Project Phase 1

Ali Ghavampour

97102293

Farhad Fallah

97102214

Sharif University of Technology

Signal and Systems

Dr. Hamid K. Aghajan

تبدیل فوریه زمان گسسته - DTFT

- **نشان دهید تبدیل فوریه گسسته زمان متناوب است و دوره تناوب آن را بیان کنید.**

$$X(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega n} \longrightarrow Note\ that: e^{j(\Omega+2\pi)} = e^{j\Omega}e^{j2\pi} = e^{j\Omega}$$

$$So: X(e^{j(\Omega+2\pi)}) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega n}e^{-jn2\pi} = \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega n} = X(e^{j\Omega})$$

در نتیجه تبدیل فوریه گسسته زمان با دوره تناوب 2π متناوب است. همچنین در فضای گسسته زمان، برای فرکانس حد بالا و پایین داریم که برابر با π برای حد بالا و $-\pi$ برای حد پایین است.

- **نشان دهید برای سیگنال های حقیقی، تنها داشتن بازه‌ی $[-\pi, \pi]$ کافی است و می‌توان سیگنال اولیه را از روی آن بازسازی کرد.**

با توجه به خواص تبدیل فوریه گسسته زمان، اگر $x[n]$ حقیقی باشد، اندازه تبدیل فوریه آن زوج و فاز آن فرد است.

$$x^*[n] = x[n] \Longrightarrow X(e^{j\Omega}) = X^*(e^{-j\Omega}) \Longrightarrow \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega n} = \sum_{n=-\infty}^{\infty} x[n]e^{-j(-\Omega)n}$$

$$\Longrightarrow |X(\Omega)| = |X(-\Omega)| \text{ and } \Phi(X(\Omega)) = -\Phi(X(-\Omega))$$

در نتیجه با داشتن محتوای تابع فرکانسی در دامنه 0 تا π می‌توان محتوا در تمام فرکانس ها را داشت. پس با داشتن این میزان و رابطه سنتز می‌توان سیگنال $x[n]$ را به دست آورد.

تبدیل فوریه گسسته - DFT

- سیگنال پیوسته و متناوب $\widetilde{x}(t)$ با دوره تناوب T را در نظر بگیرید. سیگنال $x(t)$ را یک دوره تناوب از سیگنال اولیه در نظر بگیرید؛ یعنی:

$$x(t) = \begin{cases} \widetilde{x}(t) & |t| < T/2 \\ 0 & O.W \end{cases}$$

اگر c_kها ضرایب سری فوریه مختلط سیگنال $\widetilde{x}(t)$ باشند، مقدار c_k را برحسب تبدیل فوریه $x(t)$ بیابید.

ابتدا ضرایب سری فوریه را محاسبه می‌کنیم:

$$c_k = \frac{1}{T}\int_{-T/2}^{T/2} \tilde{x}(t)e^{-jk\omega_0 t}dt$$

حال تبدیل فوریه x(t) به صورت زیر را در نظر می‌گیریم:

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt$$

به توجه به تعریف صورت سوال در بازه T/2- تا T/2 برای فرمول ضرایب سری فوریه داریم:

$$c_k = \frac{1}{T}\int_{-T/2}^{T/2} \tilde{x}(t)e^{-jk\omega_0 t}dt = \frac{1}{T}\int_{-\infty}^{\infty} x(t)e^{-jk\omega_0 t}dt \Longrightarrow c_k = \frac{1}{T}X(k\omega_0)$$

- **شرط لازم و کافی برای بازسازی x[n] از روی X[k] را بیان کنید.**

$$X(\Omega) = \sum_{n=-\infty}^{\infty} x[n]e^{-j\Omega n} = \sum_{n=0}^{N_1-1} x[n]e^{-j\Omega n}$$

با توجه با این رابطه برای X(Ω) اگر ما تعداد N1 نقطه از آن را داشته باشیم، می‌توانیم با جایگذاری در X(Ω) و حل N1 معادله، مقدار N1 مجهولی که داریم، یعنی x[0] تا x[N1-1] را به دست بیاوریم. البته در حالت کلی نیاز نیست سیگنال از 0 تا N1-1 باشد و در اینجا صرفا برای سادگی و چون فرقی نمی‌کند به این شکل بیان شده است. (مثلا سیگنال می‌تواند از 10 تا N1+9 باشد)

در نتیجه شرط لازم و کافی برای وجود امکان بازسازی سیگنال برابر است با:

$$N \geq N1$$

- **بررسی صحت مراحل ذکر شده:**

برای تبدیل فوریه گسسته داریم:

$$X[k] = X(\frac{2k\pi}{N}) = \sum_{n=0}^{N-1} x[n]e^{-j\frac{2k\pi}{N}n}$$

همچنین اگر سیگنال x[n] را با دوره تناوب N بسط تناوبی(periodic extension) بدهیم، ضرایب سری فوریه آن به صورت زیر است:

$$a_k = \frac{1}{N}\sum_{n=0}^{N-1} x[n]e^{-j\frac{2k\pi}{N}n}$$

مشاهده می‌شود که دو عبارت نوشته شده در صفحه قبل به جز در یک ضریب بسیار شبیه به هم هستند. یعنی عملا:

$$a_k = \frac{1}{N} X[k]$$

در نتیجه اگر قدم به قدم با مراحل پیش برویم، در مرحله اول به سیگنال گسسته نمونه برداری شده از $X(\Omega)$ به چشم ضرایب سری فوریه یک سیگنال نگاه می‌کنیم. در مرحله دوم با رابطه سنتز سری فوریه، سیگنال متناوب شده $x[n]$ را به دست می‌آوریم. اکنون اگر یک تناوب N از سیگنال را انتخاب کنیم، به سیگنال اولیه $x[n]$ می‌رسیم.

- **با جست و جو در اینترنت تعریف فرمال و دقیق رابطه تبدیل فوریه گسسته N نقطه ای و عکس آن را بدست آورید.**

Discrete Fourier Transform:

$$X[k] = \sum_{n=0}^{n=N-1} x[n] e^{-j\frac{2\pi}{N}kn}$$

Inverse Transform:

$$x[n] = \frac{1}{N} \sum_{n=0}^{N-1} X[k] e^{j\frac{2\pi}{N}kn}$$

- **با جست وجو در اینترنت رابطه پارسوال برای تبدیل فوریه گسسته را بیان کرده و آن را اثبات کنید.**

Parseval's Theorem

$$\sum_{n=0}^{N-1} |x[n]|^2 = \frac{1}{N} \sum_{k=0}^{N-1} |X[k]|^2$$

اثبات رابطه:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N}kn} \Rightarrow |X[k]|^2 = X[k].X^*[k] = \sum_{n=0}^{N-1} x[n] \sum_{n'=0}^{N-1} x^*[n'] e^{-j\frac{2\pi}{N}k(n-n')}$$

$$\rightarrow \sum_{k=0}^{N-1} |X[k]|^2 = \sum_{k=0}^{N-1} \sum_{n=0}^{N-1} x[n] \sum_{n'=0}^{N-1} x^*[n'] e^{-j\frac{2\pi}{N}k(n-n')} = \sum_{n=0}^{N-1} x[n] \sum_{n'=0}^{N-1} x^*[n'] \sum_{k=0}^{N-1} e^{-j\frac{2\pi}{N}k(n-n')}$$

حال برای سیگمای آخر داریم:

$$\sum_{k=0}^{N-1} e^{-j\frac{2\pi}{N}k(n-n')} = \frac{e^{-j2\pi(n-n')} - 1}{e^{-j\frac{2\pi}{N}(n-n')} - 1}$$

این سیگما تنها در صورت n=n’ برابر با صفر نیست در نتیجه داریم:

$$\sum_{k=0}^{N-1} e^{-j\frac{2\pi}{N}k(n-n')} = N\delta[n-n']$$

پس قضیه به صورت زیر ثابت می‌شود:

$$\sum_{k=0}^{N-1} |X[k]|^2 = N\sum_{n=0}^{N-1} |x[n]|^2$$

- **اندازه تبدیل فوریه یک سیگنال حقیقی ماهیت زوج دارد یا فرد؟ مشابه همین سوال را برای فاز تبدیل نیز پاسخ دهید.**

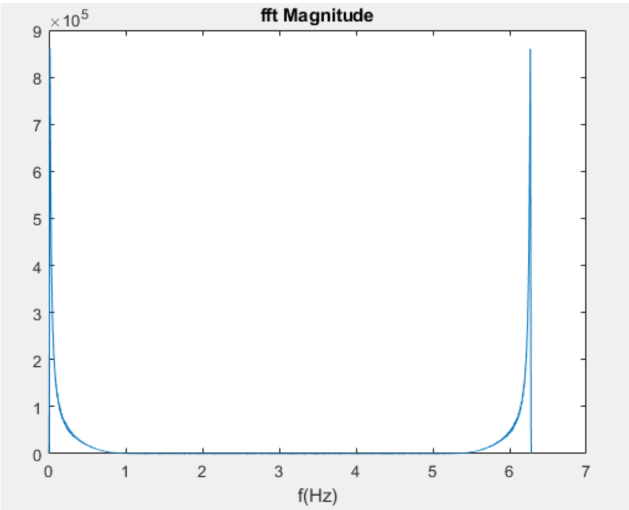
همانطور که در یکی از قسمت های قبلی نیز دیدیم، اندازه تبدیل فوریه ماهیت زوج و فاز آن ماهیت فرد دارد.

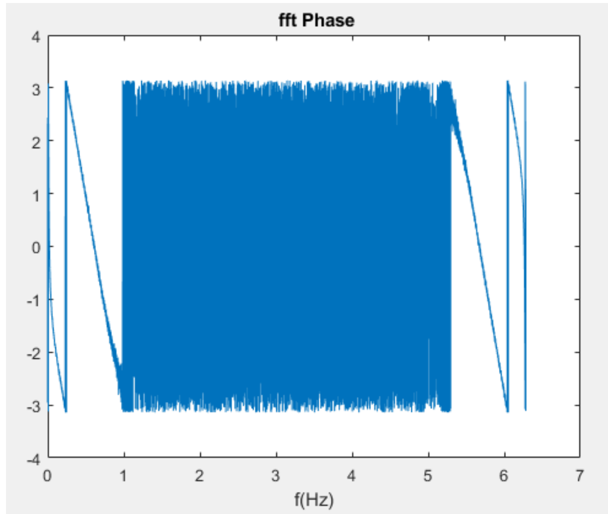
نکته: در این پروژه سوال های تئوری به صورت مستقیم از صورت پروژه آورده شده اند و به محض شروع قسمت عملی، دیگر اصل سوال های آورده نمی‌شود و با رنگ قرمز شماره گذاری می‌شوند. مثلا به صورت سوال اول، سوال دوم و غیره. همچنین در کد متلب نیز تناظر یک به یکی بین شماره سوال و کامنت های کد وجود دارد. یعنی کد سوال اول، با کامنت به صورت Question01 در کد مشخص شده است. این قرارداد برای راحتی تناسب دادن قسمت های مختلف گزارش به کد می‌باشد. (دقت کنید که ممکن است یکی از سوال ها کد متناظر نداشته باشد و فقط تئوری باشد)

توجه: **برای ران شدن درست کد ها حتما فولدر data که با پروژ آپلود شده است را در path متلب باز کنید.**

- **بررسی فایل y.mat :**

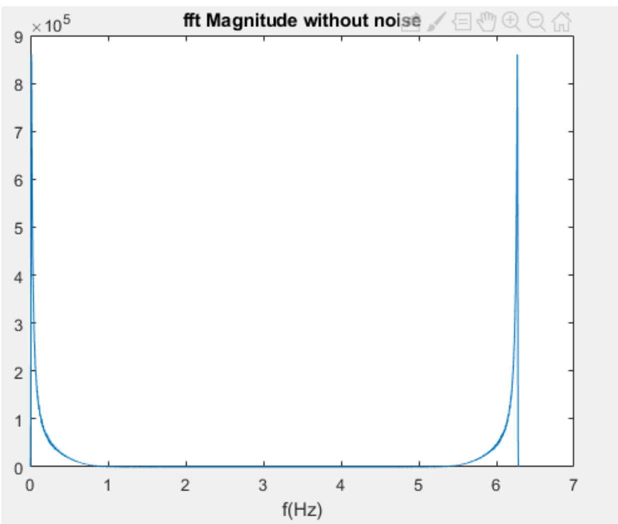
ابتدا از ماتریس داده شده fft گرفتیم و نمودار فاز و اندازه آن را رسم کردیم.

