
«بسمه تعالی»

«گزارش پروژه متلب درس سیگنال‌ها و سیستم‌ها»



دانشکده برق دانشگاه صنعتی شیراز

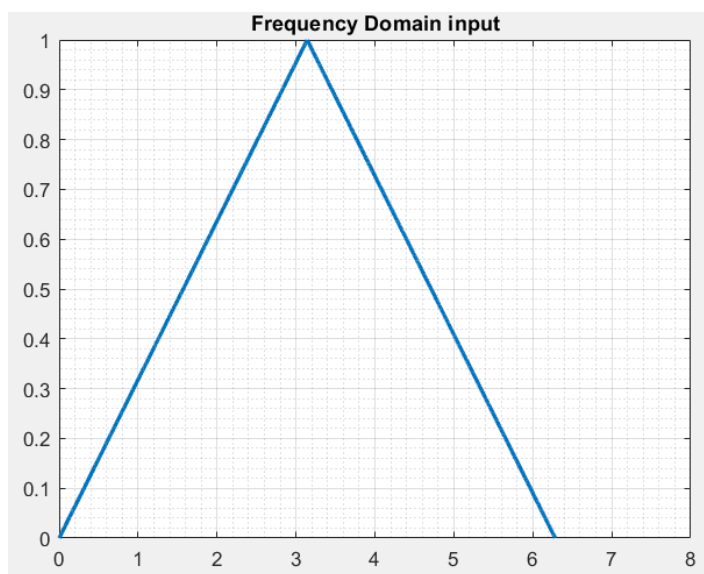
میثم امیر سرداری | امیر حسین مباشری

استاد درس: دکتر کربلایی آقاجان

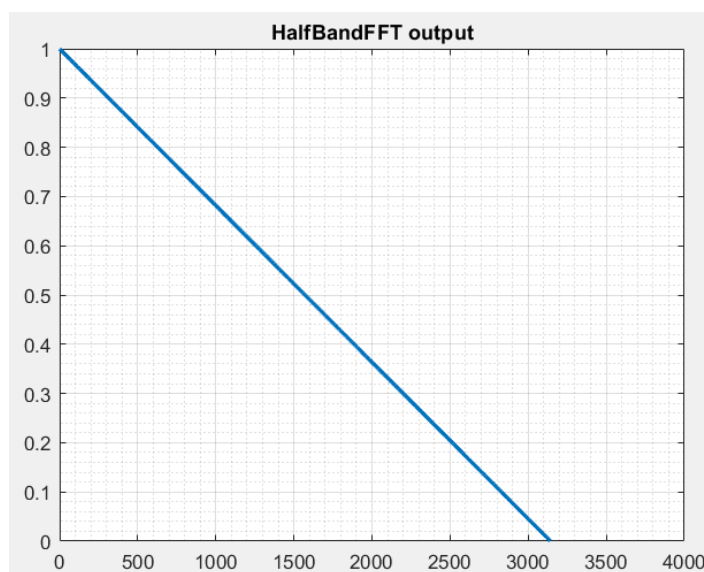
تابستان ۱۴۰۰

نمونه برداری – Sampling:

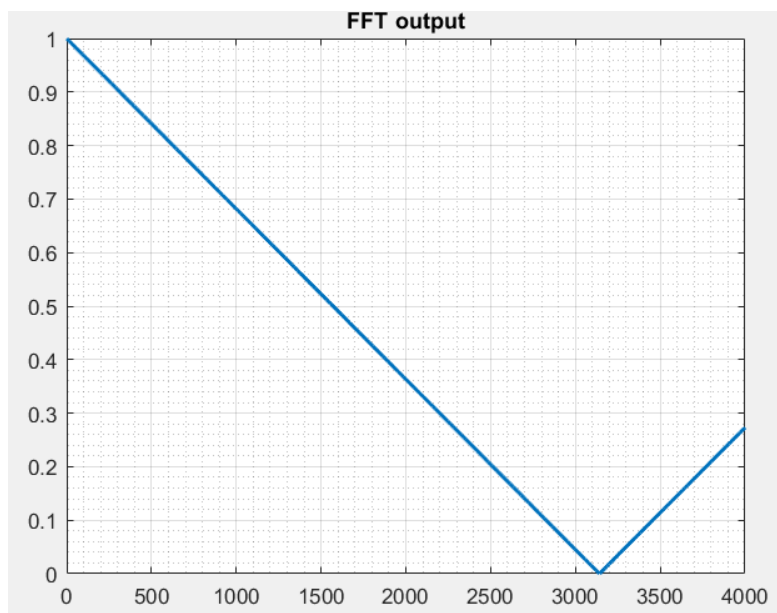
- 1) ابتدا یک پالس مثلثی را به عنوان تبدیل فوریه سیگنال مثال را با استفاده از تابع `triangularPulse` تعریف می‌کنیم تا به آرایه‌ای به شکل زیر دست پیدا کنیم. از این آرایه به دست آمده تبدیل فوریه وارون (`ifft`) می‌گیریم تا فرم زمانی سیگنال ورودی به دست بیاید.
- از آنجا که می‌دانیم تابع `fft` خروجی را به طور متقارن حول محور عمودی نمی‌چیند و صرفاً مقادیر را از درایه اول در نظر می‌گیرد، نمودار را به اندازه پی به سمت راست شیفت داده و سپس به `ifft` وارد می‌کنیم.



سیگنال به دست آمده را به عنوان آرگومان تابع `HalfBandFFT` وارد می‌کنیم و در خروجی با سیگنالی به فرم زیر مواجه می‌شویم:



همانطور که مشهود است این فرم به دست آمده همان مقادیر ما به ازای ناحیه بین صفر تا پی در نمودار حوزه فرکانس است که در ابتدا تعریف کرده بودیم.



خروجی تابع fft پیش از اعمال فیلتر پایین گذر بر روی آن، همانطور که انتظار می‌رود مقادیر ما به ازای فرکانس π به فرکانس $\pi \times F_S$ نظیر شده است.

برای پیاده سازی این تابع پس از محاسبه fft سیگنال ورودی، آن را با تابع fftshift به شکلی تنظیم میکنیم که صفر فرکانسی آن متناظر با درایه یک قرار بگیرد و حول محور عمودی تقارن پیدا کند. سپس نیمه سمت راست را تا مقدار پی جدا کرده و به نمایش می‌گذاریم بلکه به خروجی مشهود دست پیدا کنیم.

همچنین می‌دانیم در تبدیل فوریه گسسته سیگنال حوزه فرکانس به دست آمده با دوره تناوب پی متناوب است، بنابراین پس از محاسبه طول آرایه به دست آمده از خروجی تابع fft، نصف آن را متناظر با مقدار π قرار میدهیم. با توجه به توضیحات دستور العمل پروژه می‌دانیم برای به دست آوردن فرکانس‌های معادل در حالت پیوسته، می‌بایست فرکانس نمونه برداری را در مقادیر به دست آمده از فرکانس‌های حالت گسسته ضرب کرده ($\omega = F_S \times \Omega$) و در نهایت با استفاده از این بردار به دست آمده، مقادیر محور افقی را منظم می‌کنیم.

- در نمونه مورد آزمایش که تصویر آن بالاتر مشاهده شد، فرکانس نمونه برداری ۱۰۰۰ هرتز در نظر گرفته شده بود.

(2) از شرط نایکوئیست برای پیشگیری از aliasing می‌دانیم می‌بایست $F_S > 2f_m$ که در آن f_m بیشینه فرکانسی است که سیگنال دارای مقدار می‌باشد.

