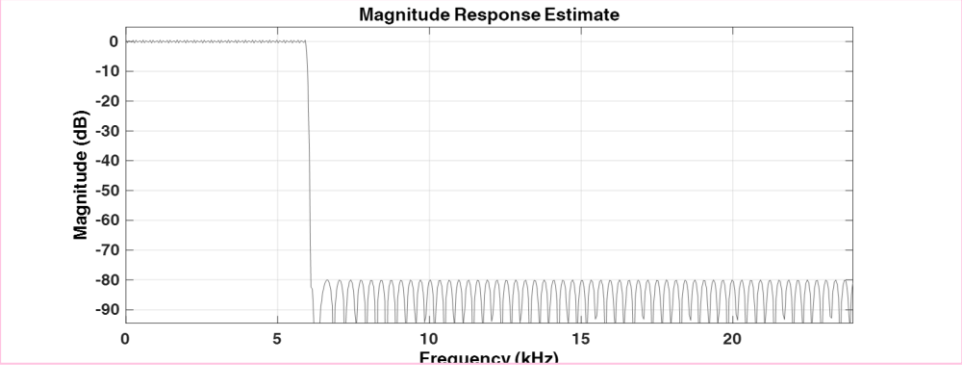
در این بخش ما به کمک فیلتر Filter design ، فیلترینگ خود را انجام میدهیم .

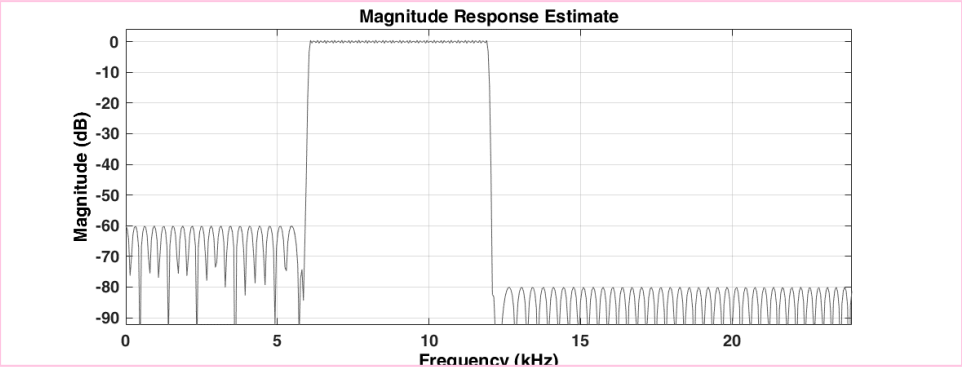
چون لازم است فیلتر بانک ایده آلی داشته باشیم ، مرتبه ی فیلتر ها بالا خواهند بود چون مد نظر ما است که فیلتر ها sharp باشند .

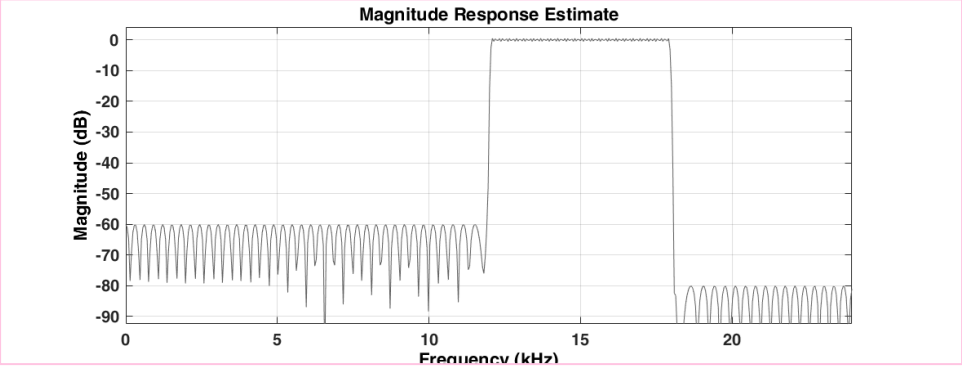
(تفاوت میان Fpass و Fstop در فیلتر های طراحی شده 100Hz است که در مقایسه با 48000 که فرکانس نمونه برداری است مقدار کوچکی است و همینطور باعث افزایش بی رویه مرتبه ی فیلتر نیز نمیشود ، چون نمیتوان به 100% ایده آل بودن رسید .)

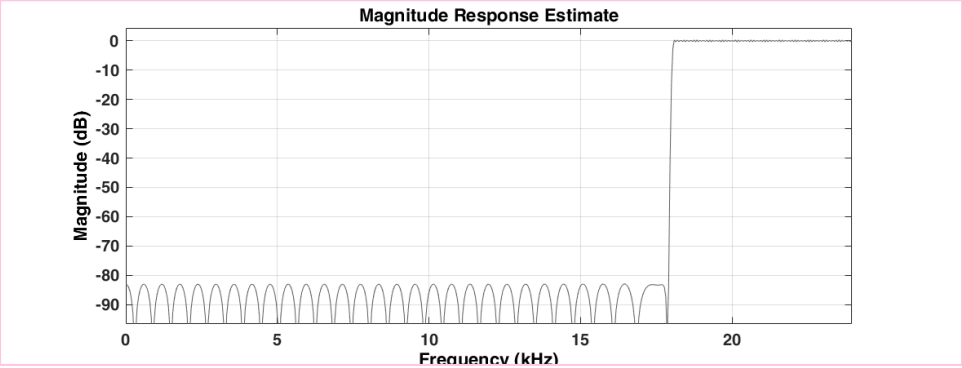
اندازه ی پاسخ فرکانسی فیلتر های طراحی شده را میتوان در 4 شکل زیر مشاهده کرد .

مرتبه ی این 4 فیلتر 868 است .