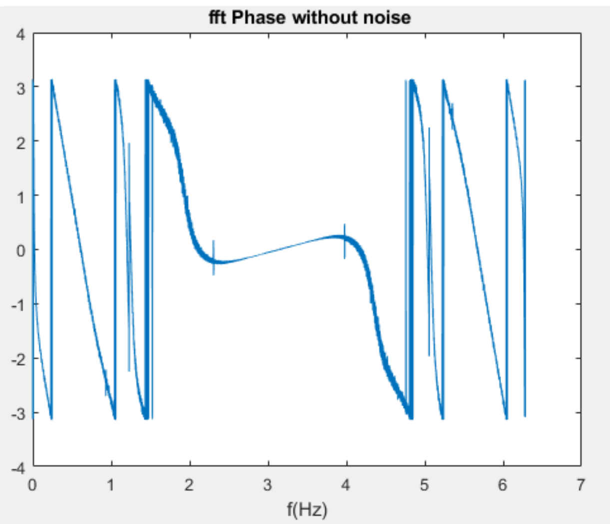




و با نوشتن کدی تقارن های زوج و فرد را برای دامنه و فاز بررسی کردیم و مشاهده که شد که هیچ یک از این تقارن ها را دارا نیستند.

سپس برای حذف نویز ها از داده های اولیه Real گیری کردیم تا نویز موهومی موجود در سیگنال حذف شود و مجدد نمودار اندازه و فاز را رسم کردیم.





اینبار مشاهده شد که اندازه دارای تقارن زوج و فاز دارای تقارن فرد است.

سوال اول و دوم مربوط به این بخش هستند.

نمونه برداری - Sampling

سوال اول)

ابتدا به چند نکته درباره خود کد بپردازیم و سپس سراغ تست آن با یک تابع که خروجی آن را می‌دانیم می‌رویم.

در کد ابتدا با رویکرد اینکه ورودی f برای ما محدودیتی ایجاد نمی‌کند، مقادیر دامنه و فرکانس ها محاسبه می‌شود و سپس با یک شرط if به بررسی ایجاد محدودیت f پرداخته می‌شود اینگونه که اگر از pi کوچکتر باشد، نسبت آن به pi معیاری از تعداد نقاطی است که از فیلتر عبور می‌کند در نتیجه یک مقدار Cut Off برای دیتا های عبوری از فیلتر تعریف می‌شود.

نکته دیگری راجب کد مقیاس کردن فرکانس گسسته روی فرکانس پیوسته می‌باشد که طبق عبارت ذکر شده برای رابطه فرکانس پیوسته و گسسته در صورت گزارش داریم:

$$\omega = \Omega F_s \Rightarrow 2\pi f = \Omega F_s \Rightarrow f = \frac{F_s}{2\pi}\Omega$$

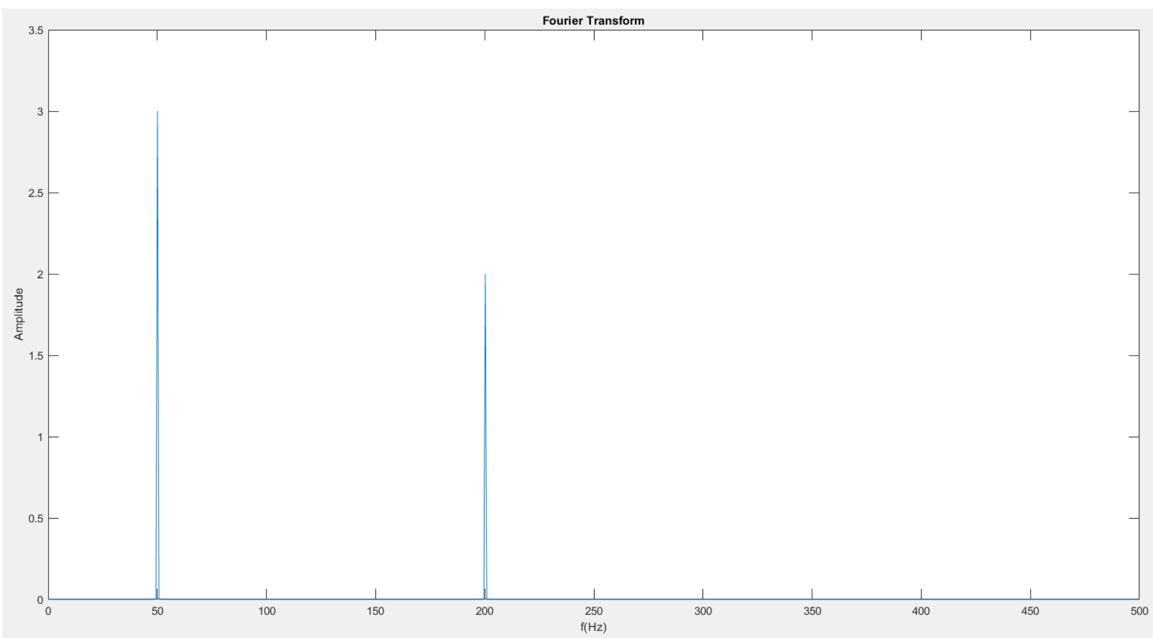
که در صورت کد نیز بردار محاسبه شده برای فرکانس گسسته در این ضریب که در بالا به دست آمده ضرب شده است.

بررسی صحت عملکرد کد:

ابتدا با مقادیر Fs=1000Hz و f=pi سیگنال زیر را نمونه برداری می‌کنیم و به فیلتر می‌دهیم و خروجی را بررسی می‌کنیم:

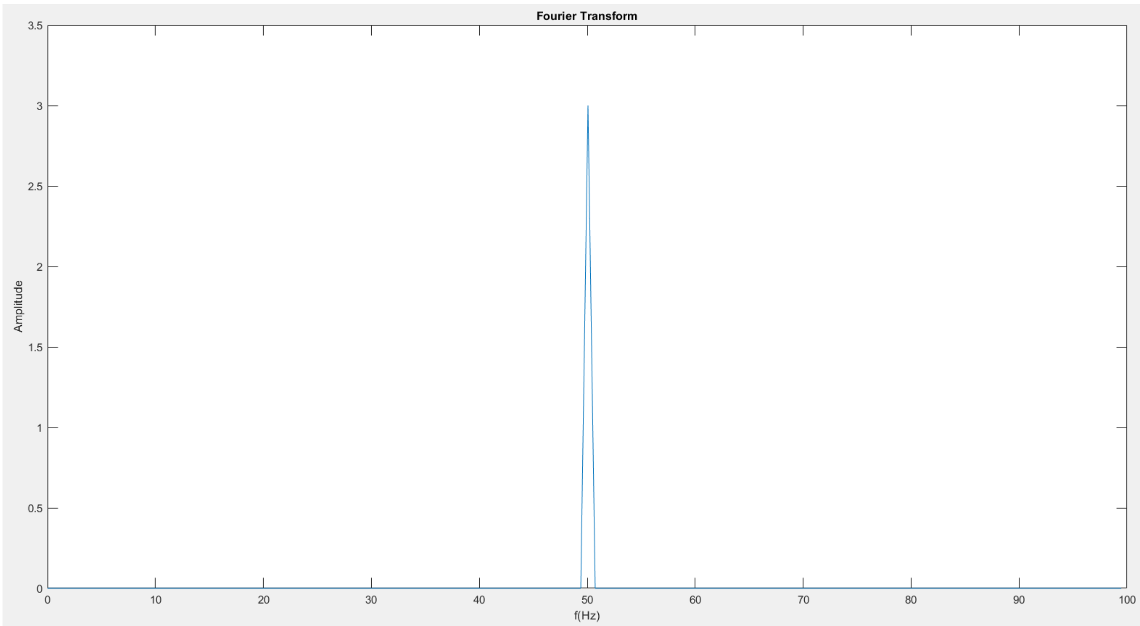
```
x = 2*cos(2*pi*200*t) + 3*cos(2*pi*50*t);
```

خروجی فیلتر:



در تصویر صفحه قبل به وضوح مشخص است که هارمونی های 200 و 50 هرتز در فیلتر خروجی با دامنه های مربوط به آنها در خروجی قرار گرفته اند که صحت این عملکرد کد را تایید می‌کند.

اکنون با تغییر مقدار f به $\frac{2}{10}\pi$ عملکرد فیلتر فرکانسی تابع زده شده را بررسی می‌کنیم:



مشاهده می‌کنیم که خروجی فیلتر، فرکانس 200 هرتز را که در Fs=1000 برابر با $\frac{4}{10}\pi$ می‌باشد را به درستی فیلتر کرده است.

سوال دوم)

اگر این مثلث های ناقص بخواهند با یکدیگر تداخل داشته باشند، باید سیگنال اولیه در حوزه گسسته حاوی فرکانس های بیشتر از pi باشد. در نتیجه مرز عدم تداخل برای این دو مثلث، در نقطه pi می‌باشد. در نتیجه طبق رابطه ای که در سوال یک برای ارتباط فرکانس گسسته و پیوسته بیان شد داریم:

$$f = \frac{F_S}{2\pi}\Omega \rightarrow \Omega = \pi \ : \ f = \frac{F_s}{2}$$

با توجه به رابطه نوشته شده، تداخل مثلث ها زمانی رخ می‌دهد که فرکانس نمونه برداری آنقدر کم باشد که در سیگنال اولیه، یک هارمونی با نصف فرکانس نمونه برداری وجود داشته باشد. پس شرط عدم تداخل عبارت است از:

$$2f_m < F_s$$

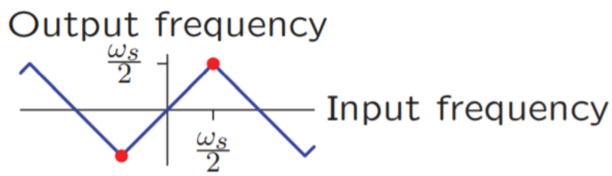
سوال سوم)

محاسبه پهنای باند سیگنال به شرط Fs=12Hz:

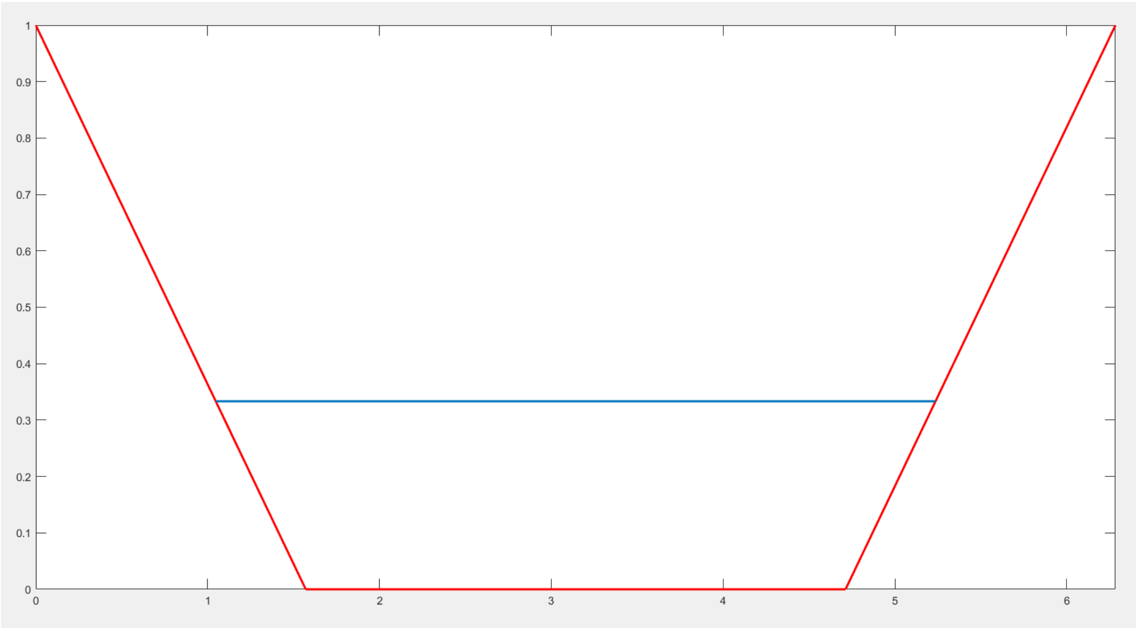
$$f = \frac{F_s}{2\pi}\Omega \Rightarrow \text{ f spectrum is } (0,3)[Hz]$$

با توجه به اینکه در سیگنال اولیه فرکانس 2 هرتز به بالا وجود دارد، اگر بخواهیم با فرکانس 4 هرتز از آن نمونه برداری کنیم قطعا به مشکل می‌خوریم و پدیده aliasing رخ می‌دهد که به بررسی آن در متلب می‌پردازیم.