همچنین می‌دانیم به هنگام تبدیل سیگنال نمونه برداری شده به رشته، خروجی آن با دوره تناوب 2π متناوب می‌شود؛ بنابراین اگر می‌خواهیم از تداخل جلوگیری کنیم می‌بایست قرار دهیم $\Omega_m < \pi$

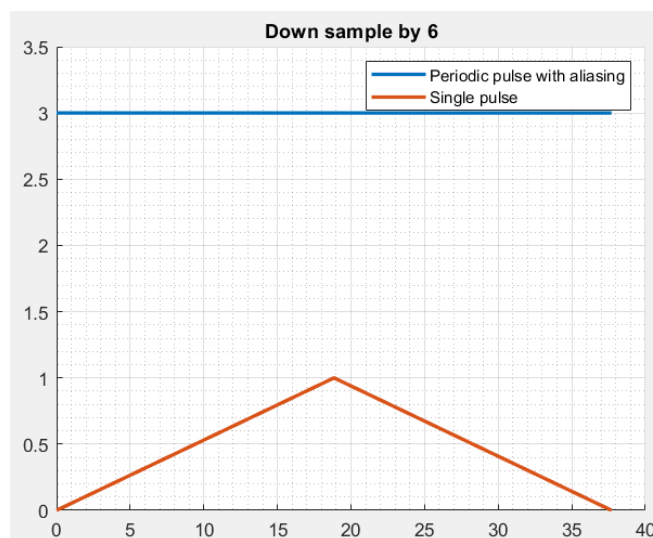
$$\Omega_m = \frac{\omega_m}{F_S} = \frac{2\pi f_m}{F_S} < \pi \quad \Rightarrow \quad f_m < F_S$$

با توجه به توضیحات مشروح در دستور العمل نیز داریم:

(3) در این بخش از هر سری شش‌تایی از نمونه‌های به دست آمده متوالی یکی را نگه داشته و باقی را حذف می‌کنیم، این امر به معنای Down Sampling با مرتبه شش است. این کار در حوزه فرکانس با رابطه زیر توصیف می‌شود:

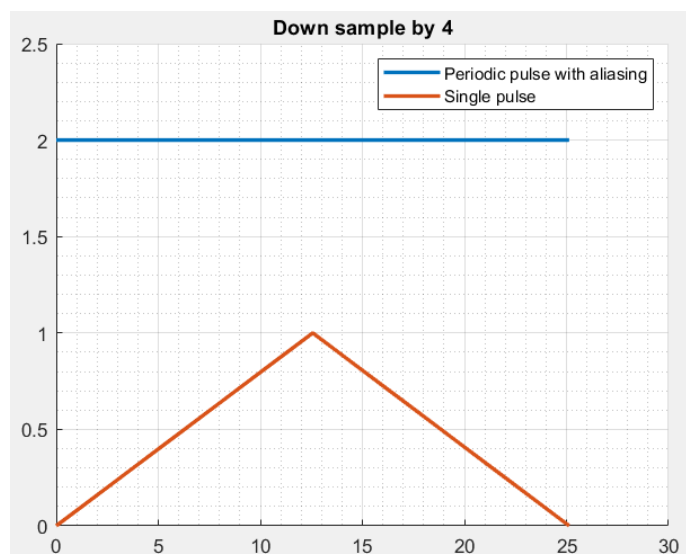
$$X_d(\Omega) = \frac{1}{6} \sum_{k=0}^5 X\left(\frac{\Omega - 2k\pi}{6}\right)$$

در اینجا نیز تک پالس درون سیگما را با انبساط عرضی با ضریب ۶ محاسبه کرده و در گام بعدی مجموع مطلوب را محاسبه می‌کنیم، از آنجا که به وضوح شرط نایکوئیست برقرار نمی‌باشد انتظار داریم با پدیده aliasing مواجه باشیم که در تصویر زیر نیز مشهود است:



در صورتی که Down Sampling را با مرتبه چهار انجام دهیم نیز مشخصاً با تداخل مواجه خواهیم بود که این بار با رابطه زیر توصیف شده و در شبیه سازی‌ای مشابه قسمت قبل، با فرم به دست آمده در تصویر زیر مواجه می‌شویم:

$$X_d(\Omega) = \frac{1}{4} \sum_{k=0}^3 X\left(\frac{\Omega - 2k\pi}{4}\right)$$



آشنایی با سیگنال‌های EEG:

1) پتانسیل‌های وابسته به رخداد (ERP) ولتاژهای بسیار کوچکی هستند که در اثر انتقالات و تحرکات سنابسی هم‌زمان تعداد زیادی نورون (هزاران یا حتی میلیون‌ها) در ساختار مغز پدید می‌آیند. این ولتاژها در پاسخ به وقایع یا محرک‌های خاص حسی، حرکتی و یا شناختی ایجاد می‌شوند و از این رو امکان مطالعه روابط فیزیکی میان فرآیندهای ذهنی را فراهم می‌کند.

ERPها را می‌توان به دو دسته تقسیم کرد؛ امواج اولیه که در حدود ۱۰۰ میلی ثانیه پس از محرک به اوج می‌رسند و تا حدود قابل توجهی به پارامترهای فیزیکی محرک بستگی دارند. در دسته دیگر ولتاژهای ایجاد شده در هنگامی رخ میدهد که مورد آزمون در حال پردازش اطلاعات است و این ERP بیشتر شناختی و درون‌زا شمرده می‌شود، بر خلاف دسته اول که حسی و برون‌زا دانسته می‌شوند.

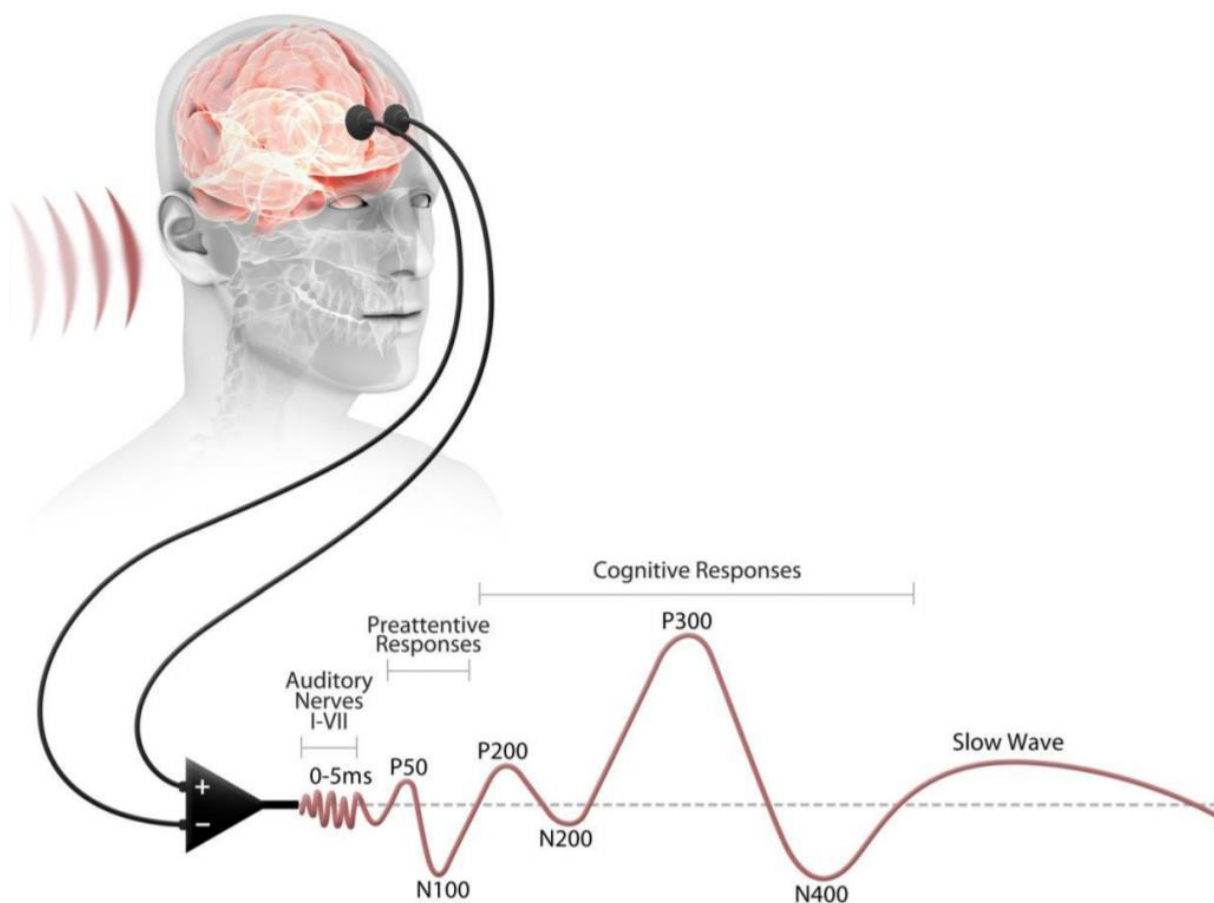
امواج مشاهده شده از ERP بر اساس اینکه بالاترین دامنه آن در محدوده مثبت یا منفی قرار می‌گیرد و بازه زمانی مشاهده آن نام‌گذاری می‌شوند.