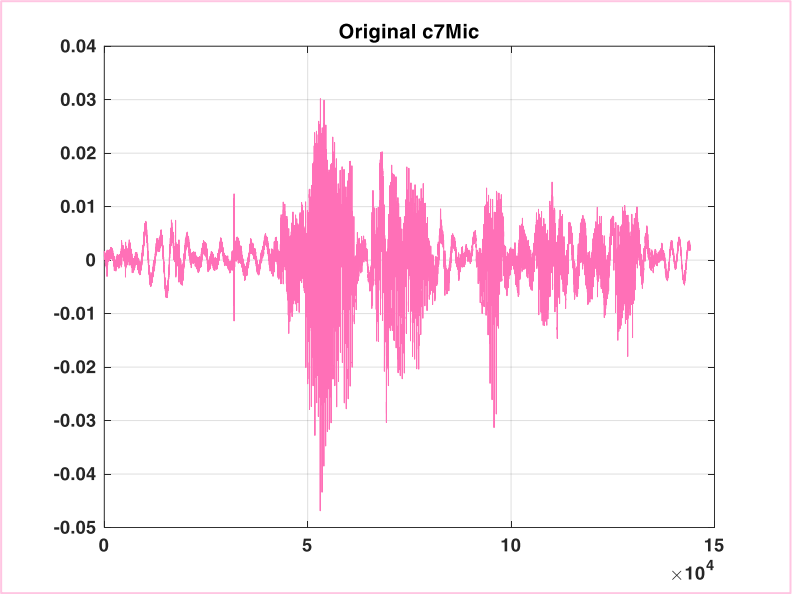




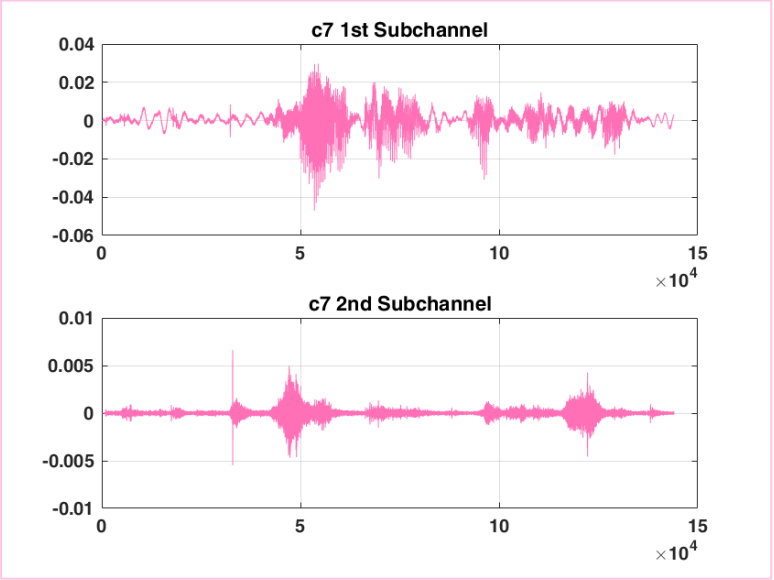


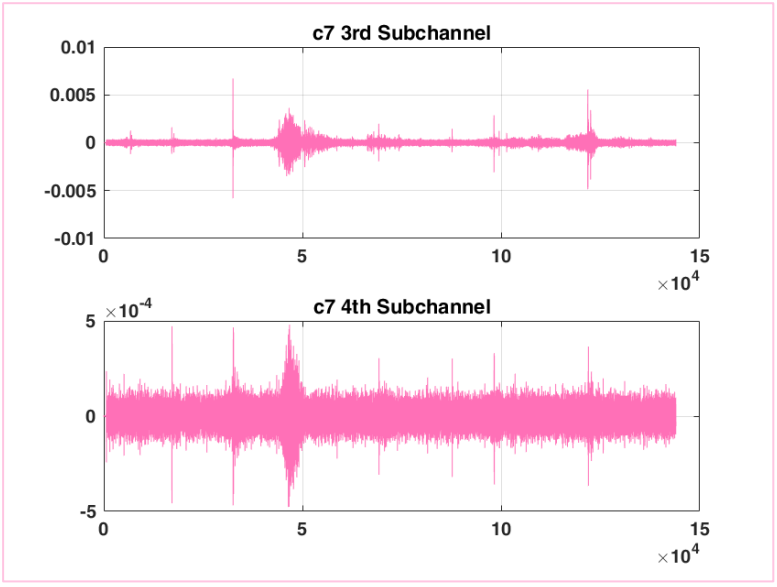


شکل زیر ، شکل اصلی سیگنال c7 قبل از عبور از فیلتر است .

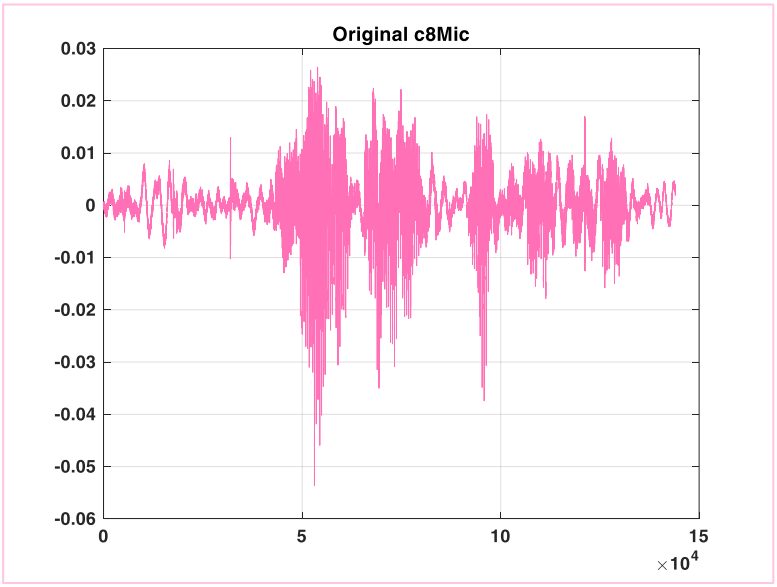


شکل های زیر، شکل های سیگنال c7 بعد از عبور از 4 فیلتر طراحی شده است .