شکل پدیده aliasing به صورت زیر است:



با توجه به این شکل، با فرکانس نمونه برداری 4، مقدار فرکانس خروجی از 4pi در حوزه پیوسته و pi/3 در حوزه گسسته فرا تر نمی‌رود. در نتیجه شکل زیر را خواهیم داشت:



مشاهده می‌شود که مطابق پدیده aliasing فرکانس ها از مقداری به بعد cut off می‌شوند.

آشنایی با سیگنال های EEG

- **با جست و جو در اینترنت فرق داده گیری invasive و noninvasive سیگنال های EEG را بیابید. هر کدام از روش های فوق در چه حالاتی مورد استفاده قرار می‌گیرند؟**

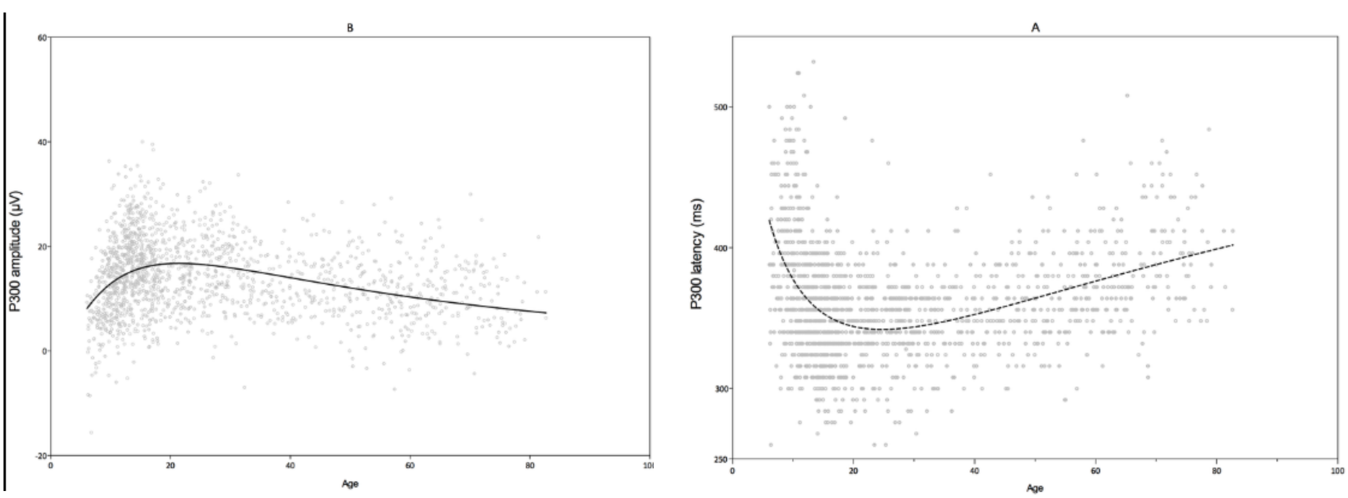
داده گیری invasive اینگونه است که الکترود ها با جراحی روی سطح مغز یا درون آن قرار می‌گیرند و برای کاربرد های پزشکی مثلا پیدا کردن محل خیزش حمله صرع در مغز است. البته در برخی مقاله به روش های invasive ای اشاره شده که الکترود ها به میزان کمی در پوست سر نفوذ می‌کنند. یعنی آنقدر هم invasive نیستند اما به هر حال به خاطر نفوذ به پوست سر اینگونه نام گذاری شده اند.

داده گیری غیر invasive دیتا گیری از مغز از روی پوست و بدون نفوذ به آن شامل روش های خازنی که در آن الکترود از پوست سر فاصله دارد و روش های مختلف تماسی می‌باشد. در حال حاضر مورد قبول ترین نوع آن از نظر دقت نوع الکترود wet می‌باشد که با استفاده از ژل مخصوص امپدانس بین الکترود و پوست سر در بازه مناسبی برای دریافت سیگنال قرار می‌گیرد. این روش حتی کاربرد کلینیکی و درمانی نیز دارد. همچنین دانشمندان زیادی در حال کار روی dry electrodes از نوع های مختلف می‌باشند که این نوع الکترود ها نیز در مقالات و تحقیقات دقت بسیار خوبی نشان داده اند، در عین حال که پایداری و راحتی استفاده از آنها بیشتر است.

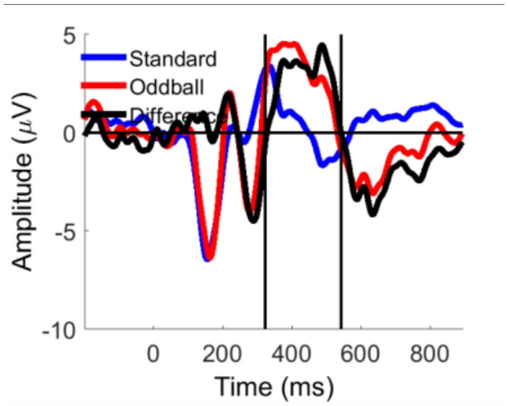
- **با جست و جو در اینترنت درباره مشخصه P300، P600 و N100 که نمونه هایی از مشخصه‌های ERP هستند، اطلاعات کسب کنید و به صورت خلاصه در گزارش ذکر کنید.**

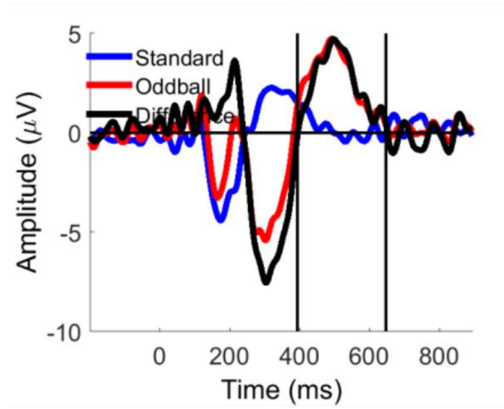
مشخصه P300:

مشخصه P300 مربوط به عملکرد مغز در پردازش های Decision making می‌باشد. این سیگنال وابسته به فیزیک تحریک بیرونی نیست و یک تصمیم و جواب درونی مغز به یک تحریک می‌باشد. این مشخصه به صورت یک افزایش ولتاژ در سیگنال مغزی ظاهر می‌شود و معمولا برای ثبت آن از الکترود ها در ناحیه parietal lobe استفاده می‌شود چون این سیگنال در این ناحیه قوی تر است. همچنین این سیگنال با یک تاخیر اعمال می‌شود که این تاخیر در سنین مختلف، متفاوت است. برای بررسی این مشخصه از تسک های oddball استفاده می‌شود. نمودار های زیر نشان دهنده تغییر تاخیر و دامنه این سیگنال در سنین مختلف است:



نمونه های از ثبت این مشخصه در تسک های oddball:





مشخصه P600:

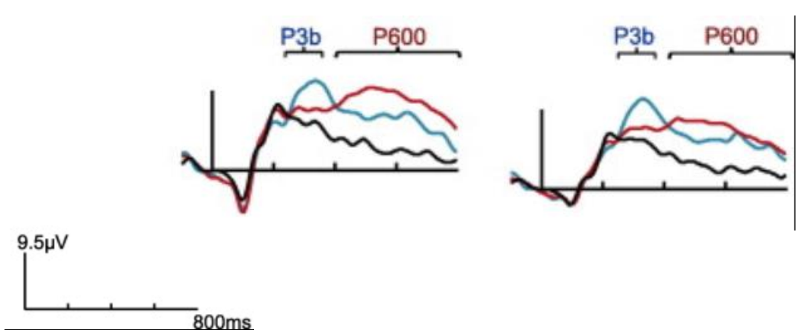
مشخصه P600 افزایش ولتاژ سیگنال EEG در اثر تحریک های اغلب زبانی و لینگویستیک مثلا ایراد های نگارشی یا جمله های نسبتا پیچیده می‌باشد. این تحریک ها هم می‌توانند به صورت شنیداری و هم به صورت دیداری اعمال شوند. همچنین در پژوهش هایی ایراد در نت های موسیقی نیز باعث دیده شدن این پدیده می‌شود پس این پدیده منحصر به تحریک های زبانی و لینگویستیک نیست. این مشخصه در الکترود های مرکزی parietal به خوبی دیده می‌شود همچنین در برخی پژوهش ها دریافت این مشخصه از Frontal lobe نیز نشان داده شده است. مشخصه P600 حدود 500 میلی‌ثانیه بعد از تحریک خودش را نشان می‌دهد و در حدود 600 میلی‌ثانیه به پیک ولتاژ خود می‌رسد که از روی همین زمان نام گذاری شده است.

مثالی از یک جمله که به عنوان تحریکی برای این پدیده است:

The broker persuaded to sell the stock was tall.

در حال که دارید جمله بالا را می‌خوانید مشخصه های P600 در حال رخ دادن در مغز شما می‌باشند.

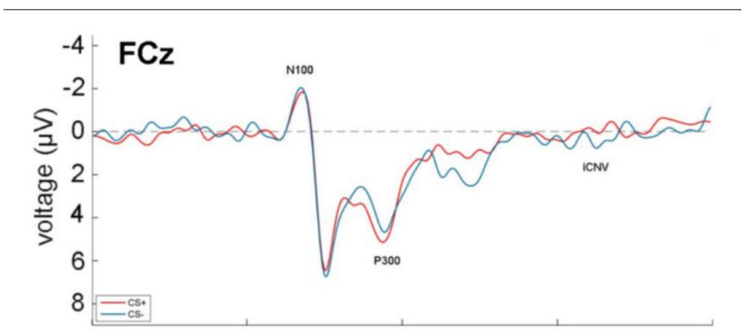
نمونه هایی از سیگنال P600 دریافت شده از قسمت های مرکزی parietal lobe:

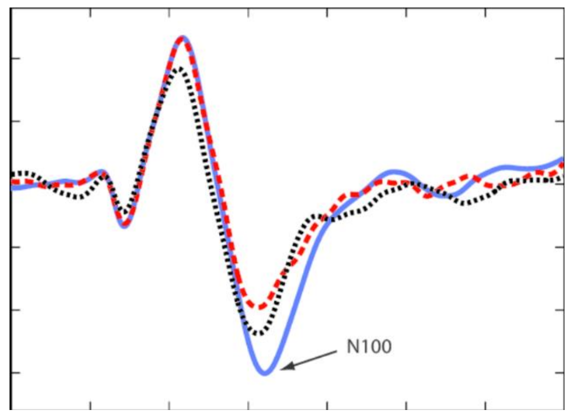


مشخصه N100:

این مشخصه اغلب به معروف به رخ دادن در اثر تحریک های شنیداری است اما اثبات شده که در اثر تحریک های دیگر مانند دیداری، بویای، درد و نیز رخ می‌دهد. تحریک های این پدیده تحریک های غیر قابل پیش بینی هستند و تغییر ولتاژ آن که منفی است می‌تواند وابسته به شرایط فیزیکی تحریک مانند فرکانس و بلندی یک صدا باشد. این مشخصه اغلب در مناطق fronto-central ثبت می‌شود. همچنین نشان داده شده که در موقع خواب نیز این مشخصه دیده می‌شود. همانطور که گفته شد تحریک های این مشخصه غیر قابل پیش بینی هستند و اگر یک تحریک به صورت تکراری اعمال شود، این مشخصه ضعیف تر می‌شود. تغییر ولتاژ در این مشخصه منفی است و تاخیر آن حدود 80 تا 120 میلی‌ثانیه می‌باشد.