مثلاً در مشخصه N100 موج منفی در ۱۰۰ میلی ثانیه پس از تحریک مشاهده می‌شود.

مشخصه‌های مختلف EPR به طور خلاصه:

- **موج P50:** در این موج بیشترین قله مثبت در بازه ۴۰ تا ۷۵ میلی ثانیه پس از تحریک رخ می‌دهد.
- **موج N100 یا N1:** بزرگترین قله منفی در بازه ۹۰ تا ۲۰۰ میلی ثانیه پس از یک محرک غیر منتظره مشاهده می‌شود و عموماً این پاسخ عمدتاً یک پاسخ فرآیند تطبیقی دانسته می‌شود، به این مفهوم که زمانی که یک محرک ارائه می‌گردد با محرک از قبل تجربه شده تطبیق داده می‌شود. این مولفه بیشترین دامنه را در Cz دارد که تحت تاثیر محرکات بینایی است. اهمیت این موج در بررسی توجه فضایی می‌باشد و به هنگام خواب نیز این مشخصه قابل مشاهده است.
- **موج N170:** این مولفه مرتبط با پردازش چهره می‌باشد و در مقایسه با سایر محرک‌های بینایی این موج زمانی مشاهده می‌شود که فرد با چهره یا بخشی از آن مواجه می‌شود و در بازه ۱۷۰ تا ۲۰۰ میلی ثانیه پس از ارائه محرک مشاهده می‌شود.
- **موج N200:** یک موج با قطبیت منفی در بازه حدود ۲۰۰ تا ۳۵۰ میلی ثانیه است که دارای سه مولفه عمده می‌باشد:
- **MMN (N2a):** یک مولفه منفی که به وسیله هر تغییر قابل تشخیصی در یک پس زمینه تکراری از تحریک شنیداری ایجاد می‌شود.
- **N2b:** این مولفه کمی پس از مولفه بالا در زمانی که تغییرات در ویژگی‌های فیزیکی محرک، هدف و تکلیف وجود دارد ظاهر می‌شود.
- **N2c:** این مولفه زمانی که طبقه بندی محرک‌های مختلف مد نظر باشد پدید می‌آید.

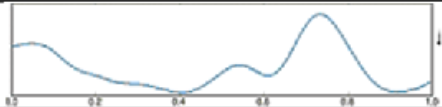
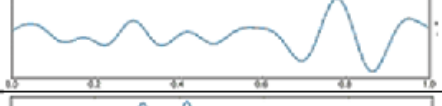
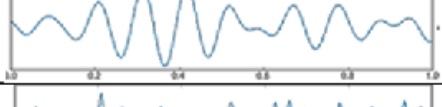
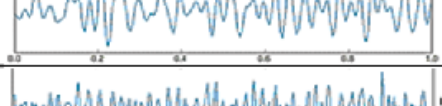

- **موج P200 یا P2:** این موج با قطبیت مثبت در بازه ۱۰۰ تا ۲۵۰ میلی ثانیه پس از تحریک مشاهده شده و توسط تکالیف شناختی ختلف از جمله پردازش ادراکی سطح بالا ایجاد می‌شود.
- **موج P300 یا P3:** این مولفه پس از کشف در ۱۹۶۵ بخش عمده‌ای از تحقیقات در زمینه ERP را به خود اختصاص داد. برای محرک‌های شنیداری در بزرگسالان، در بازه زمانی ۲۵۰ تا ۴۰۰ میلی ثانیه در نظر گرفته می‌شود. دامنه این موج عمدتاً به گونه‌ای منعکس کننده اطلاعات محرک است که هر چه توجه بیشتر باشد امواج P3 بزرگتری نیز تولید می‌شود.
کاهش دامنه این موج شاخص آسیب پذیر گسترده نورویولوژیکی است.
- **موج N400:** این موج منفی مرتبط با عدم تجانس معنایی در بازه ۳۰۰ تا ۶۰۰ میلی ثانیه پس از تحریک شناخته شد. این موج به طور معکوس با پیشبینی یک کلمه داده شده برای پایان دادن به جمله مرتبط است.
- **موج P600:** این موج در پردازش‌های زبانی در جملاتی که شامل اشکال نحوی، ساختار بیانی نامناسب و یا داری ساختار بیانی پیچیده باشند ظاهر می‌شود.
- علاوه بر این‌ها مشخصه‌های دیگری همچون ERN، LPC، N200 و N300 از سنجش EPRها قابل استخراج است.



2) باندهای فرکانسی در سیگنال‌های EEG:

- **باند دلتا:** این باند فرکانس‌های ۰/۵ تا ۴ هرتز را در بر می‌گیرد، دامنه‌های نسبتاً بالایی داشته و از نظر فعالیت، این موج در مرحله سوم خواب شروع به پدیدار شدن می‌کند و در مرحله چهار موج غالب امواج مغزی می‌شود. در همه جانوران دیده می‌شود و در اجناس مذکر و مونث و همچنین غالبیت نیم کره‌ای دارای تفاوت است.
- **باند تتا:** ۴ تا ۸ هرتز را در بر گرفته و در انسان وابسته به خواب REM است و با تغییر وضعیت از خواب به بیداری در بازه‌هایی کوتاه (کمتر از یک ثانیه) قادر به مشاهده تتا خواهیم بود. تتا با منشا قشری نیز در حالت گذر از خواب به بیداری و نیز در بیداری کامل دیده می‌شود. تتا با خلاقیت و تکانشگری در ارتباط است. با حواس‌پرری و عدم توجه، خیال پردازی و افسردگی و اضطراب افزایش می‌یابد و در کودکان بیشتر از بزرگسالان دیده می‌شود. همچنین این باند در اختلالاتی هم چون نقص توجه و بیش‌فعالی و نیز افسردگی تغییر می‌کند.
- **باند آلفا:** این باند شامل فرکانس‌های ۸ تا ۱۲ هرتز می‌باشد. در حالت ریلکس در بیداری با چشمان بسته عمده‌تاً از لوب پس سری منشاء گرفته و با باز کردن چشم‌ها و یا خستگی و خواب کاهش می‌یابد. امواج آلفا مهار فعالانه و کافی برای مسیرهای حسی غیر مرتبط ایجاد می‌کند و به نوعی دروازه حسی تالاموس به قشر مخ را می‌بندد. در حالت چشم بسته قوی‌ترین امواج مغزی در لوب پس سری هستند. اگر دامنه آلفا در حالت چشم بسته کم شود، نشانه خواب آلودگی و افزایش آلفا به طور کلی احتمال وقوع اشتباهات را زیاد می‌کنند. در اختلالات فوبیا، بیش‌فعالی، لکنت زبان و افسردگی، موج آلفا دچار تغییر می‌شود.
- **باند بتا:** این باند که فرکانس‌های ۱۲ تا ۳۰ هرتز را در بر می‌گیرد با حالت هوشیاری در زمان بیداری مطابق بوده و به سه دسته بتا ۱، ۲ و ۳ بر اساس فرکانس تقسیم می‌شود. بتا ۱ (۱۲،۵ - ۱۶ هرتز) با حالت تمرکز فعال یا تفکر اضطرابی منطبق است. امواج بتا در سرتاسر قشر حرکتی وابسته به انقباض عضلات در حرکات ایزوتونیک است و قبل و در طی حرکت کاهش می‌یابد. دوره‌های فعالیت بتا وابسته با تقویت فیدبک‌های حسی در کنترل حرکات ثابت است و وقتی حرکت تغییر می‌کند، کاهش می‌یابد. فعالیت بتا وقتی که باید حرکت حفظ شود / ادامه یابد و یا ارادی کاهش یابد، افزایش می‌یابد. در اعتیاد، خشم، افسردگی، اضطراب، نقص توجه، گرفتگی عضلانی و پارانویا، امواج بتا تغییر می‌کند.
- **باند گاما:** این باند در فرکانس حدود ۲۵ - ۱۰۰ هرتز (معمولاً ۴۰ هرتز) به نظر می‌رسد، که در واحد سازی درک خودآگاه و آگاهی binding problem نقش دارد. همچنین در طی خواب REM و بیهوشی نیز دیده می‌شود، که مطابق با تصویرسازی است. امواج گاما در آلزایمر، شیزوفرنی، پارکینسون و صرع دچار تغییر می‌گردد.

- **باند مو:** آمیخته‌ای از امواج آلفا با فرکانس حدود ۷,۵-۱۲,۵ هستند، که در قشر حرکتی یافت می‌شوند و با حرکت، قصد حرکت و یا تصور حرکت کاهش می‌یابند و وقتی بدن در حالت استراحت است شاخص می‌باشد. احتمالاً ناشی از فعالیت الکتریکی هم زمان نوروهای هری قشر کنترل کننده حرکات ارادی هستند. برخلاف امواج آلفا که در لوب پس سری غالب‌اند، این باند در فاصله گوش تا گوش در ناحیه قشر حرکتی بیشتر دیده می‌شود.

Waves	Frequency bands (Hz)	Behaviour Trait	Signal Waveform
Delta	0.3 – 4	Deep sleep	
Theta	4 – 8	Deep Meditation	
Alpha	8 – 13	Eyes closed, awake	
Beta	13 – 30	Eyes opened, thinking	
Gamma	30 and above	Unifying consciousness	

3) فرکانس نمونه برداری مناسب برای سیگنال‌های EEG:

- در گذشته برای پردازش سیگنال‌های مغزی ماکسیمم فرکانس تحرکات نورنی را در حدود ۱۳۰ الی ۱۴۰ هرتز در نظر می‌گرفتند. بعدها بهره‌گیری از فرکانس‌های بالاتر حتی تا حدود ۴۸۰ الی ۵۲۰ هرتز برای برخی پردازش‌های جزئی‌تر نیز مورد استفاده قرار گرفت. در این مورد برای این پروژه می‌توان حد آستانه فرکانسی را همان مقدار ۱۳۰ هرتز در نظر گرفت که با توجه به شرط نایکسونیست نیازمند فرکانس نمونه برداری در حدود ۲۶۰ هرتز می‌باشد.
- در اندازه‌گیری‌های عملیاتی معمولاً فرکانس نمونه برداری تجهیزات و تقویت‌کننده‌ها ۵۱۲/۵۰۰ هرتز، ۱۰۲۴/۱۰۰۰ هرتز یا ۲۵۶/۲۵۰ هرتز می‌باشد که با توجه به ملزومات این پروژه فرکانس ۲۵۶ هرتز مقدار بهینه‌ای برای حد آستانه تحلیل طیف فرکانسی می‌باشد.

4) محاسبه فرکانس نمونه برداری:

از سطر اول فاصله زمانی نمونه برداری میان دو نمونه متوالی را به دست می‌آوریم، این مقدار دوره تناوب نمونه برداری می‌باشد که با محاسبه مقدار عکس آن فرکانس نمونه برداری به دست می‌آید.

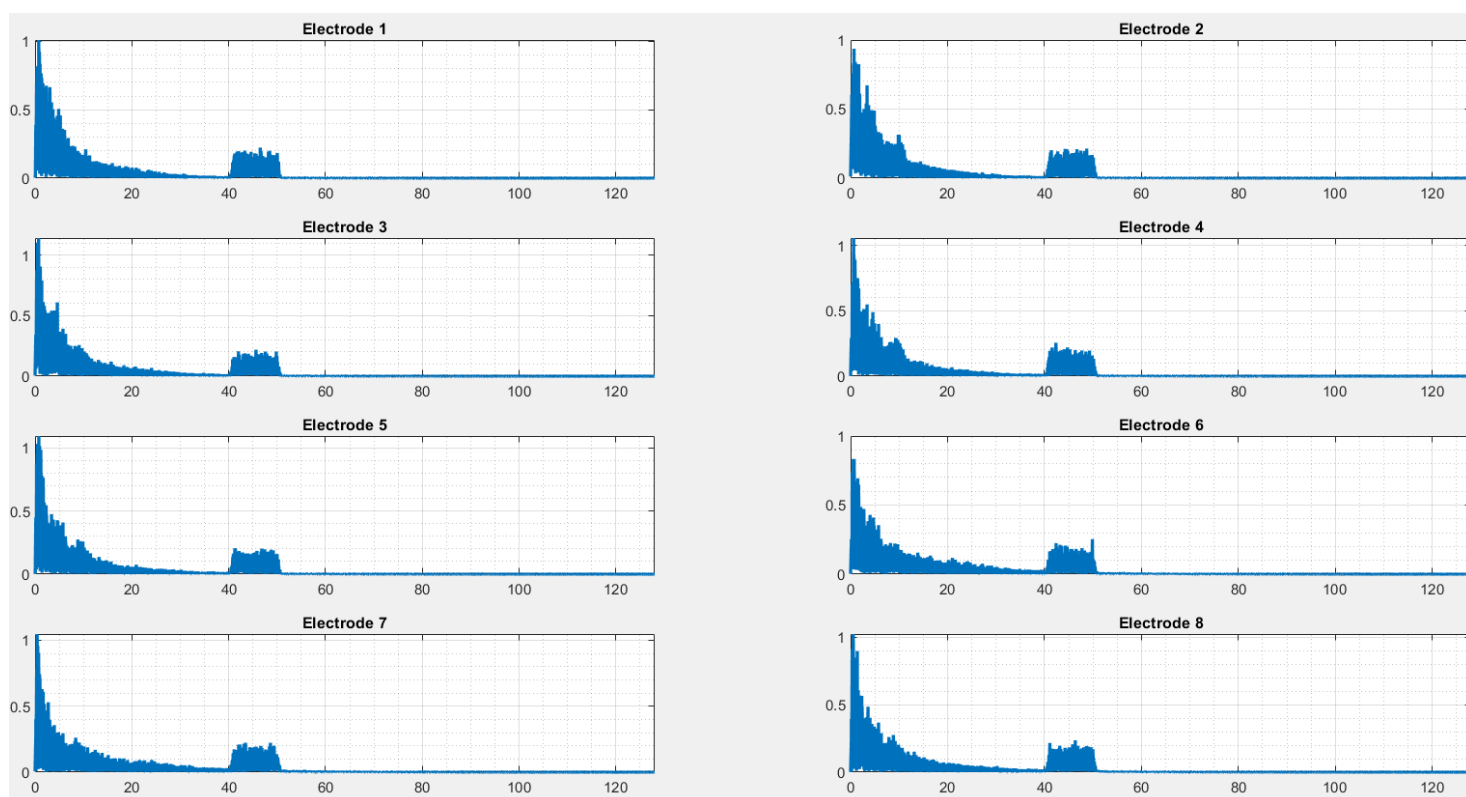
- مقادیر به دست آمده در این بخش $F_s = 256 \text{ Hz}$ و $T_s = 0.0039 \text{ s}$ بودند.