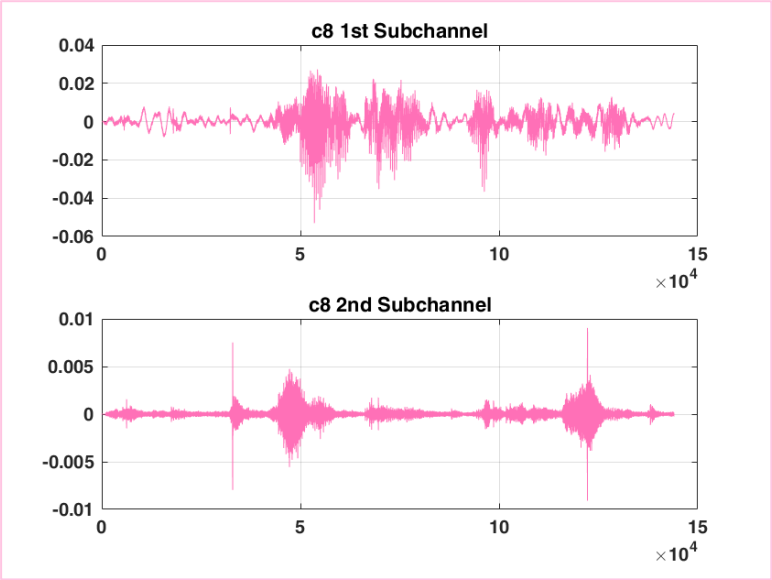


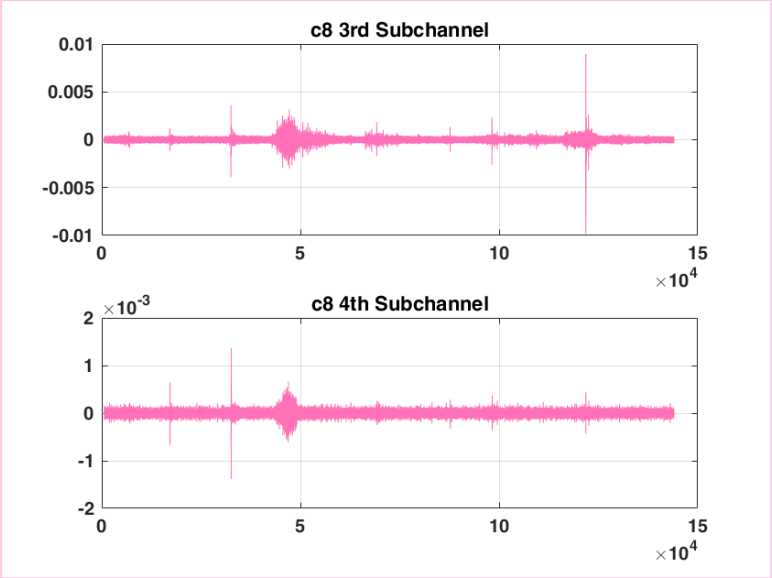


شکل زیر ، شکل اصلی سیگنال c8 قبل از عبور از فیلتر است .

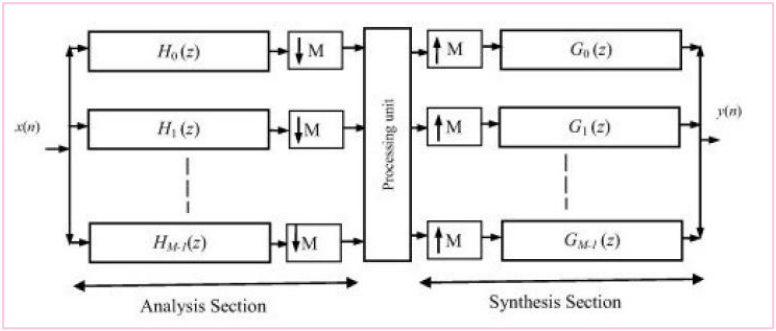


شکل های زیر، شکل های سیگنال c7 بعد از عبور از 4 فیلتر طراحی شده است .





سپس لازم است به کمک decimate، مرحله ی دوم، یعنی down sampling را انجام دهیم، تا سپس، مطابق با شکل زیر ، مرحله ی processing انجام شود .


$H_0(z)$
$H_1(z)$
$H_{M-1}(z)$
M
M
M
Processing unit
M
M
M
$G_0(z)$
$G_1(z)$
$G_{M-1}(z)$
x(n)
y(n)
Analysis Section
Synthesis Section

.3

در این بخش به کمک تابع reshape داده هایی ک داریم را تقسیم بندی میکنیم .

چون عدد 36001 بر 256 بخش پذیر نبود ، تعدادی صفر به اخر آن اضافه میکنیم تا بخش پذیر شود .

ماتریسی ک بدست آمده 256*141 است که همانطور ک در صورت سوال گفته شده ، 141 chunk داریم که طول هریک 256 است .

.4

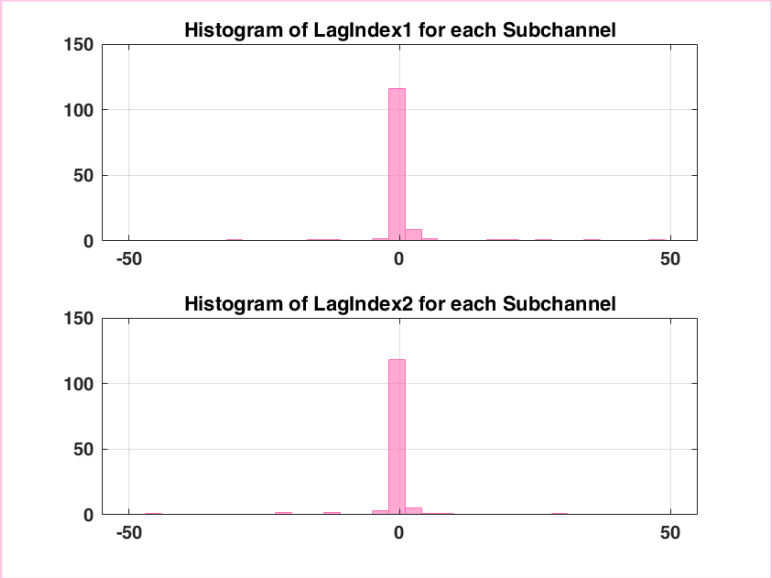
در این قسمت به کمک تابع xcorr متلب ، کرولیشن میان هر یک از chunk های گیرنده ی 7 را با گیرنده ی 8 محاسبه میکنیم و 4 ماتریس lag index را بدست می آوریم .

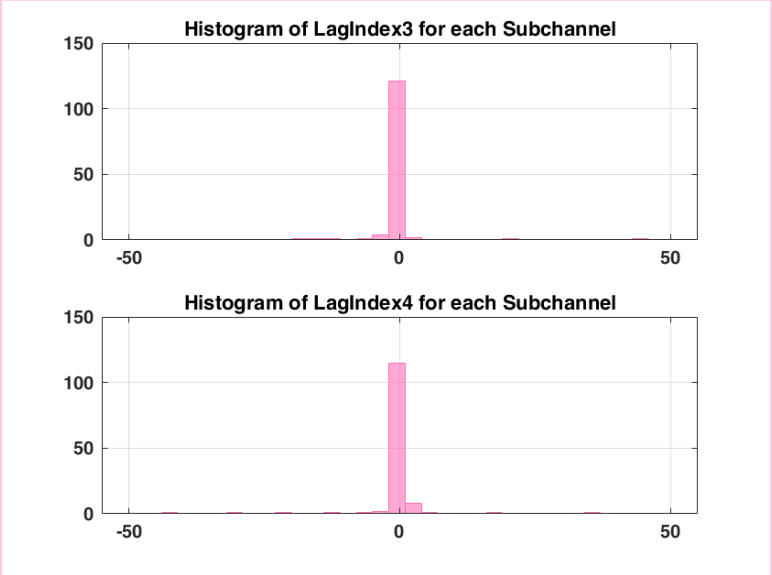
بعد از محاسبه ی xcorr، اندازه ی آن را محاسبه میکنیم ، سپس از آن ماکسیمم میگیریم تا lag index ها محاسبه شوند.