نمونه هایی از رخ دادن N100:





دقت شود که در تصویر سمت چپ، قسمت بالای محور نماینده ولتاژ منفی می‌باشد.

- **با جستجو در اینترنت در مورد باند های فرکانسr مختلف اطلاعات لازم را بدست آورید و در گزارش ذکر کنید. هر باند فرکانسی نمایانگر چه فرکانس هایی است؟**

باند دلتا:

دامنه فرکانسی بین 0.5 تا 4 هرتز. معمولا مربوط به استیج سوم خواب NREM است و از می‌تواند نشان دهنده عمق خواب باشد.

باند تتا:

فرکانس بین 4 تا 10 هرتز. دو نوع مختلف شامل نوع هیپوکمپال و نوع کورتیکال دارد که نوع کورتیکال کم فرکانس تر است. نوع هیپوکمپال در فعالیت های موتوری مانند راه رفتن و همچنین مرحله خواب REM دیده می‌شود و فرکانس آن بین 6 تا 10 هرتر می‌باشد. البته در بعضی حیوانات این فرکانس ممکن است کمتر نیز باشد.

باند آلفا:

فرکانس این باند بین 8 تا 12 هرتز می‌باشد. قوی ترین سیگنال های آلفا زمانی دریافت می‌شود که فرد در حالت استراحت است و چشمانش بسته است.

باند میو:

فرکانس این باند از 7.5 تا 12.5 هرتز با تمرکز ویژه بر 9 تا 11 هرتز می‌باشد. این باند مربوط به فعالیت های عضلانی ارادی می‌باشد. وقتی بدن از لحاظ فیزیکی در حالت استراحت می‌باشد، این امواج در بدون تغییر ترین حالت خود قرار دارند.

باند بتا:

فرکانس این بند بین 12.5 تا 30 هرتز می‌باشد. امواج بتا به سه دسته از نظر فرکانس تقسیم می‌شوند. low beta با فرکانس 12.5 تا 16 هرتز، beta با فرکانس 16.5 تا 20 هرتز و high beta با فرکانس 20.5 تا 28 هرتز. امواج بتا مربوط به هوشیاری عادی در حالت بیداری هستند.

باند گاما:

امواج باند گاما دارای فرکانس 25 تا 140 هرتز هستند. از بین این فرکانس ها توجه ویژه ای روی 40 هرتز وجود دارد. باند گاما با فعالیت های شبکه های بزرگ مغزی و فعالیت های شناختی مانند حافظه فعال، توجه و غیره مربوط است.

- **با توجه به این باند های فرکانسی و قضیه نایکوئیست، چه فرکانس های نمونه برداری مناسب سیگنال های EEG است؟**

بیشترین فرکانس موجود در مغز برابر با 140 هرتز می‌باشد پس طبق قضیه نمونه برداری، از نظر تئوری فرکانس نمونه برداری باید بیشتر از 280 هرتز باشد. البته این نکته حائز اهمیت است که با توجه به نیاز آزمایش شاید بتوان از فرکانس های نمونه برداری کمتری نیز استفاده کرد. اصولا شاید لزومی به بازسازی فرکانس های 140 هرتز باند گاما نداشته باشیم و بتوانیم فرکانس نمونه برداری را کاهش دهیم تا هم حجم محاسبات و هم میزان حافظه برای ذخیره سازی کاهش یابد. البته در تعدادی از مقاله هایی که ما برخوردیم، به صورت پیش فرض از فرکانس نمونه برداری 256 هرتز استفاده می‌کردند و کاری به نوع تسک نداشتند زیرا که ماژول های این فرکانس به صورت آماده در بازار برای تحقیق های مغزی وجود دارند. البته این میزان هم می‌توان به صورت نرم افزاری با حذف کردن نقاط نمونه برداری شده، کاهش داد.

سوال اول)

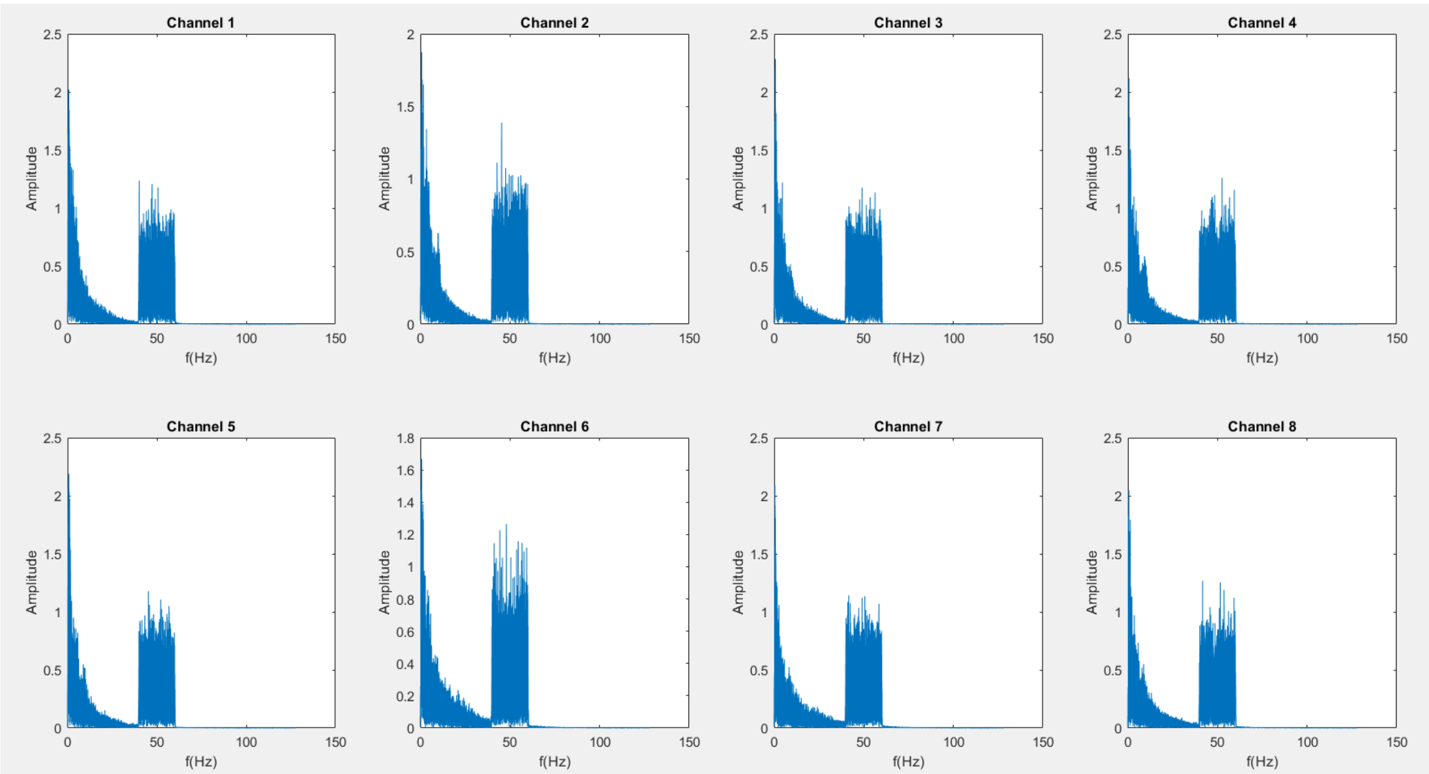
با دیتای زمانی موجود در آزمایش به راحتی Ts و در نتیجه Fs به دست می‌آیند که اینجا برابر با 256 هرتز می‌باشد.

سوال دوم)

اگر بخواهیم تمام دیتای احتمالی EEG را حفظ کنیم، مشخصا باید فرکانس را از پایین ترین فرکانس باند یعنی 0.5 هرتز و بالا ترین فرکانس باند یعنی 140 هرتز، قطع کنیم.

سوال سوم)

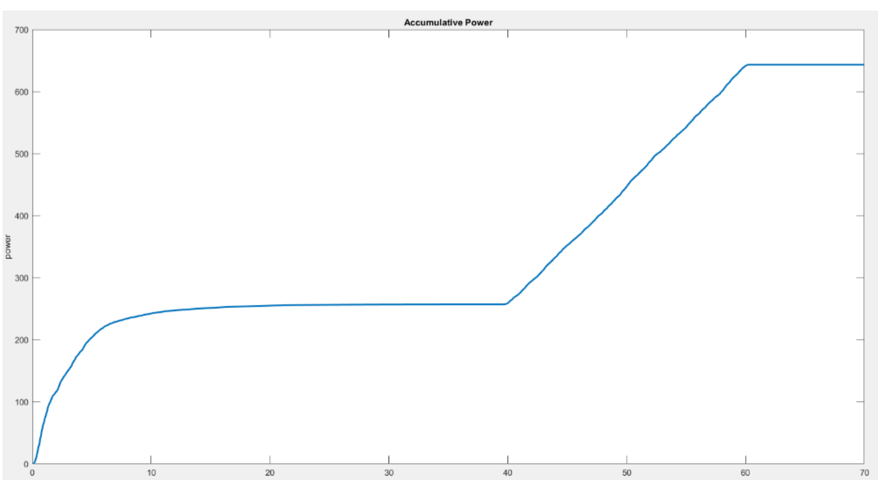
نمودار طیف فرکانسی کانال ها:



از روی نمودار ها به نظر می‌رسد که تقریبا از 60 هرتز به بعد دیتا صفر است، پس همین فرکانس را می‌توان معیار فرکانس قطع قرار داد.

سوال 4)

به جز مقدار DC در فرکانس صفر، بقیه موارد را بررسی می‌کنیم. به صورت چشمی که فرکانس 60 هرتز برای این کار کاندید شد، همچنین با استفاده از تابع bandpower() در متلب این مورد را برای یکی از الکترود ها بررسی می‌کنیم.



نمودار بالا میانگین توان تجمعی از فرکانس 0.1 تا 70 هرتز را نمایش می‌دهد. می‌بینیم که تا حدود 40 هرتز بخش خوبی از توان وجود دارد و بعد از آن از 40 تا 60 هرتز بخش بزرگتری از توان وجود دارد. اگر بخواهیم طبق بیشترین میزان انرژی سیگنال یک فرکانس قطع انتخاب کنیم باید فرکانس 60 را انتخاب کنیم. اما این معیار می‌تواند مشکل داشته باشد، زیرا بازه فرکانسی 40 تا 60 هرتز می‌تواند مربوط به نویز باشد که اینگونه نیز هست. یعنی درست است که بیشتری انرژی سیگنال تا 60 هرتز است اما این انرژی نویز برای نویز بدون استفاده است و باید حذف شود.

- **به کمک موارد بالا، فرکانس قطع فیلتر پایین گذر را نهایی کنید.**

با توجه به داده های به دست آمده تا اینجا می‌توان گفت که بهترین فرکانس قطع برای ما اطراف 40 هرتز می‌باشد. چون از این فرکانس به بعد نویز داریم و همچنین بخش بیشتر انرژی سیگنال اصلی تا این فرکانس موجود است.

- **چرا تنها کم کردن میانگین داده ها از سیگنال برای حذف فرکانس DC کافی نیست؟**

نویز اعمالی بر سیگنال نیز می‌تواند یک مقدار DC داشته باشد. اما در میان گیری از خود سیگنال، اثر میانگین نویز در سیگنال از بین نمی‌رود در نتیجه با این مقدار DC سیگنال خالص از آن کم شده ولی هنوز در فرکانس 0 یک مقدار DC مربوط به نویز خواهیم داشت. پس باید از یک فیلتر بالاگذر نیز استفاده کنیم تا این مقدار فیلتر شود.