5 محاسبه فرکانس‌های قطع فیلتر میان‌گذر:

باتوجه به اطلاعاتی که در ابتدای این بخش مطرح گردید مشخصه‌های مورد استفاده از فرکانس ۰/۵ هرتز شروع شده و تا فرکانس‌های حدود ۱۳۰ الی ۱۴۰ هرتز مقدار دارند؛ اما در این میان در فرکانس‌های پایین و همچنین حول فرکانس ۵۰ هرتز انتظار می‌رود با نویزهای قابل توجهی مواجه باشیم.

6 طیف فرکانسی کانال‌ها:

با استفاده از تابع HalfBandFFT که در بخش قبل پیاده سازی شد و با اعمال تغییراتی جزئی در آن، طیف فرکانسی برای هر کانال در هر یک از هشت سطر ماتریس از فرکانس صفر تا ۱۳۰ هرتز به شکل زیر حاصل شد:



همانطور که مشهود است در اطراف فرکانس‌های ۵۰ هرتز و صفر هرتز با مقادیر بزرگتری مواجه هستیم که بخش قابل توجهی از آن ناشی از نویز تاثیر گذار بر نمونه برداری است.

با توجه این مسئله و لحاظ کردن این مشاهده که مقدار سیگنال اصلی از حدود ۴۰ هرتز به مقادیر بسیار ناچیزی میل میکند می‌توان فرکانس قطع حد بالای فیلتر را مقدار ۴۰ هرتز و حد پایین آن را در حدود ۰/۵ هرتز در نظر گرفت.

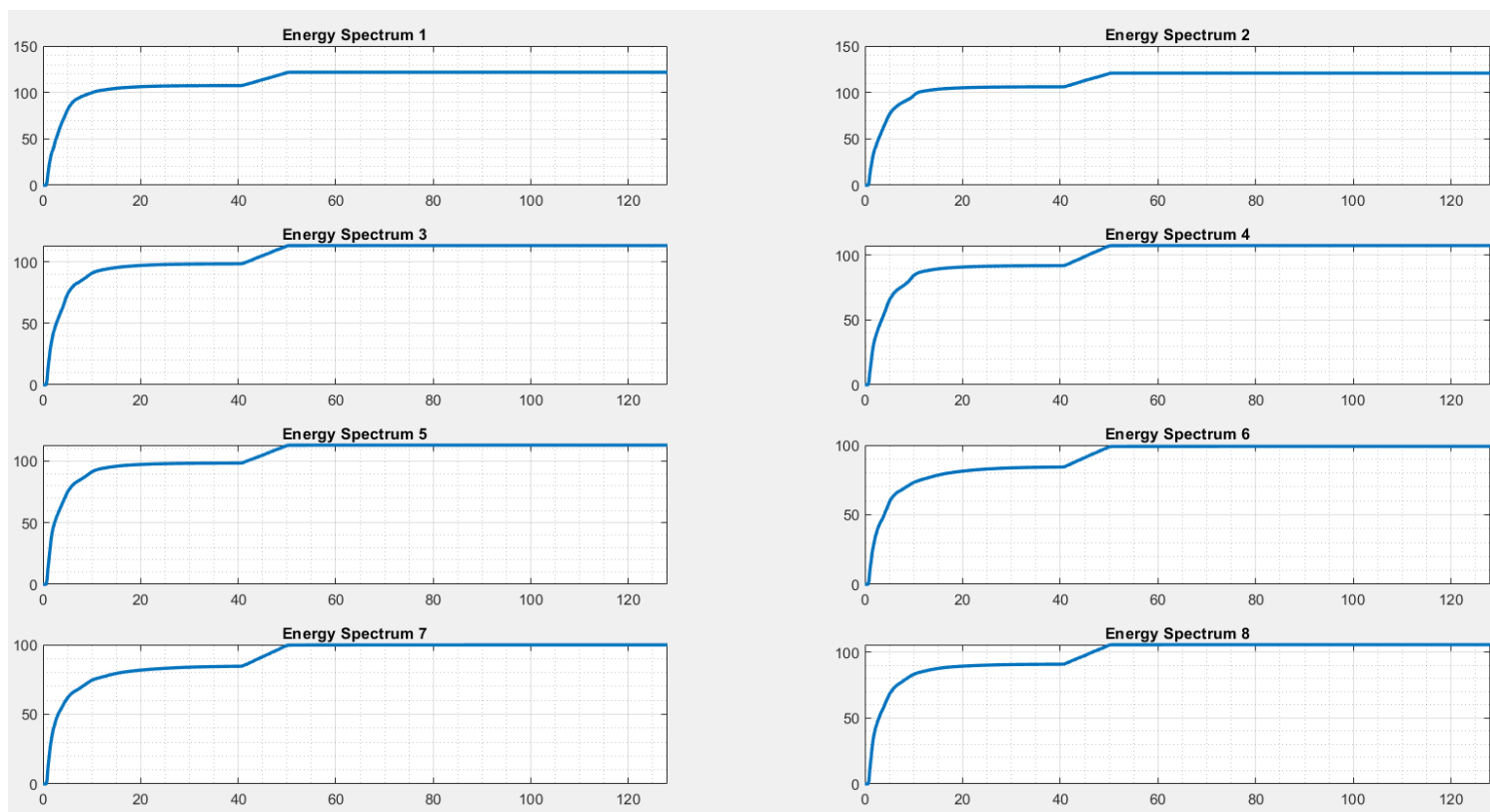
7) محاسبه فرکانس قطع با توجه به انرژی باندهای فرکانسی:

می‌دانیم انرژی یک سیگنال گسسته با بهره‌گیری از رابطه زیر به دست می‌آید:

$$E_s = \langle x(n), x(n) \rangle = \sum_{n=-\infty}^{\infty} |x(n)|^2$$

در این قسمت نیز برای محاسبه طیف فرکانسی انرژی سیگنال از فرم زمانی سیگنال تبدیل فوریه (fft) گرفته و مقدار انباشت انرژی تبدیل فوریه به دست آمده برای هر فرکانس، در یک حلقه به شکل رابطه مذکور محاسبه شده است. همچنین برای حذف مقادیر بسیار زیاد انرژی فرکانس‌های نزدیک به صفر، محاسبات انرژی از اندیس‌هایی کمی بزرگتر از صفر آغاز شده است.