.5

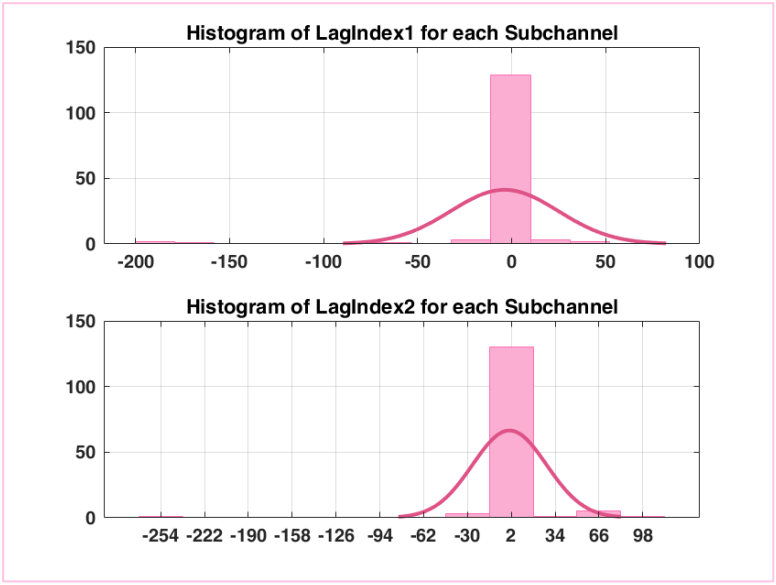
در این بخش به کمک تابع Histogram ، هیستوگرام مربوطه به هر زیرکانال را رسم میکنیم.

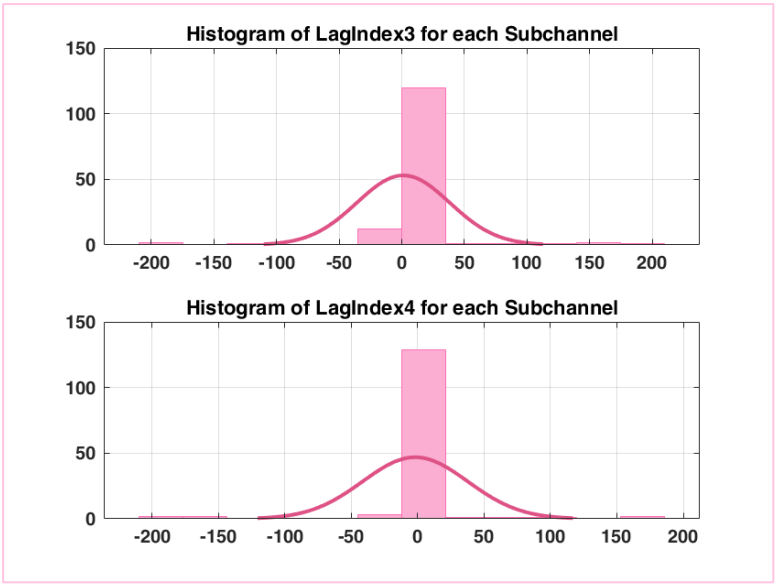
اگر شکل های زیر را با شکل هایی که از فایل main بدست اورده ایم مقایسه کنیم میبینیم اطراف 0 ، دو شکل بیسار مشابه هم هستند . پس بادید از سیگماهایی ک در این قسمت بدست میآوریم در پارت 7 استفاده کنیم .





میتوان یک تابع نرمال به هریک از 4 سپکتوگرام بالا فیت کرد که در مشاهده میکنیم :


Histogram of LagIndex1 for each Subchannel
150
100
50
0
-200
-150
-100
-50
0
50
100
Histogram of LagIndex2 for each Subchannel
150
100
50
0
-254
-222
-190
-158
-126
-94
-62
-30
2
34
66
98


Histogram of LagIndex3 for each Subchannel
150
100
50
0
-200
-150
-100
-50
0
50
100
150
200
Histogram of LagIndex4 for each Subchannel
150
100
50
0
-200
-150
-100
-50
0
50
100
150
200

به کمک histfit میانگین و سیگما های هریک از lag index ها بدست آمده اند .

First sub channel

```
Normal distribution
     mu = -3.78014    [-8.54854, 0.988261]
  sigma =  28.6394    [25.6417, 32.4371]
```

Second sub channel

```
Normal distribution
     mu = 0.680851    [-3.82508, 5.18678]
  sigma =   27.063    [24.2303, 30.6516]
```

Third sub channel

```
Normal distribution
     mu = 1.22695    [-4.95956, 7.41346]
  sigma = 37.1567    [33.2674, 42.0838]
```

Fourth sub channel

```
Normal distribution
     mu = -1.9078    [-8.49754, 4.68193]
  sigma = 39.5785    [35.4357, 44.8268]
```

همانطور ک از شکل های قسمت5 پیداست ، یک نرمال وجود دارد که هدف استخراج ان است که کمترین lag را دارد .

برای محاسبه ی واریانس ، برای بالا بردن دقت ، محدوده ی 20– تا 20 انتخاب شد و مساحت آن محاسبه شد ، همچنین میدانیم اگر به اندازه ی 99.7% از مساحت پیش برویم ، میتوانیم 3σ را محاسبه کنیم .

برای محاسبه ی واریانس ، باید از میانگین به دو طرف نمودار حرکت کرد تا به 99.7% مساحت رسید .

برای میانگین میتوان از داده های بالا بهره برد چون داده های پرت بر روی میانگین تاثیر انچنانی ندارد .