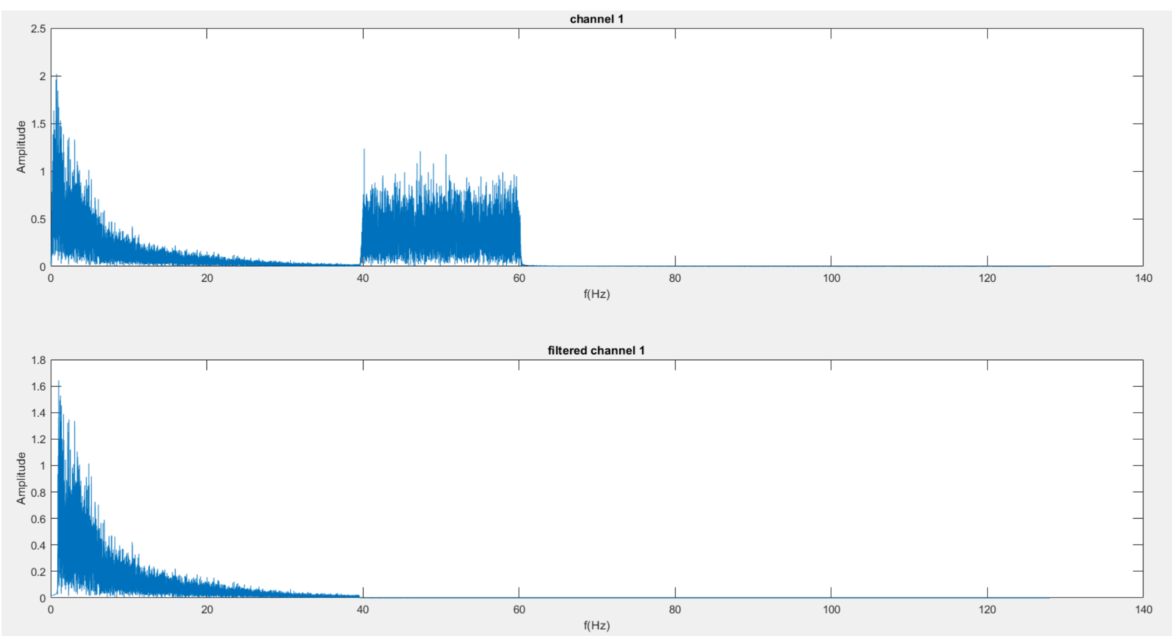
سوال 5، طراحی فیلتر میان گذر)

با توجه به مطالب گفته شده، ابتدا میانگین سیگنال را از سیگنال کم می‌کنیم و سپس یک فیلتر میان گذر با حد پایین فرکانس 1 هرتز و حد بالای 39.5 هرتز می‌سازیم.

این عملیات به صورت یک تابع در سکشن function های کد متلب زده شده(EEG.m) موجود است.

برای تابع فیلتر از تابع bandpass() متلب استفاده شده است که یک فیلتر میانگذر با دامنه فرکانسی خواسته شده به ما می‌دهد.

خروجی فیلتر برای کانال 1 در کنار سیگنال فیلتر نشده:

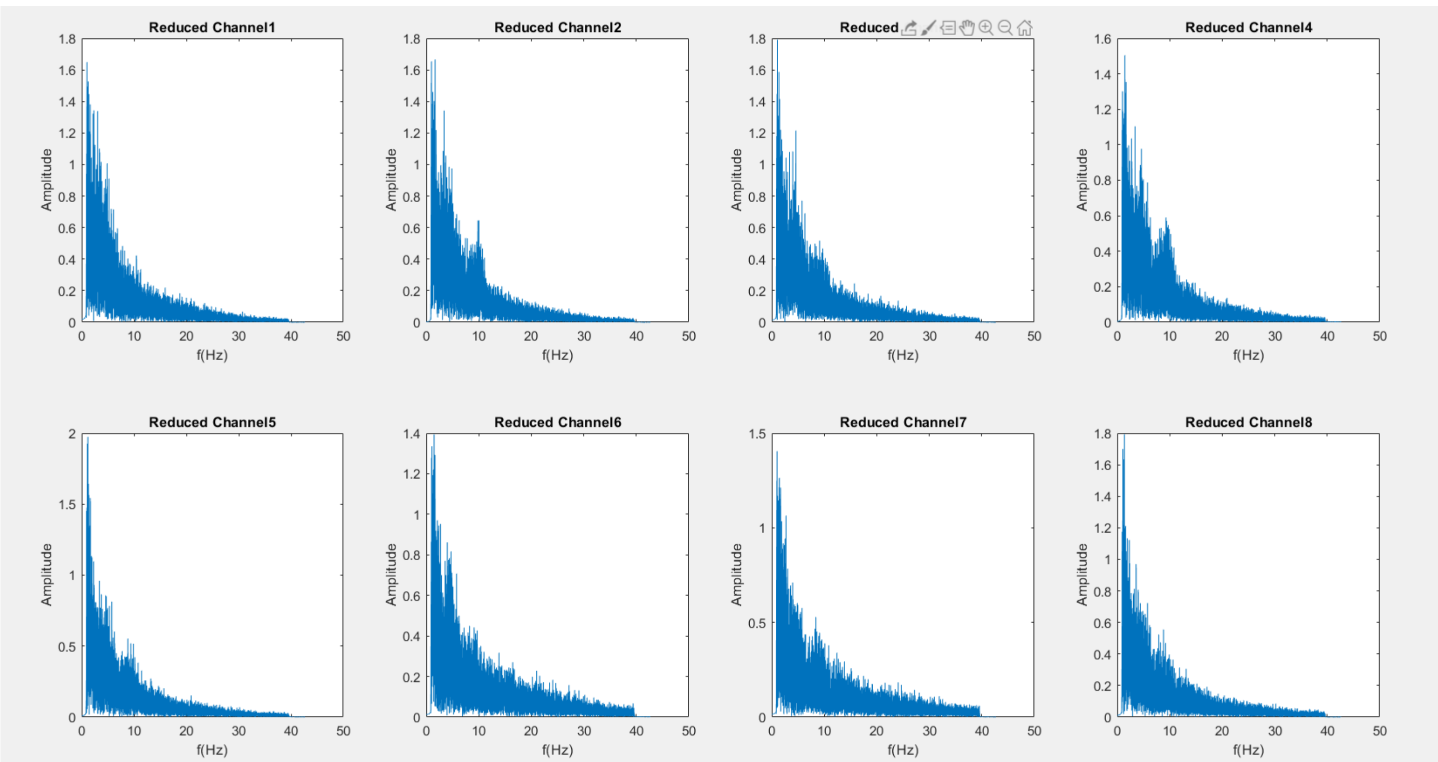


سوال 6، کاهش فرکانس نمونه برداری)

با توجه به اینکه اولا سیگنال اولیه که داریم عملا محدود به 60 هرتز است و دوما می‌خواهیم از 40 هرتز به بعد را فیلتر کنیم، عملا می‌توانیم مطابق قضیه نمونه برداری، فرکانس نمونه برداری را تا حدود 80 هرتز نیز کاهش دهیم. برای اینکار اگر از دیتایی که داریم از هر 3 تا یکی را به طور یکنواخت انتخاب کنیم، فرکانس تقریبا 85.3 می‌شود که برای رخ ندادن aliasing کافی است. البته در این فرایند چون تعداد دیتا ها لزوما بخش پذیر بر 3 نیست، ممکن است تعداد خیلی کمی دیتا در حد یکی دو تا از تنها گم شود که اصلا قابل توجه نیست و مهم این است که ما فرکانس را آنقدر کم کرده ایم که نه به مشکل aliasing می‌خوریم و نه دیتای زیادی را از دست و در عین حال پردازش های ما ساده تر می‌شود.

حالا جدای ازین توضیحات کمی به قسمت عملی کاری می‌پردازیم. در یک حالت ممکن است برای کار ما مشکل ایجاد شود. این حالت این است که در ابتدا سیگنال اولیه را که در حوزه فرکانس تا 60 هرتز ادامه دارد، با فرکانس 85.3 هرتز نمونه برداری کنیم. در این حالت مشکلی که ایجاد می‌شود این است که aliasing رخ می‌دهد و تمام مراحل ما از این به بعد اشتباه خواهد بود. برای اینکه به این مشکل برخورد نکنیم باید ابتدا سیگنال را از فیلتری که در سوال 5 ساختیم عبور دهیم و سپس با فرکانس 85.3 هرتز از آن نمونه برداری کنیم.

خروجی حوزه فرکانسی سیگنال فیلتر شده با نمونه برداری 85.3 هرتز:



همچنین دیتا های کاهش یافته و فیلتر شده با نام reduced_data ذخیره شده اند تا قابل استفاده برای قسمت های بعد باشند.

سوال 7، epoching)

در این قسمت برای دیتا های ورودی تابع از فایل reduced_data.mat که در قسمت قبل به دست آمد استفاده می‌کنیم زیرا در این فایل هم فرکانس نمونه برداری به 85.3 هرتز کاهش پیدا کرده و هم نویز از سیگنال حذف شده.

تابع epoching نوشته شده در سکشن function کد متلب EEG.mat موجود است.

بردار StimuliOnset ابتدا اینگونه ساخته می‌شود که ابتدا ایندکس همه تحریک ها در سیگنال اصلی که کاهش فرکانس داده نشده پیدا می‌شود. سپس از آنجایی که محرک ها 4 بار ذخیره شده اند از هر 4 ایندکس یکی انتخاب می‌شود تا ایندکس شروع همه محرک ها را به صورت متمیز داشته باشیم. این مقدار برابر با 2700 می‌باشد. یعنی 2700 بار تحریک اعمال شده است. سپس با توجه به رابطه بین فرکانس نمونه برداری اولیه و نمونه برداری ثانویه که یک سوم اولیه می‌باشد، این ایندکس ها را تقسیم بر 3 می‌کنیم تا ایندکس تحریک ها برای دیتای داون سمپل شده پیدا شود. البته توجه شود که این ایندکس های ممکن است در حد یک ایندکس جابجا شوند که با انتخاب مناسب بازه زمانی، اثر آن دیده نمی‌شود زیرا در نهایت فاصله زمانی بین ایندکس های پیدا شده برابر با فاصله زمانی خواهد بود.

در این تابع نوشته شده، ابتدا با استفاده از مقادیر زمانی وارد شده به تابع، تعداد نقاطی که باید در آن دیتا را قبل و بعد از یک تحریک ثبت کنیم را می‌یابیم. این تعداد نقاط در تابع با a1 و a2 نام گذاری شده اند. باید توجه کنیم که این نقاط ممکن است در حد یک نقطه جابجا شوند که همانطور که بالاتر گفته شد اثر آن قابل صرف نظر است چون بازه زمانی، ماکسیموم در حد یک دوره تناوب نمونه برداری که مقدار ناچیزی است جابجا می‌شود.

سپس با استفاده از این اینکس ها یک ماتریس 2 در 2 با نام time-trial می‌سازیم که هر سطر آن نشان دهنده‌ی نقاط زمانی در هر trial است.

سپس شکل ماتریس epoch را که ابعاد آن به صورت زیر می‌باشد را می‌سازیم:

بعد اول: برابر با نقطه های زمان است که باید در آن های دیتا را ثبت کنیم یعنی a1+a2+1