در نهایت نمودار انباشت انرژی سیگنال در حوزه فرکانس برای هر کدام از کانال‌ها به شکل زیر به دست آمد:



در این نمودارها، مقدار متناظر با هر فرکانس نشان‌دهنده مجموع چگالی انرژی تمامی فرکانس‌های کوچک‌تر است.

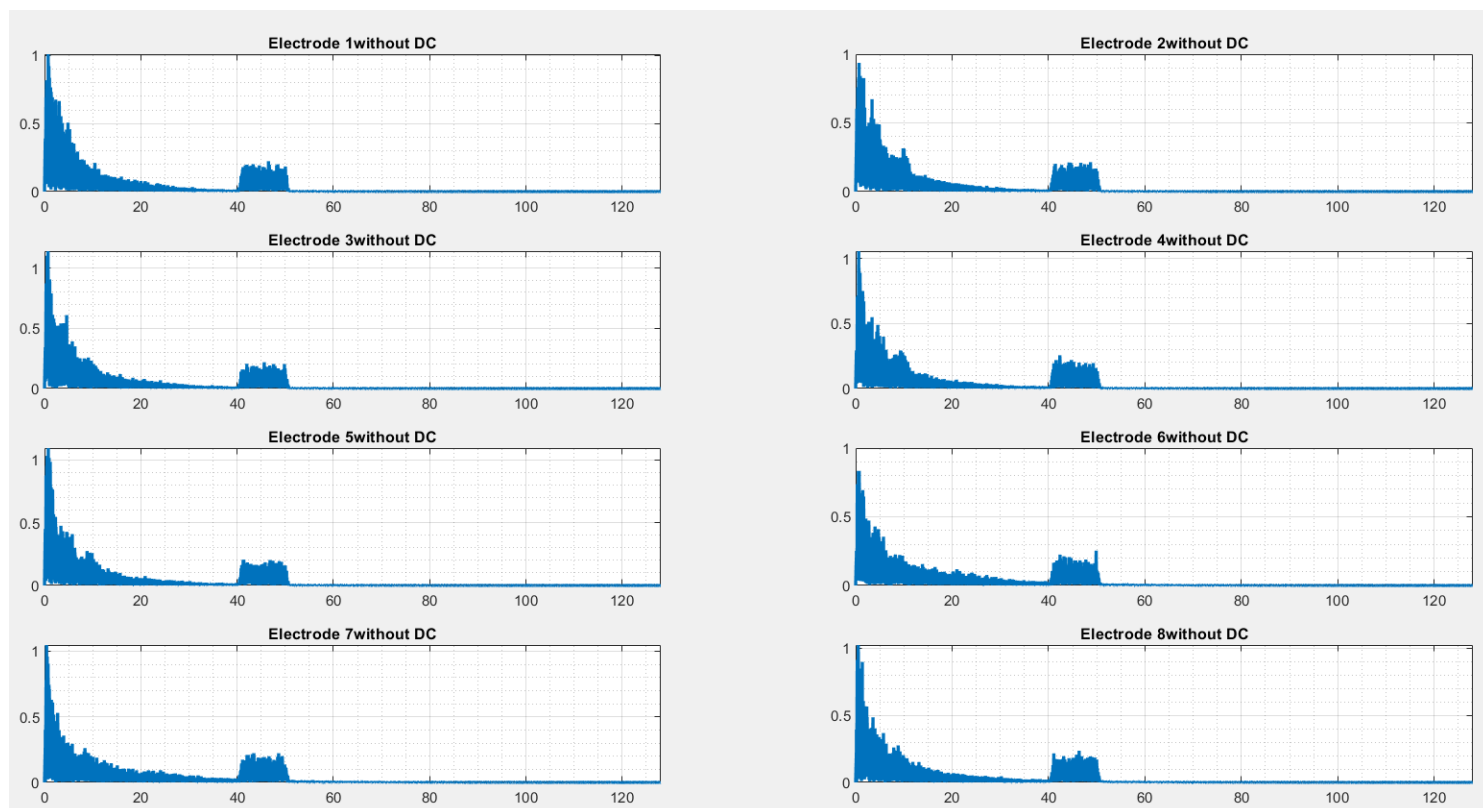
همانطور که مشهود است در بازه میان ۲۰ تا ۴۰ هرتز و همچنین بازه پس از ۵۰ هرتز نمودار به شکل یک خط صاف در آمده است؛ این مشاهده به این معنا است که در این بازه‌ها سیگنال نمونه برداری شده اصلی مقداری بسیار کوچک داشته و افزایش انرژی‌ای که در بازه اطراف ۵۰ هرتز مشاهده می‌کنیم ناشی از نویز محیط است.

برای حذف این نویز می‌بایست فرکانس قطع حد بالای فیلتر را در حدود ۴۰ هرتز در نظر بگیریم.

8 حذف مقادیر DC و اعمال فیلتر:

برای حذف مقادیر میانگین از هر سطح، با استفاده از تابع mean میانگین هر ستون از ترانزاده ماتریس کانال‌ها را به دست می‌آوریم که برابر است با مقادیر میانگین برای هر کانال؛ سپس با استفاده از تابع meshgrid ماتریسی از میانگین‌ها به دست می‌آوریم که در هر سطر مقدار میانگین سیگنال متناظر با کانال آن سطر را دارد و مقادیر همه ستون‌ها با یکدیگر برابر هستند.