سیگما های بدست آمده از تابع calculate_sigma در زیر آورده شده :

sigma1	0.6667
sigma2	3.3333
sigma3	5.6667
sigma4	0.3333

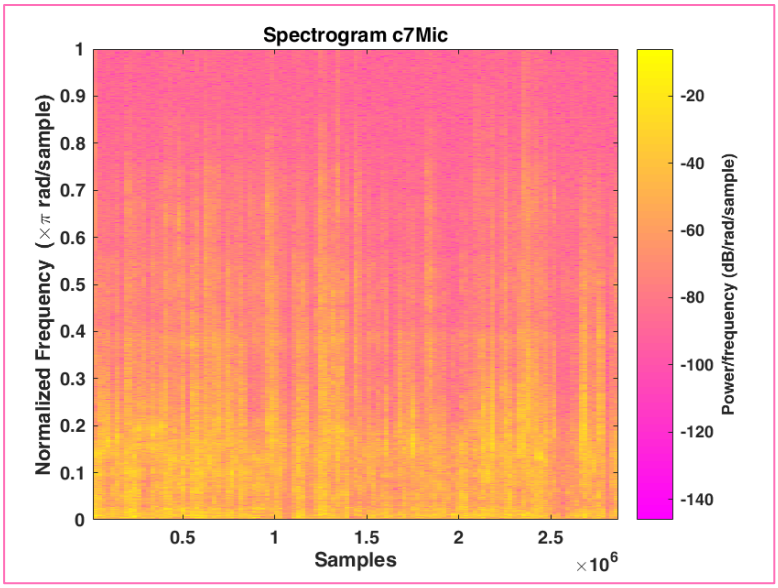
6.

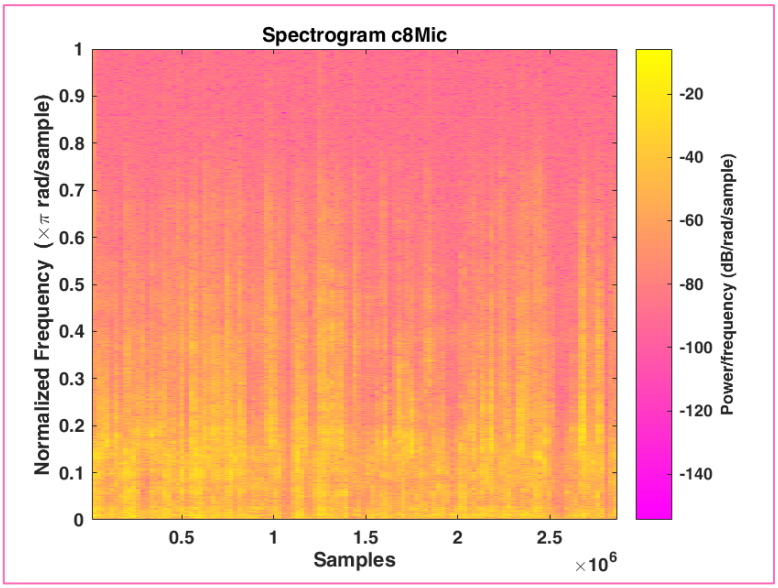
تمام مراحل 1 تا 4 را برای فایل main نیز انجام میدهیم .

دو شکل اول فقط با دادن سیگنال و فرکانس نمونه برداری به spectrogram کشیده شده است .

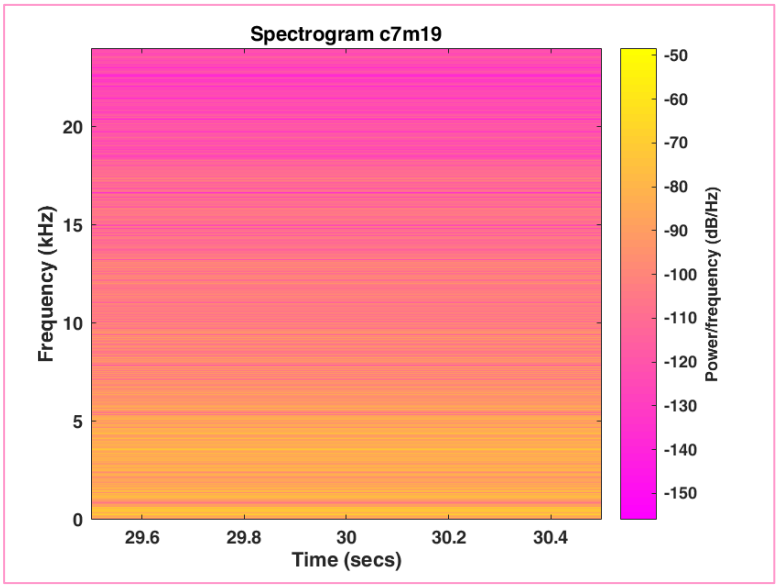
شکل دوم و سوم ورودی های تابع کامل تر شده اند و مقدار overlap نیز داده شده است .

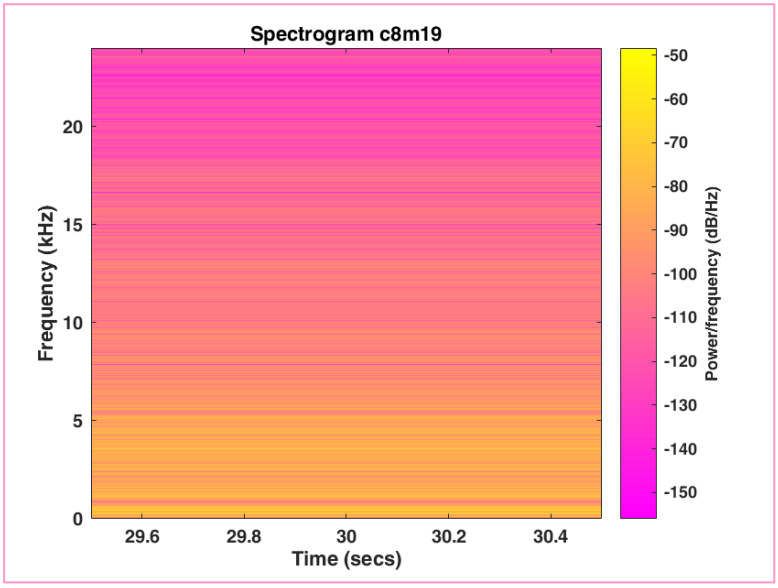
شکل چهارم و پنجم با دادن پنجره ی Hann رسم شده است .