بعد دوم: برابر با تعداد trial ها است که همیشه مقدار ثابتی است. یعنی 2700

بعد سوم: برابر تعداد الکترود ها است که مقدار ثابتی دارد. یعنی 8

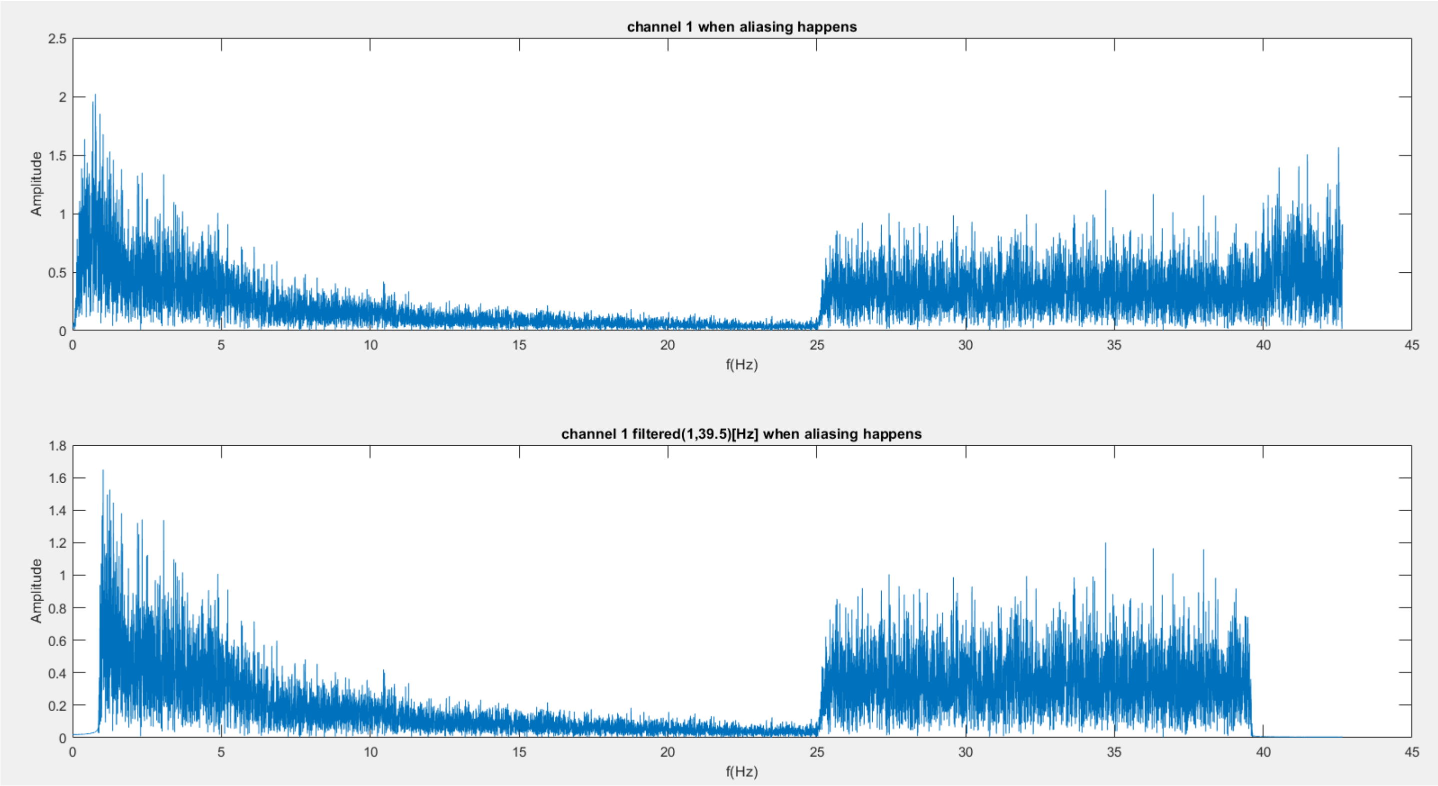
اکنون زمان پر کردن خانه های ماتریس epoch است که برای این کار از 3 حلقه for استفاده می‌کنیم که البته زمان ران کردن کد برای این الگوریتم زیاد است. مثلا برای قسمت بعدی که قرار است trial ها برابر 1000 میلی‌ثانیه باشند، مدت زمان لازم برای epoch کردن حدود 197 ثانیه بود. البته برای اینکه نیاز به ران دوباره نباشد، دیتا های قسمت بعد ذخیره شده اند.

سوال 7) انجام عملیات ایپاچینگ با پنجره های 1000 میلی ثانیه ای

این قسمت همانطور که در بالا به آن اشاره شد، انجام شده و داده های آن با نام epoch.mat در فایل های آپلودی موجود هستند. توجه کنید که اگر برای آزمایش کد، آن را اجرا کنید دوباره این فایل ساخته می‌شود، فقط حدود 3 دقیقه طول می‌کشد تا ران آن کاملا شود.

- **با توجه به اصل نایکوئیست و پدیده aliasing بگویید که چرا نمی‌توانیم فرکانس نمونه برداری را قبل از فیلتر کردن کاهش دهیم.**

این مورد با مثال در سوال 6 توضیح داده شده است و اینجا نیز به آن اشاره ای می‌کنیم. ما اصولا فرکانس نمونه برداری تا نزدیکی دو برابر ماکسیموم فرکانس سیگنال کاهش می‌دهیم. اینجا ماکسیموم فرکانس مورد سیگنال فرکانسی است که بعد از فیلتر کردن داریم. اما اگر بخواهیم قبل از اعمال فیلتر، فرکانس نمونه برداری را کاهش دهیم، ممکن است از دو برابر ماکسیموم فرکانس سیگنال کمتر شود چونکه ما این فرکانس را برای حالت فیلتر شده انتخاب کرده ایم، در نتیجه aliasing رخ می‌دهد و دیتا سیگنال به هم می‌ریزد. برای نمونه مشکل aliasing که ممکن است رخ دهد را در سوال 6 بررسی می‌کنیم(توجه کنید که در کد متلب قسمت جداگانه ای برای این قسمت تعیین نشده است و فقط خروجی مورد نظر را آورده ایم):



مشاهده می‌شود که برای سیگنال کانال یک در حوزه فرکانسی، نموداری به دست آورده ایم که با نمودار های پیشین هم از نظر فرکانسی و هم دامنه ای متفاوت است. این تفاوت به دلیل رخ دادن پدیده aliasing می‌باشد.

- **تفاوت ابتدا epoch کردن با ابتدا فیلتر کردن**

اگر ابتدا epoch کنیم و بعد بخواهیم فیلتر کنیم، مشکل ذکر شده در سوال بالا رخ می‌دهد. یعنی ممکن است به علت نقض قضیه نمونه برداری، پدیده aliasing رخ دهد و سیگنال ما را به هم بزند. در نتیجه باید ابتدا فیلتر کنیم و بعد epoch کنیم یا طول فرکانسی فیلتر از طول فرکانسی epoch کوچکتر باشد.

- **چرا باید شروع آزمایش را با تاخیر شروع کنیم؟**

در ابتدای کار الکترود ها نویز زیادی دریافت می‌کنند تا پوست سر به الکترود ها عادت کند و در بعضی موارد امپدانس بین پوست سر و الکترود ها تنظیم شود. در نتیجه اگر اول کار شروع به انجام آزمایش کنیم، سیگنال ما بسیار نویزی خواهد بود.

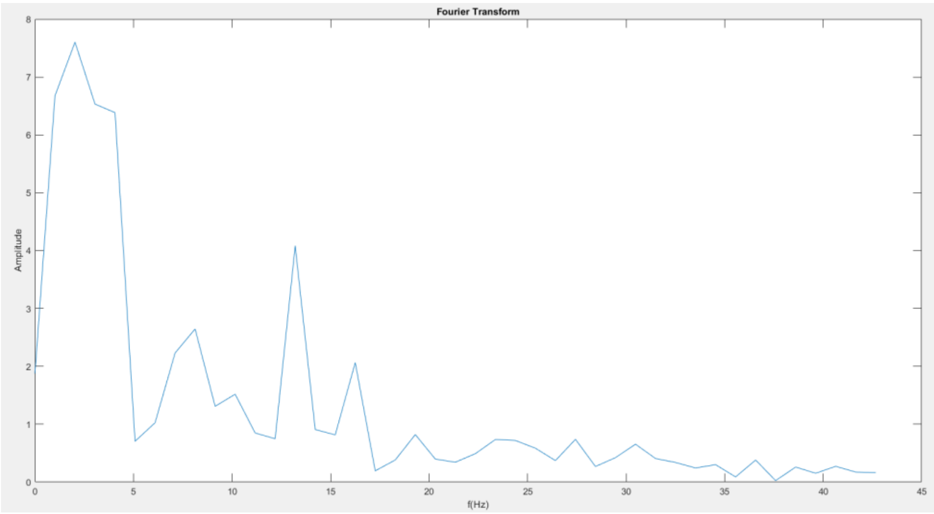
سوال 8) به دست آوردن انرژی هر باند در هر epoch

در ابتدا این نکته قابل ذکر است که تابع های فیلتر مختلف، از جمله تابع bandpass() خود متلب که قبلا استفاده شد، اینجا جواب نمی‌داد و بسیار کند بود. برای همین در این بخش از فیلتر داده شده یعنی BPF استفاده شده است. البته یک رفتاری که این فیلتر در حوزه زمان دارد این است که دیتا را از زمانی به قبل تقریبا صفر می‌کند. که این مورد اصولا باعث کاهش انرژی هر epoch نیز می‌شود.

انرژی سیگنال گسسته:

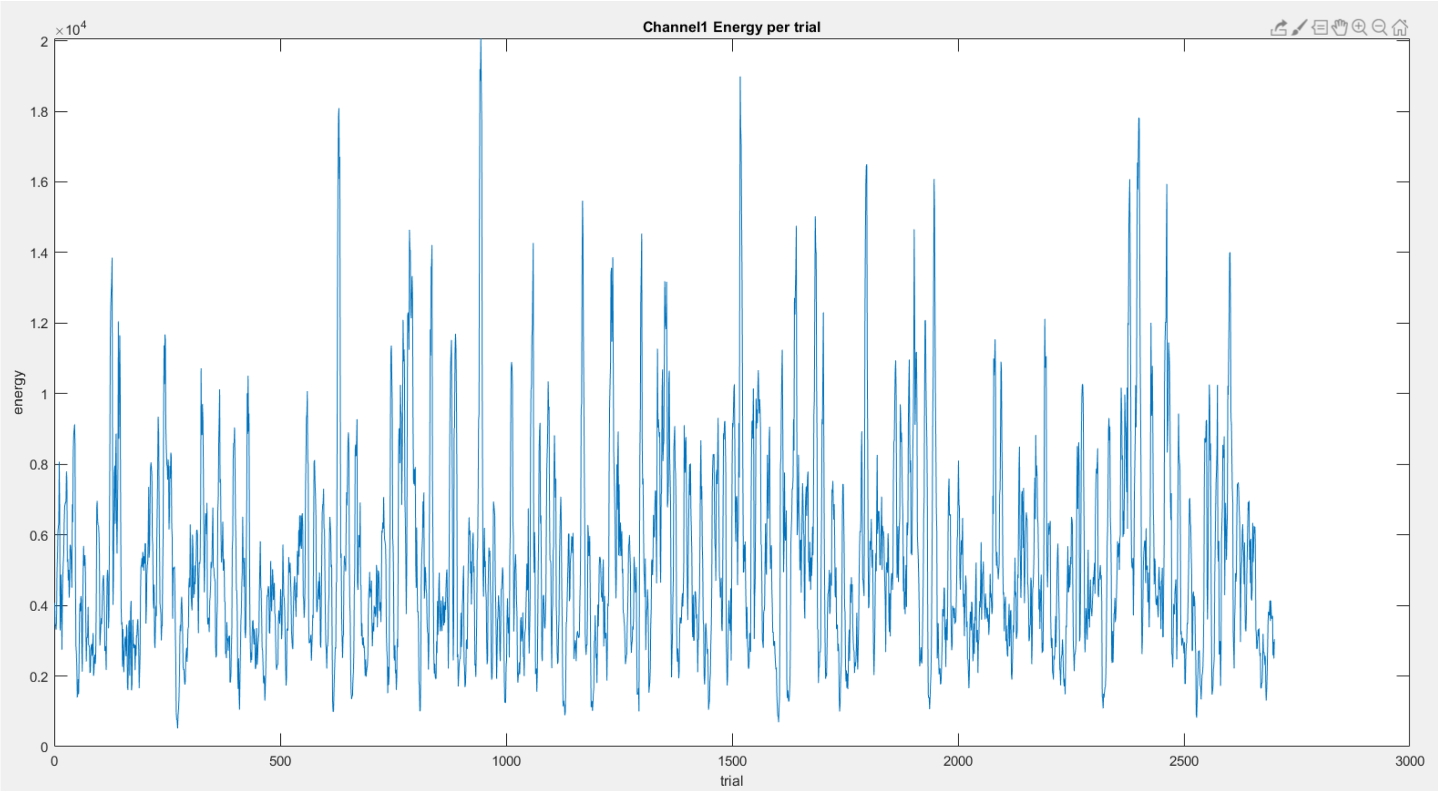
$$E = \sum_{n=-\infty}^{\infty} |x[n]|^2$$

اگر در حوزه فرکانسی به دیتا ها نگاه کنیم، اصولا بیشترین مقدار فرکانس آن ها در حدود 2 تا 15 هرتز می‌باشد. یعنی اغلب انرژی سیگنال ها در این باند فرکانسی می‌باشد. در نتیجه در همین باند سیگنال ها را فیلتر می‌کنیم.



سپس انرژی هر epoch فیلتر شده را محاسبه می‌کنیم.