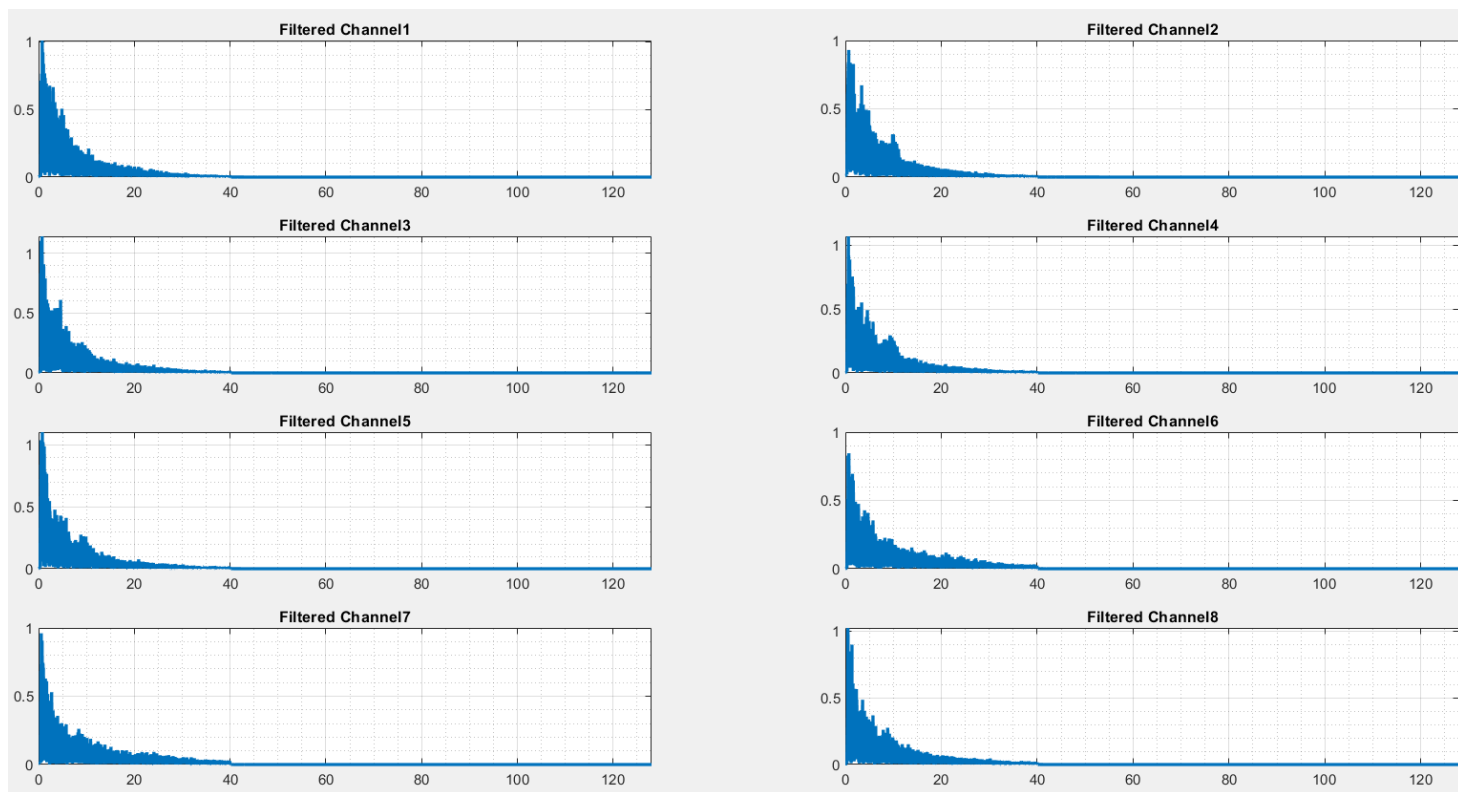
ماتریس به دست آمده را از ماتریس کانال‌ها کم می‌کنیم تا مقادیر میانگین از سیگنال هر کدام از کانال‌ها کم شود. در پایان این مراحل طیف فرکانسی ماتریس خروجی برای هر کانال به شکل زیر به دست می‌آید:



با کمی بزرگنمایی در فرکانس‌های نزدیک به صفر می‌توان مشاهده کرد که مقدار DC سیگنال (مقادیر متناظر با فرکانس دقیقاً صفر) بسیار ناچیز شده است، اما از آنجا که می‌دانیم نویزهای DC با کمی پراکندگی در نزدیکی فرکانس صفر اعمال شده‌اند، حذف میانگین به تنهایی کارساز نخواهد بود.

از این رو حد پایین فیلتر پایین گذر را فرکانس ۰/۵ هرتز قرار می‌دهیم و مقادیر ما به ازای فرکانس‌های کوچکتر از ۰/۵ سیگنال را کم می‌کنیم.

برای فیلتر کردن سیگنال نیز از تابع bandpass متلب برای بازه ۰/۵ تا ۴۰ هرتز روی هرکدام از کانال‌ها مستقلاً استفاده شد و طیف فرکانسی خروجی به شکل زیر به دست آمد:



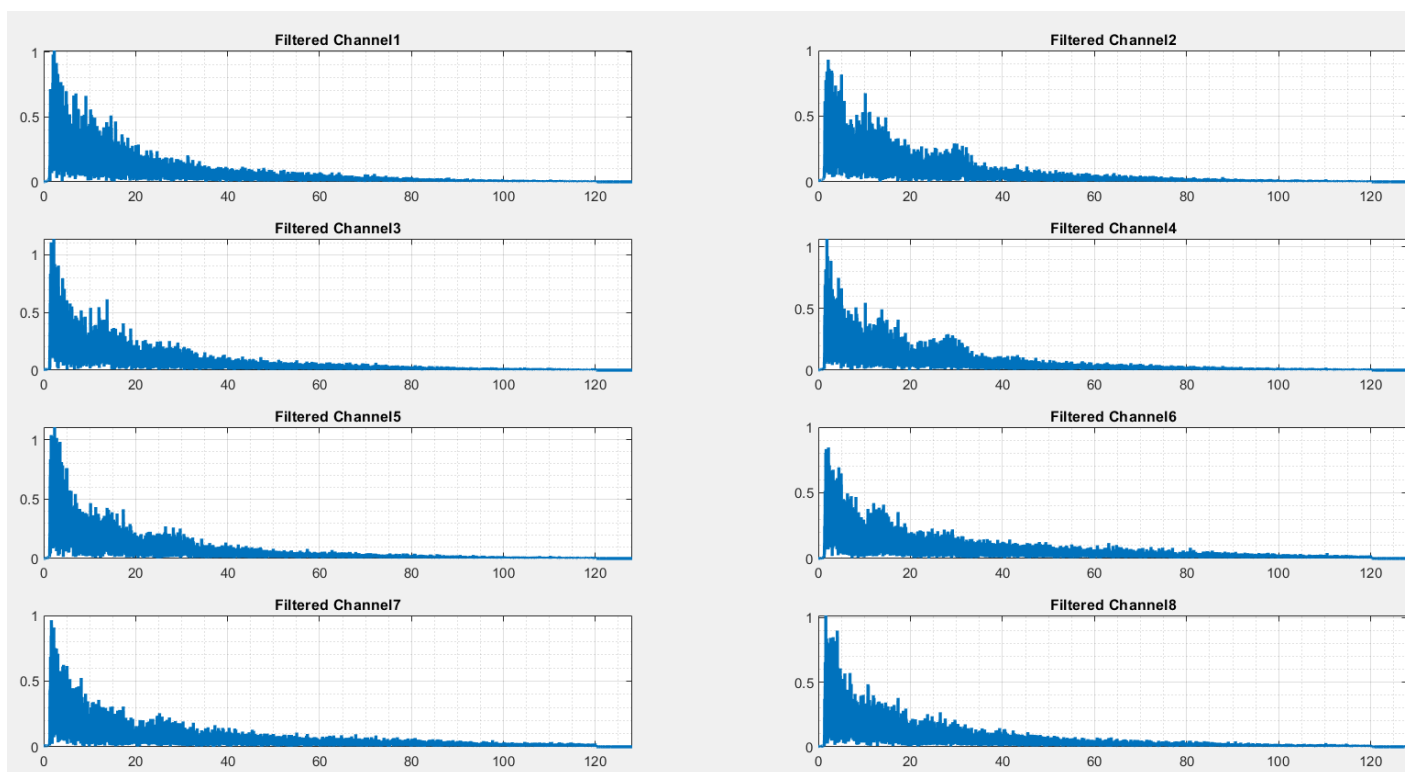
9) کاهش فرکانس نمونه برداری:

با توجه به فرکانس قطع فیلتر، می‌دانیم مقادیر ما به ازای فرکانس‌های بزرگ‌تر از ۴۰ هرتز تقریباً صفر هستند. بنابراین با لحاظ کردن شرط نایکوئیست می‌دانیم که فرکانس نمونه برداری می‌تواند تا حدود دو برابر این مقدار، یعنی حدود ۸۰ هرتز پایین بیاید.

برای کاهش فرکانس نمونه برداری از ۲۵۶ هرتز تا نزدیک‌ترین مقدار به آستانه ۸۰ هرتزی می‌توان با نرخ ۳ برابری سیگنال موجود را down sample کنیم. در این حال، فرکانس نمونه برداری به مقدار حدود ۸۵ هرتز می‌رسد که شرط نایکوئیست را نیز همچنان رعایت می‌کند.