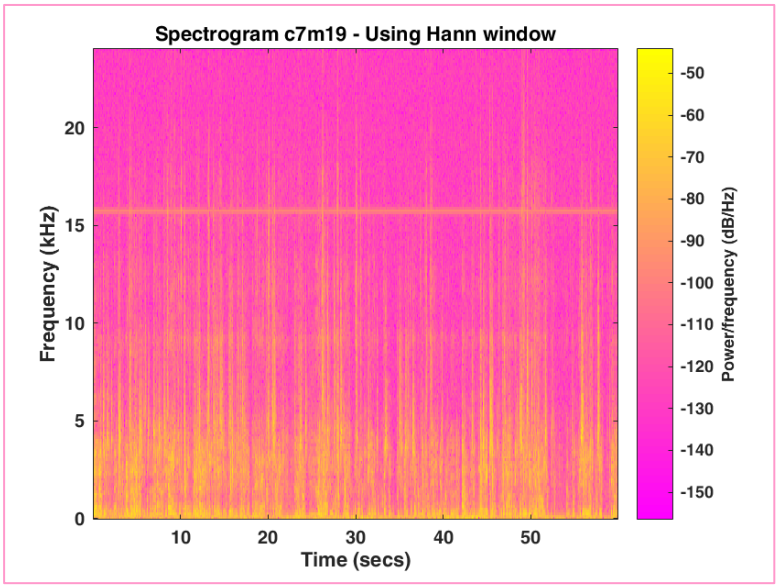


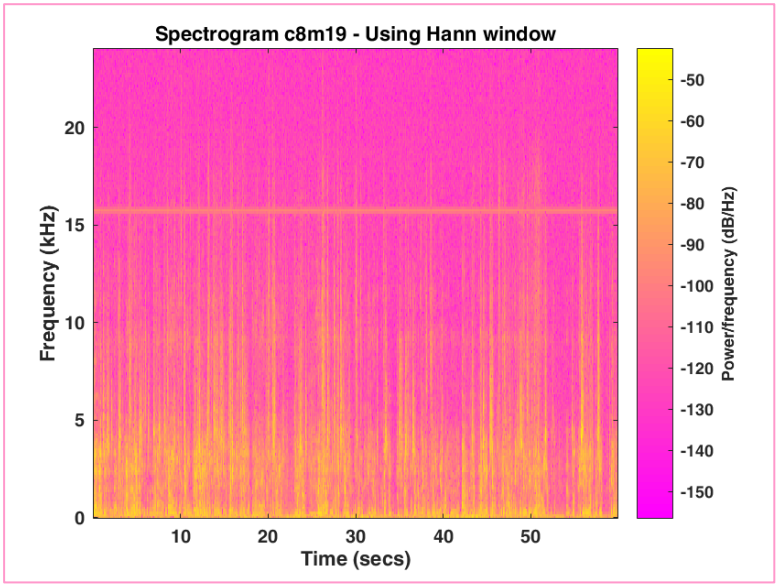

Spectrogram c8Mic
Normalized Frequency (×π rad/sample)
Samples
×10^6
Power/frequency (dB/rad/sample)

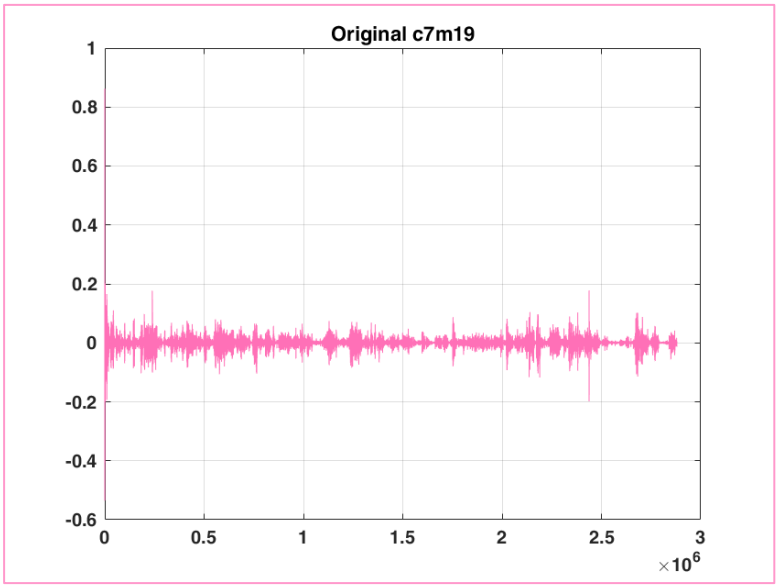
4

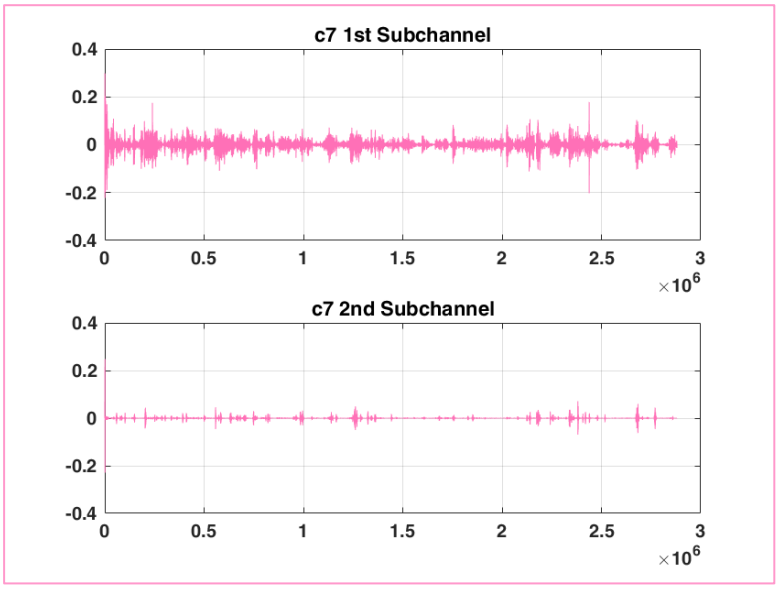

Spectrogram c7m19
Frequency (kHz)
Time (secs)
Power/frequency (dB/Hz)