برای مثال، مقدار انرژی سیگنال کانال 1 را در ترایال های مختلف رسم می‌کنیم.



خوشه بندی بر مبنای همبستگی

- **اثبات رابطه:**

این رابطه عملا کمی تغییر یافته نامساوی هولدر یا اینجا در واقع نامساوی کوشی شوارتس می‌باشد. طبق این نامساوی:

$$[\int_{-\infty}^{\infty} X(t)Y(t)]^2 \leq \int_{-\infty}^{\infty} X^2(t)dt \int_{-\infty}^{\infty} Y^2(t)dt \Rightarrow \frac{[\int_{-\infty}^{\infty} X(t)Y(t)]^2}{\int_{-\infty}^{\infty} X^2(t)dt \int_{-\infty}^{\infty} Y^2(t)dt} \leq 1$$

$$\Rightarrow \frac{|\int_{-\infty}^{\infty} X(t)Y(t)|}{\sqrt{\int_{-\infty}^{\infty} X^2(t)dt \int_{-\infty}^{\infty} Y^2(t)dt}} \leq 1$$

خود قضیه کوشی شوارتس نیز به روش های مختلف قابل اثبات است اما اینجا دیگر نیاز نیست چون قضیه بسیار معروف و پذیرفته شده ای می‌باشد و کاربرد آن برای ما کافی است.

- **اثبات رابطه:**

اثبات یک طرف قضیه که در حد یک جاگذاری ساده است.

برای طرف دیگر:

$$\Rightarrow \frac{\int_{-\infty}^{\infty} X^2(t)dt}{\int_{-\infty}^{\infty} X(t)Y(t)dt} = \alpha \text{ and } \frac{\int_{-\infty}^{\infty} Y^2(t)dt}{\int_{-\infty}^{\infty} X(t)Y(t)dt} = \frac{1}{\alpha} \rightarrow \int_{-\infty}^{\infty} X^2(t)dt = \alpha \int_{-\infty}^{\infty} X(t)Y(t)dt$$

$$\Rightarrow \int_{-\infty}^{\infty} X(t)(X(t) - \alpha Y(t))dt = 0(1)$$

If we do the same for the other equation:

$$\int_{-\infty}^{\infty} Y(t)(X(t) - \alpha Y(t))dt = 0(2)$$

Subtracting $\alpha \times (2)$ from (1) we get:

$$\Rightarrow \int_{-\infty}^{\infty} (X(t) - \alpha Y(t))^2 dt = 0$$

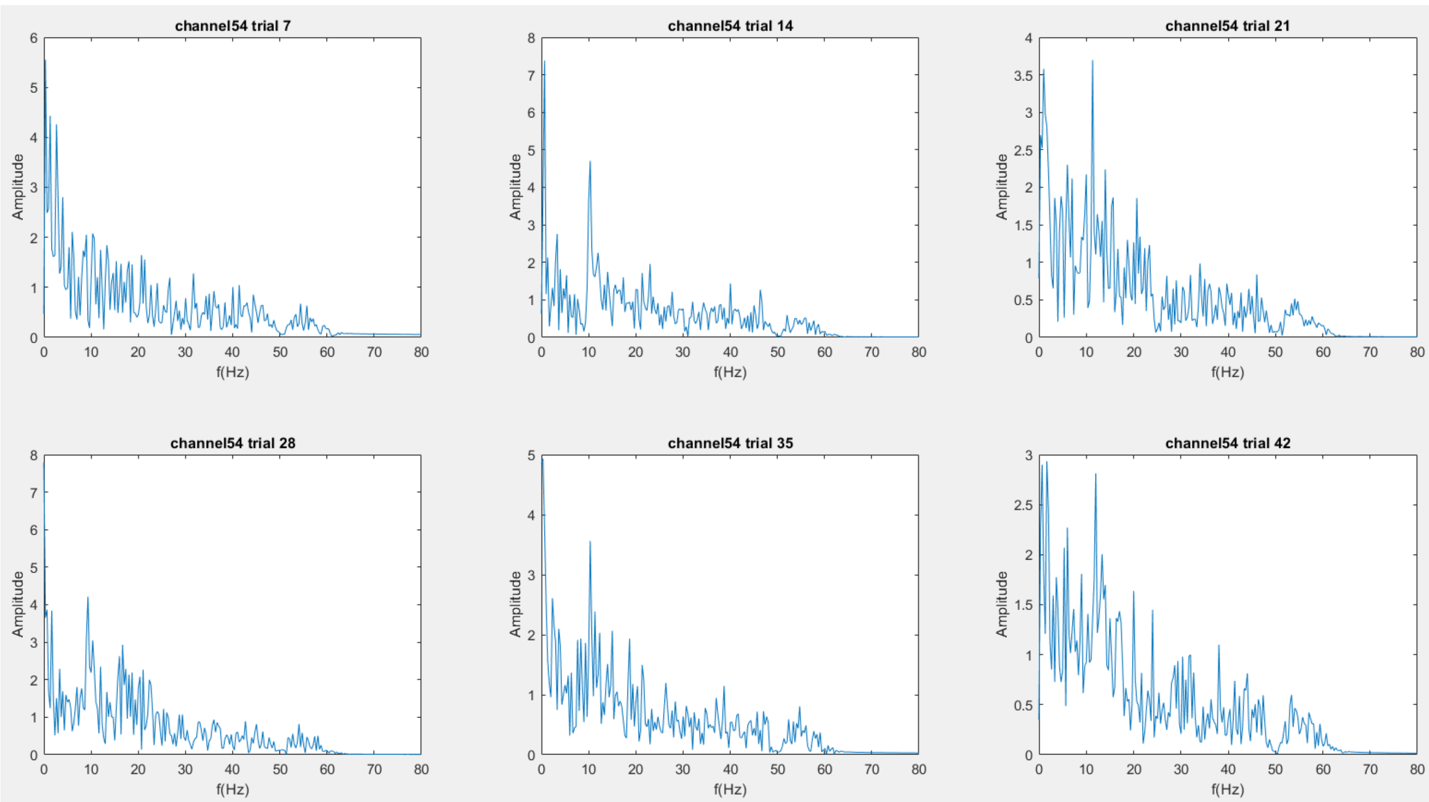
Because inside the integral is a positive number, the only answer to this equation is $X(t) = \alpha Y(t)$

- **استدلال کنید چرا این معیار، معیار مناسبی برای سنجش شباهت دو سیگنال است؟**

معیار همبستگی در واقع پشتوانه ریاضی قوی ای دارد و تئوری آن کاملا مشخص و شناخته شده می‌باشد. همچنین عملا سیگنال ها را در تمام بازه های زمانی بررسی می‌کند. علاوه بر آن هر چه قدر دو سیگنال شباهت بیشتری داشته باشند، همبستگی آن ها بیشتر می‌شود تا وقتی که یکی از سیگنال ها ضریبی از سیگنال دیگر باشد که همبستگی برابر یک خواهد بود.

سوال 1) اعمال فیلتر اولیه بر داده ها

مانند قبل، ابتدا به کمک تابع SemiBandFFT سیگنال ها را در حوزه فرکانسی بررسی می‌کنیم.

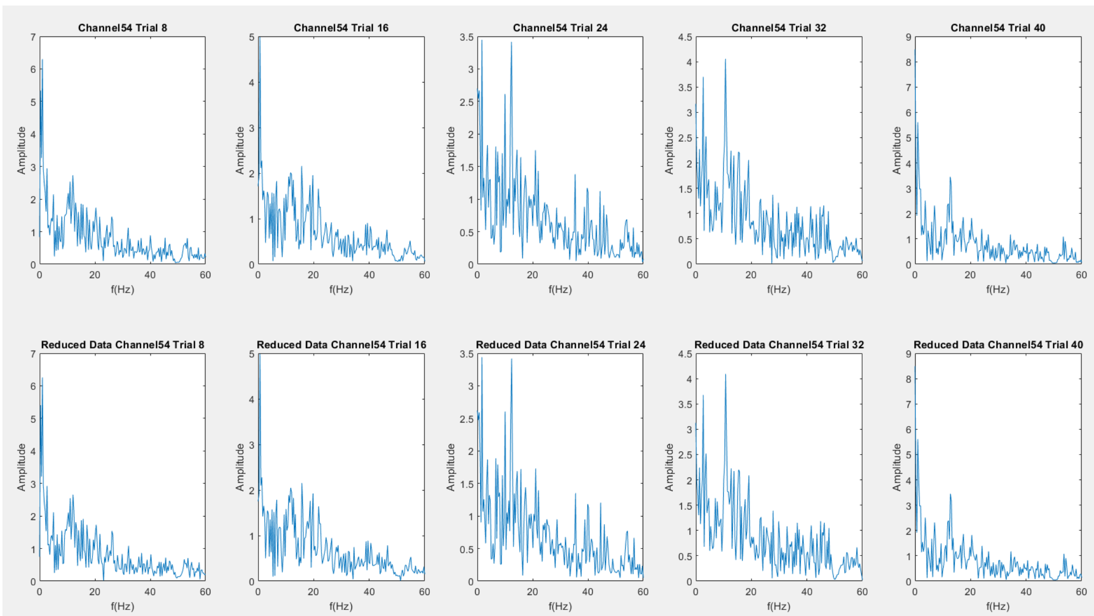


نمودار های بالا برای چند ترایال از کانال 54 می‌باشند. مشاهده می‌کنیم که در خود فرکانس 50 هرتز که اصولا فرکانس نویز ما می‌باشد، دامنه نداریم و در کل سیگنال بدون نویزی به نظر می‌رسد. اگر به بقیه کانال ها به غیر از کانال 54 نگاه کنیم نیز همینگونه است و این مورد تنها برای مثال آورده شده است. به همین دلایل، فیلتر فرکانسی خاصی به سیگنال اعمال نمی‌کنیم و تنها مقدار میانگین را از آن کم می‌کنیم.

سوال 2) کاهش فرکانس نمونه برداری

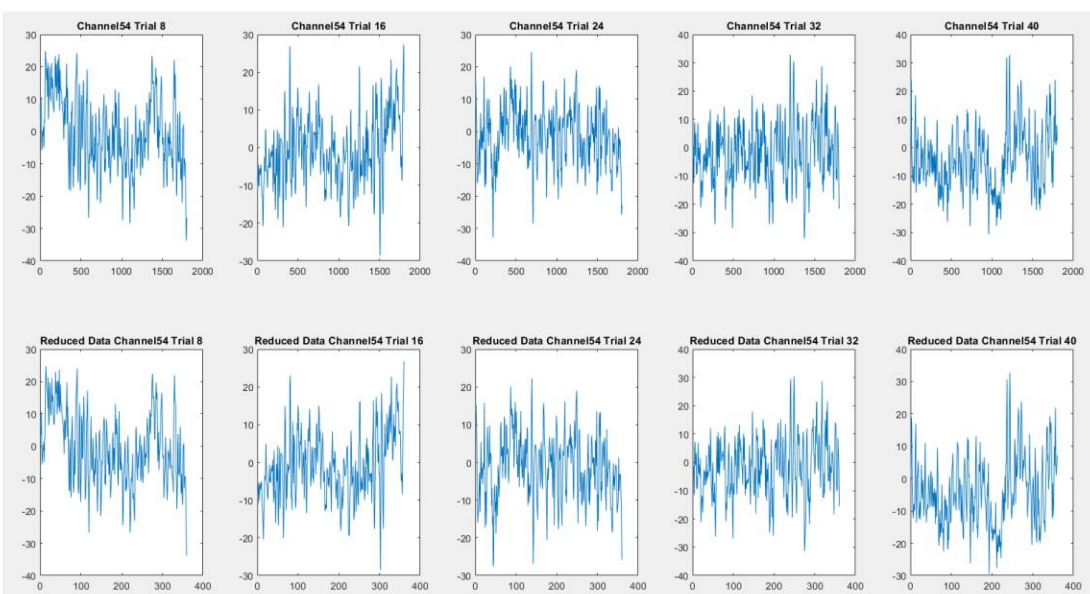
همانطور که در سوال قبل مشاهده شد، بازه فرکانسی سیگنال ها حدودا تا 60 هرتز می‌باشد پس می‌توانیم فرکانس نمونه برداری را تا حدود 120 هرتز کاهش دهیم. در نتیجه با توجه به فرکانس اولیه که برابر با 600 است، از هر 5 دیتا یکی را انتخاب می‌کنیم. همچنین از آنجایی که می‌خواهیم از این داده ها استفاده کنیم، آن را با نام reduced_64channeldata.mat ذخیره می‌کنیم.

مقایسه فرکانسی داده های کاهش یافته با داده های اصلی:



مشاهده می‌کنیم که عملیات کاهش تعداد نقاط به درستی انجام شده است.(دقت کنید در نمودار های بالا که مربوط به دیتای اصلی می‌باشند، محور فرکانس محدود به 0 تا 60 شده است)

مقایسه زمانی داده های کاهش یافته با داده های اصلی:



در شکل صفحه قبل نیز، می‌بینیم که در حوزه زمانی نیز داده ها مطابقند.

سوال 3) یافتن ماتریس همبستگی و خوشه بندی

برای پیدا کردن ماتریس همبستگی، از آنجایی که این ماتریس متقارن است و برای کاهش محاسبات، می‌توانیم نصف ماتریس همبستگی را بسازیم و سپس آن را در قسمت دیگر کپی کنیم. البته اینجا چون کل محاسبات ماتریس حدود 2 ثانیه طول می‌کشد، احساس نیاز به این فرایند اینجا وجود ندارد.

برای اینکه تمام طول سیگنال را برای هر الکترود داشته باشیم، ابتدا همه ترایال ها را به ازای هر کانال به یکدیگر متصل می‌کنیم. این عملیات بسیار ساده و با حلقه for و concatenate کردن انجام می‌شود. سپس عملیات گفته شده در بالا انجام می‌شود.

- **فاصله دو کانال**

برای فاصله دو کانال رابطه Eisen correlation distance پیشنهاد می‌شود:

$$d_{eisen}(x,y) = 1 - \frac{\left|\sum_{i=1}^{n} x_i y_i\right|}{\sqrt{\sum_{i=1}^{n} x_i^2 \sum_{i=1}^{n} y_i^2}}$$

- **الگوریتم های فاصله دو خوشه**

الگوریتم اول:

با استفاده از روش زیر، که فاصله بین دو خوشه را تعریف، می‌کند، فاصله ها را در هر مرحله می‌توان آپدیت کرد:

$$\frac{1}{|\mathcal{A}|\cdot|\mathcal{B}|}\sum_{x\in\mathcal{A}}\sum_{y\in\mathcal{B}} d(x,y) \qquad d_{(\mathcal{A}\cup\mathcal{B}),X} = \frac{|\mathcal{A}|\cdot d_{\mathcal{A},X} + |\mathcal{B}|\cdot d_{\mathcal{B},X}}{|\mathcal{A}|+|\mathcal{B}|}$$

الگوریتم دوم:

$$d_{(i\cup j),k} = \frac{d_{i,k}+d_{j,k}}{2}$$

روش اول اصطلاحا UPGMA و روش دوم، WPGMA نام دارد.

کد نوشته شده برای قسمت خوشه بندی در ادامه کد مربوط به محاسبه ماتریس همبستگی می‌باشد.

در این کد تابع خوشه بندی به صورت زیر شده شده:

```
function Cluster = CorrelationCluster(InputCorrMat, DistanceMeasure, method);
```

ورودی method برابر با 1 یا 2 است که با انتخاب 1، الگوریتم اول و با انتخاب 2، الگوریم دوم برای آپدیت فاصله ها استفاده می‌شود. همچنین DistanceMeasure برابر با مقدار ماکسیموم فاصله قابل قبول برای یکی کردن دو خوشه می‌باشد.

توجه: به این نکته توجه کنید که جزییات این تابع زده شده تا حد خیلی خوبی پیچیده است و از توان نوشتاری کمی خارج می‌باشد. به همین دلیل اینجا توضیح جزییات تابع آورده نشده است.

خروجی کلاسترینگ برای روش دوم:

	1	2	3	4	5	6	7	8	9	10	11	12
1	1	6	7	10	19	20	24	28	52	58	62	63
2	2	11	8	43	0	21	29	30	0	0	0	0
3	3	15	9	0	0	23	31	32	0	0	0	0
4	4	35	12	0	0	25	57	59	0	0	0	0
5	5	39	13	0	0	26	61	60	0	0	0	0
6	14	44	16	0	0	27	0	0	0	0	0	0
7	33	48	17	0	0	49	0	0	0	0	0	0
8	34	0	18	0	0	53	0	0	0	0	0	0
9	38	0	22	0	0	54	0	0	0	0	0	0
10	0	0	36	0	0	55	0	0	0	0	0	0
11	0	0	37	0	0	56	0	0	0	0	0	0
12	0	0	40	0	0	0	0	0	0	0	0	0
13	0	0	41	0	0	0	0	0	0	0	0	0
14	0	0	42	0	0	0	0	0	0	0	0	0
15	0	0	45	0	0	0	0	0	0	0	0	0
16	0	0	46	0	0	0	0	0	0	0	0	0
17	0	0	47	0	0	0	0	0	0	0	0	0
18	0	0	50	0	0	0	0	0	0	0	0	0
19	0	0	51	0	0	0	0	0	0	0	0	0

مشخصا در این خوشه بندی، سیگنال های دور نسبتا دور از هم از نظر مکانی در یک خوشه وجود ندارند و همچنین بسیاری از سیگنال های نزدیک به هم، در یک خوشه قرار گرفته اند. برای مثال پنج سیگنال 1 2 3 4 5 که بسیار نزدیک هستند را در خوشه اول می‌بینیم.

خروجی کلاسترینگ برای روش اول:

	1	2	3	4	5	6	7	8	9	10	11
1	1	6	10	15	19	23	24	52	58	62	63
2	2	7	43	39	0	27	29	0	0	0	0
3	3	8	0	48	0	28	31	0	0	0	0
4	4	9	0	0	0	30	32	0	0	0	0
5	5	11	0	0	0	55	57	0	0	0	0
6	14	12	0	0	0	56	60	0	0	0	0
7	33	13	0	0	0	59	61	0	0	0	0
8	34	16	0	0	0	0	0	0	0	0	0
9	38	17	0	0	0	0	0	0	0	0	0
10	0	18	0	0	0	0	0	0	0	0	0
11	0	20	0	0	0	0	0	0	0	0	0
12	0	21	0	0	0	0	0	0	0	0	0
13	0	22	0	0	0	0	0	0	0	0	0
14	0	25	0	0	0	0	0	0	0	0	0
15	0	26	0	0	0	0	0	0	0	0	0
16	0	35	0	0	0	0	0	0	0	0	0
17	0	36	0	0	0	0	0	0	0	0	0
18	0	37	0	0	0	0	0	0	0	0	0
19	0	40	0	0	0	0	0	0	0	0	0
20	0	41	0	0	0	0	0	0	0	0	0
21	0	42	0	0	0	0	0	0	0	0	0
22	0	44	0	0	0	0	0	0	0	0	0
23	0	45	0	0	0	0	0	0	0	0	0

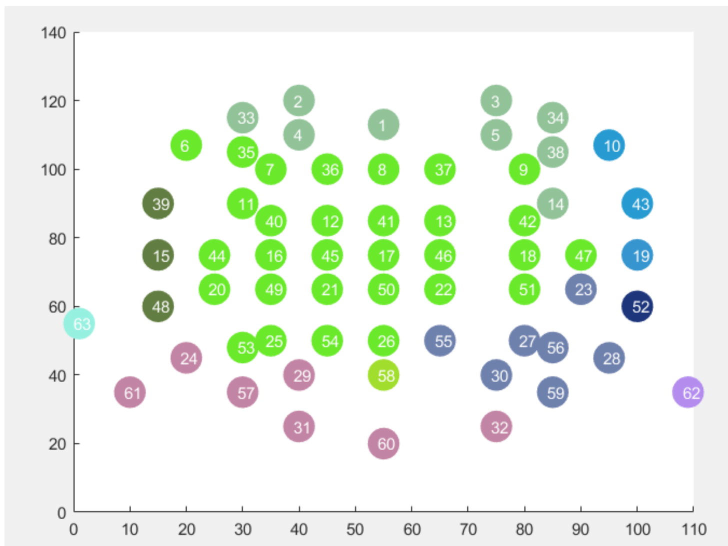
توجه: در انتهای کد این بخش، بعد از قسمت خروجی گرفتن از تابع Cluster، این خروجی فرمت استاندارد ندارد برای همین با روشی این تابع استاندارد شده است. در این روش از یک تابع پیش فرض به نام groupcount() استفاده شده است که گویا برای متلب 2019 به بعد می‌باشد و ممکن است ران کردن این قسمت انتهایی، برای متلب های قبلی عملی نباشد. این خروجی ها برای آستانه 0.3 می‌باشند.

- **تحلیل خوشه ها**

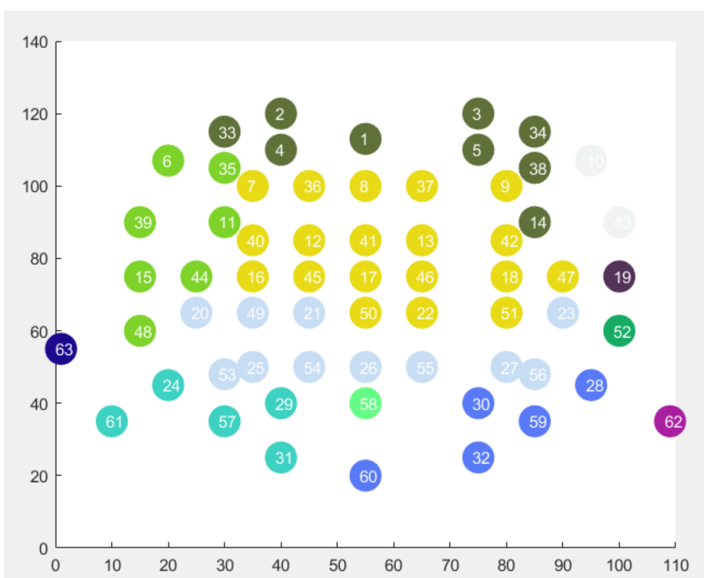
برای بررسی نتیجه خوشه بندی، خوشه ها یک به یک با نقشه مغزی که در صورت گزارش قرار گرفته، بررسی شده و همانطور که بالا تر گفته شد، تعداد خوبی از کانال های هر خوشه، کانال های نزدیک به هم هستند و همچنین با بررسی lobe های مغز، تعدادی از کانال که نزدیک هم نیستند، روی یک lobe مغزی قرار گرفته اند.

همچنین یکی از گروه ها، یک تابع خیلی جذاب برای بررسی خروجی خوشه ها زده بود که با اجازه آنها، از این تابع برای تهیه خروجی استفاده شد، اما با توجه به حفظ کپی رایت، کد آنها در سورس این پروژه آورده نشده است. در واقع نیازی به این کد در پروژه نبود ولی از جهت جذابیت و عملکرد، خروجی آن آورده شده است. در خروجی های زیر، قسمت های رنگی، مربوط به کانال های قرار گرفته در یک خوشه هستند.

خروجی این تابع برای الگوریتم اول:



کانال های همرنگ، در یک خوشه قرار دارند.



سوال 4) خوشه بندی برای سیگنال 8 کاناله

برای دیتاست 8تایی که اطلاعات کافی راجع به آن نداشتیم هم مطابق بخش قبل عمل کردیم؛ به این صورت که سیگنال پیوسته خود را به عنوان full signal بخش قبل مستقیما وارد تابع کردیم و با کمی تغییر در ابعاد ماتریس(که از تفاوت نوع epoching ما و دیتاست موجود نشئت میگرفت) و آستانه 0.1 نتیجه زیر حاصل شد.

1	1	3	5	7
2	2	4	6	8
3	0	0	0	0
4	0	0	0	0
5	0	0	0	0
6	0	0	0	0
7	0	0	0	0
8	0	0	0	0

برای کد این قسمت، تابع قبلی کمی تغییر کرده و به شکل زیر تعریف شده است:

```
function Cluster = CorrelationCluster_for8channel(InputCorrMat, DistanceMeasure, method);
```

در این تابع ثانویه همه چیز مانند تابع اصلی است و تنها در داخل تابع برخی بعد ها تغییر کرده اند.