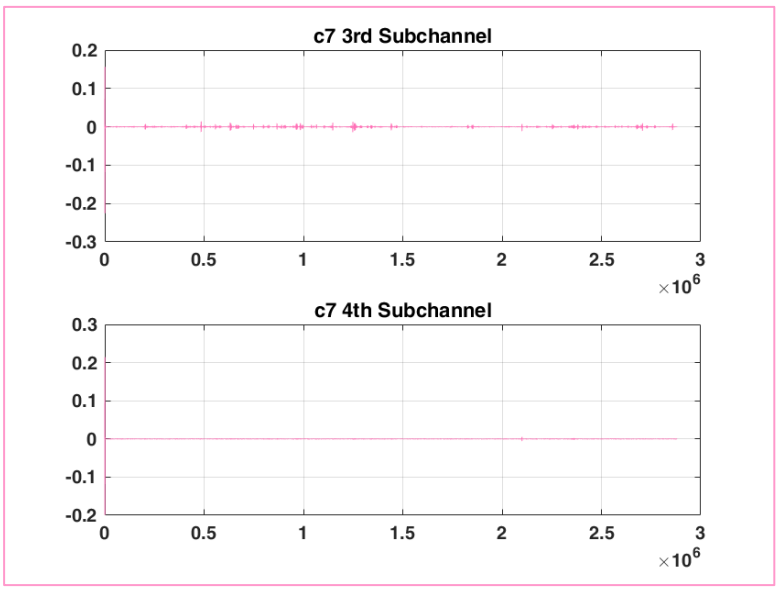

Spectrogram c8m19
Frequency (kHz)
0
5
10
15
20
29.6
29.8
30
30.2
30.4
Time (secs)
-50
-60
-70
-80
-90
-100
-110
-120
-130
-140
-150
Power/frequency (dB/Hz)


Spectrogram c7m19 - Using Hann window
Frequency (kHz)
0
5
10
15
20
10
20
30
40
50
Time (secs)
-50
-60
-70
-80
-90
-100
-110
-120
-130
-140
-150
Power/frequency (dB/Hz)

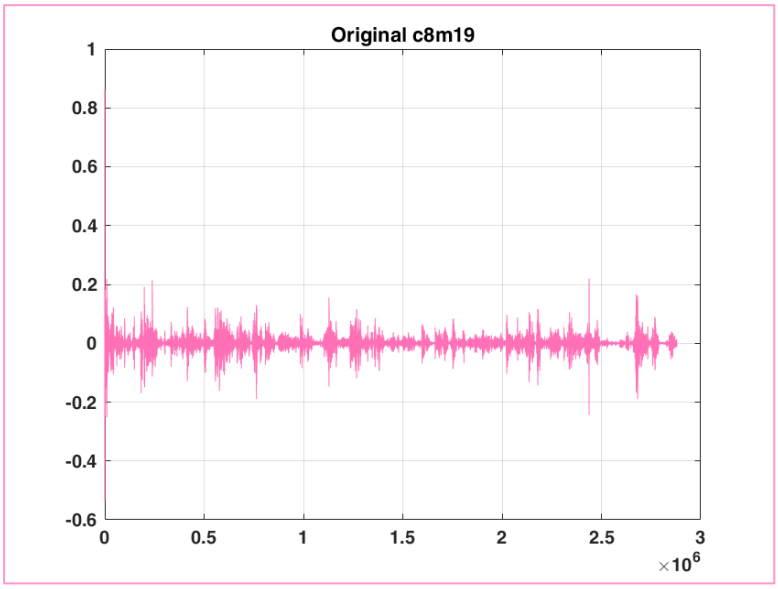

Spectrogram c8m19 - Using Hann window
Frequency (kHz)
Time (secs)
Power/frequency (dB/Hz)
0
5
10
15
20
10
20
30
40
50
-50
-60
-70
-80
-90
-100
-110
-120
-130
-140
-150

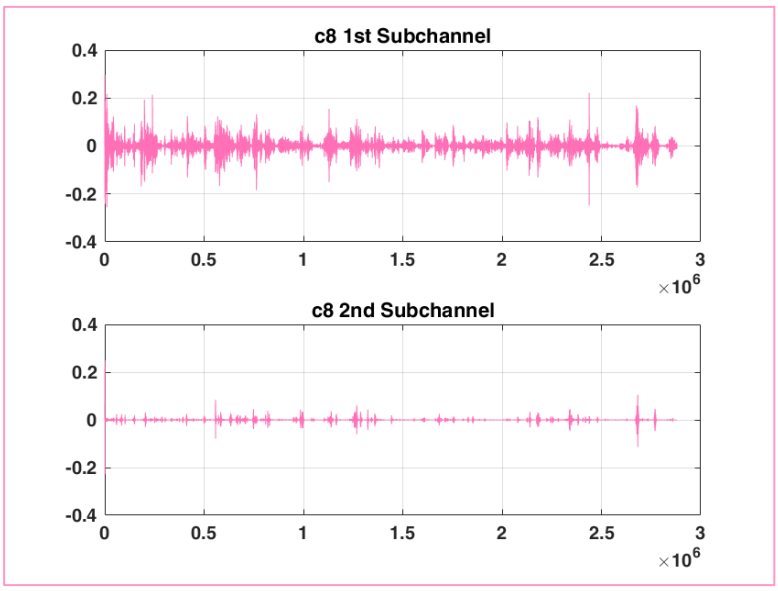

Original c7m19
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
0
0.5
1
1.5
2
2.5
3
×10^6

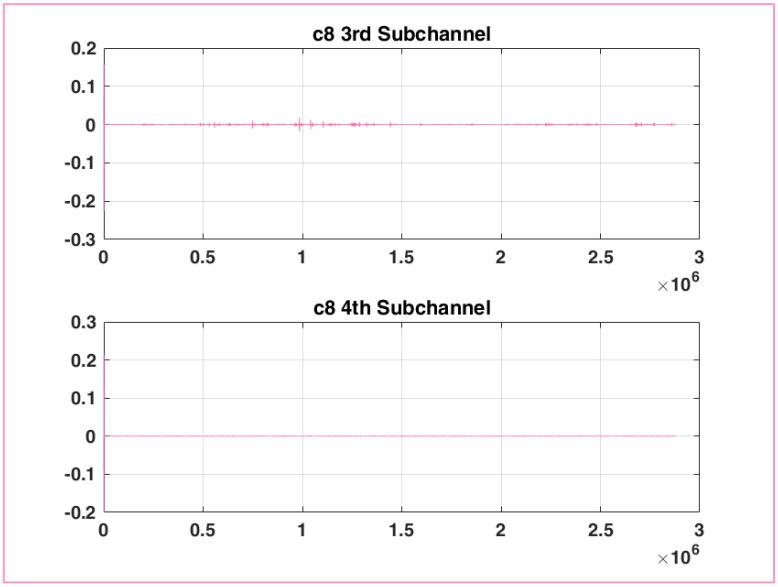




تمام فیلتر ها را به کانال 8 نیز اعمال میکنیم :

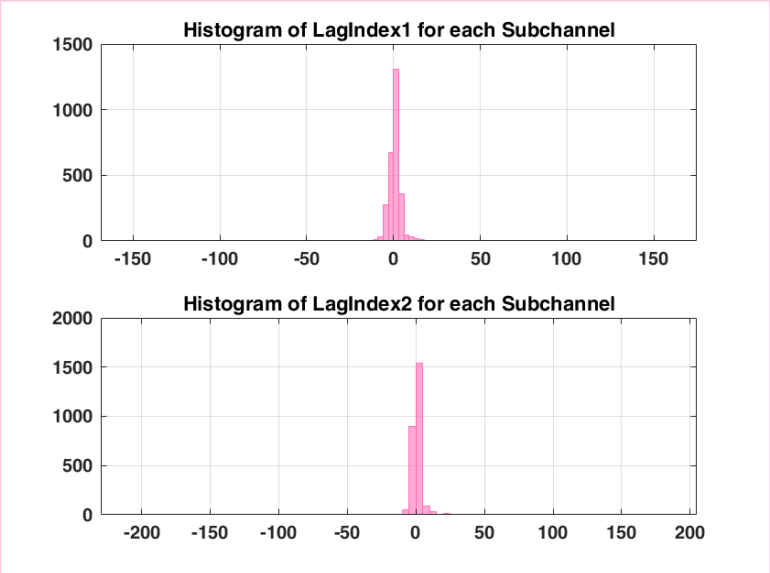

Original c8m19
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
0
0.5
1
1.5
2
2.5
3
×10^6

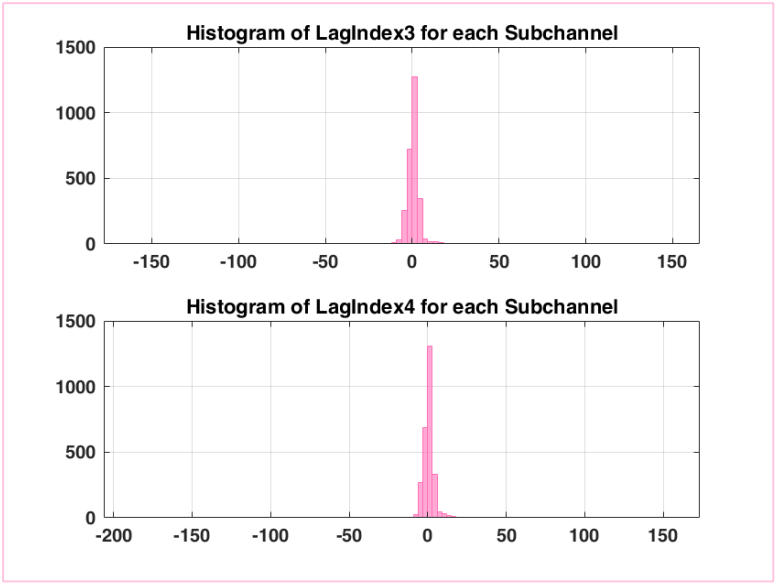

c8 1st Subchannel
0.4
0.2
0
-0.2
-0.4
0
0.5
1
1.5
2
2.5
3
×10^6
c8 2nd Subchannel
0.4
0.2
0
-0.2
-0.4
0
0.5
1
1.5
2
2.5
3
×10^6



4

همانطور که مشخص است شکل های فایل main به نرمال شبیه تر اند .




Histogram of LagIndex3 for each Subchannel
1500
1000
500
0
-150
-100
-50
0
50
100
150
Histogram of LagIndex4 for each Subchannel
1500
1000
500
0
-200
-150
-100
-50
0
50
100
150

7.

با استفاده از سیگما هایی که از بخش intro بدست آمده و میانگین ها، ضرایب را ، به کمک فرمول زیر،بدست میاوریم و در هر chunk ضرب میکنیم تا chunk های جدید بدست بیایند.

$$w_{ch}[n] = \exp^{\frac{-(lag-\mu_{ch})^2}{2\sigma^2_{ch}}}$$

.8

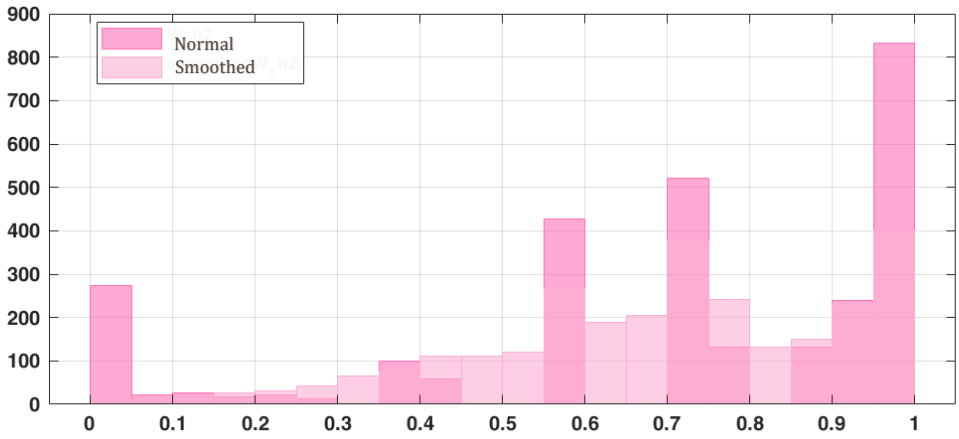
پس از اعمال ضرایب به هریک از سلول ها، چانک های ایجاد شده ی جدید ابتدا reshape شده تا به تک چنل تبدیل شوند ، سپس باهم جمع میشوند تا سیگنال اصلی بازیابی شود .

پس از جمع شدن ، باید عمل up sampling اجرا شود

دو سیگنال با نام های righthand_Signal و lefthand_sigmal در فایل پروژه ذخیره شده اند .

.9

چون صوت های استخراج شده دارای نویز بسیار زیادی است به کمک تابع smooth در متلب که یک تابع با فیلتر پایین گذر است ، اندکی از نویز سیگنال رفع میشود . چون غیرپیوستگی بین ضرایب اندکی از بین میرود و ضرایب نرم تر میشوند .



10.

پس از اعمال ضرایب به هریک از سلول ها، چانک های ایجاد شده ی جدید ابتدا reshape شده تا به تک چنل تبدیل شوند ، سپس باهم جمع میشوند تا سیگنال اصلی بازیابی شود .

دو سیگنال با نام های smoothed_righthand_Signal و smoothed_lefthand_sigmal در فایل پروژه ذخیره شده اند .