برای کاهش سه برابری فرکانس نمونه برداری، از هر سه نمونه متوالی یک نمونه را نگه داشته و باقی را حذف می‌کنیم، این کار را برای کل ماتریس داده‌ها (نه فقط داده‌های کانال‌ها) انجام می‌دهیم.

با توجه به رابطه مذکور در بخش یک برای down sampling با کاهش نرخ نمونه برداری انتظار داریم نمودارهای طیف فرکانسی در راستای افقی انبساط پیدا کنند که این امر در نمودارها نیز مشهود است:



Epoching (10

در این قسمت برای پیدا کردن اندیس‌های شروع تسک‌ها (StimuliOnset) مطابق مشروحات دستورالعمل پروژه، با در نظر گرفتن سطر دهم ماتریس down sample شده داده‌ها، اندیس مقادیر غیر صفر را در بردار StimuliOnset ثبت می‌کنیم.

چالشی که در این قسمت وجود دارد این است که در برخی نقاط، با چند درایه غیر صفر متوالی مواجه هستیم؛ برای آن‌که این اندیس‌های متوالی ثبت نشوند تنها اندیس درایه‌های غیر صفری را در نظر می‌گیریم که پس از یک درایه صفر قرار گرفته باشند.

در ادامه تابع epoching به فرم مطروحه در دستورالعمل در انتهای فایل کدها پیاده‌سازی شده است. در این تابع ابتدا فرکانس نمونه برداری جدید را بر اساس سطر اول ماتریس داده‌های ورودی تابع محاسبه می‌کنیم و سپس با لحاظ کردن فرکانس به دست آمده، تعداد داده‌های متناظر با بازه‌های زمانی BackwardSamples و ForwardSamples را محاسبه می‌کنیم و به این تعداد از هر اندیس موجود در بردار StimuliOnset عقب‌تر و جلوتر رفته و اندیس‌های شروع و پایان هر epoch را به دست می‌آوریم. حال با استفاده از بازه‌های به دست آمده epochها را از ماتریس داده‌ها جدا کرده و در ماتریس epoched می‌چینیم تا به خروجی سه بعدی مطلوب دست پیدا کنیم. ابعاد ماتریس خروجی:

 epoched 8x2700x85 double

همانطور که انتظار میرفت هر epoch یک بازه ۸۵ درایه‌ای است که با لحاظ کردن فرکانس نمونه‌برداری می‌دانیم این تعداد متناظر است با یک بازه یک ثانیه‌ای در حوزه زمان. (۸۰۰+۲۰۰ میلی‌ثانیه)

11) علت فیلتر کردن پیش از کاهش فرکانس نمونه برداری:

می‌دانیم با کاهش فرکانس نمونه برداری سیگنال دچار انبساط عرضی در حوزه فرکانس می‌شود و از این رو با علم به آنکه سیگنال فیلتر نشده تا فرکانس‌هایی در حدود ۶۰ هرتز دارای مقادیر قابل توجه است، می‌دانیم با کاهش سه برابری نرخ نمونه برداری شرط نایکوئیست برقرار نبوده و دچار aliasing خواهیم شد. در واقع اگر پیش از فیلتر کردن سیگنال بخواهیم نرخ نمونه برداری را کاهش دهیم، این فرکانس حداکثر تا مقدار $2 \times 60 = 120$ هرتز می‌تواند پایین بیاید که متناظر با down sampling مرتبه دو است. با این حال اگر عملیات فیلترینگ را پیش از کاهش نرخ نمونه‌برداری انجام دهیم این چالش مرتفع خواهد بود.

12) تفاوت توالی epoch نسبت به فیلترینگ:

همانطور که ذکر شد، از آنجا که طول هر پنجره یک ثانیه است و فرکانس نمونه برداری پس از کاهش به ۸۵ هرتز می‌رسد، متناظر با هر epoch ۸۵ درایه داریم که این مقدار کوچک‌تر از بازه ۱۰۰ تایی مطرح شده در دستور العمل است. از طرف دیگر می‌دانیم طول فیلتر می‌بایست کوچک‌تر از طول هر epoch باشد و در صورت برقرار نبودن این شرط با ارور متلب مواجه خواهیم شد. برای آزمودن این تفاوت، یک فیلتر میان‌گذر به طول ۸۵ در بازه فرکانس ۲۰ تا ۴۰ هرتز با استفاده از تابع fir1 طراحی شد و این فیلتر به دست آمده را با کمک تابع filter متلب بر روی هر کدام از سطور ماتریس داده‌ها اعمال کردیم. در این شرایط با توجه به امکان پذیر نبودن استفاده از فیلتر با طول ۱۰۰، طبعاً با خروجی متفاوتی مواجه خواهیم بود.

13) علت تاخیر در شروع نمونه‌برداری:

تزریق ژل برای کاهش امپدانس پوست سر معمولاً در همان ابتدا صورت می‌گیرد و مدتی طول میکشد تا این ژل بر روی سطح پوست گسترده شده و در میان الکترودها و عمقی از پوست قرار بگیرد. همچنین در ابتدای امر ممکن است با واکنش‌های ابتدایی نورون‌ها به علت تحریکات عصبی اولیه مواجه باشیم و اندکی شرایط نوپزی با ادامه آزمایش متفاوت باشد. از این رو لازم است تا مدت زمانی را صبر کرده تا اثرات شرایط اولیه از بین رفته و با حالت پایدارتر نمونه‌برداری را آغاز کنیم.

خوشه‌بندی بر مبنای همبستگی

• می‌دانیم:

$$(a - b)^2 \geq 0 \Rightarrow a^2 + b^2 \geq 2ab$$

از جایگزینی a و b به ترتیب با مقادیر $\sqrt{\int_{-\infty}^{\infty} X^2(t)dt}$ و $\sqrt{\int_{-\infty}^{\infty} Y^2(t)dt}$ خواهیم داشت:

$$\int_{-\infty}^{\infty} X^2(t)dt + \int_{-\infty}^{\infty} Y^2(t)dt \geq 2 \sqrt{\int_{-\infty}^{\infty} X^2(t)dt \int_{-\infty}^{\infty} Y^2(t)dt}$$

همچنین با توجه به خاصیت خطی بودن انتگرال داریم:

$$\int_{-\infty}^{\infty} X^2(t)dt + \int_{-\infty}^{\infty} Y^2(t)dt = \int_{-\infty}^{\infty} (Y^2(t) + X^2(t))dt = \int_{-\infty}^{\infty} [(Y(t) - X(t))^2 + 2X(t)Y(t)]dt$$

در نتیجه:

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

$$\Rightarrow \int_{-\infty}^{\infty} X^2(t)dt + \int_{-\infty}^{\infty} Y^2(t)dt \geq 2 \int_{-\infty}^{\infty} X(t)Y(t)dt$$

$$\Rightarrow 2 \sqrt{\int_{-\infty}^{\infty} X^2(t)dt \int_{-\infty}^{\infty} Y^2(t)dt} \geq \int_{-\infty}^{\infty} X^2(t)dt + \int_{-\infty}^{\infty} Y^2(t)dt \geq 2 \int_{-\infty}^{\infty} X(t)Y(t)dt$$

$$\Rightarrow \sqrt{\int_{-\infty}^{\infty} X^2(t)dt \int_{-\infty}^{\infty} Y^2(t)dt} \geq \int_{-\infty}^{\infty} X(t)Y(t)dt$$

حال با توجه به بزرگتر بودن مقدار مخرج از صورت می‌توان نوشت:

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

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

$$\Rightarrow -1 \leq r_{XY} \leq 1$$

• حالتی که $|r_{XY}| = 1$ باشد:

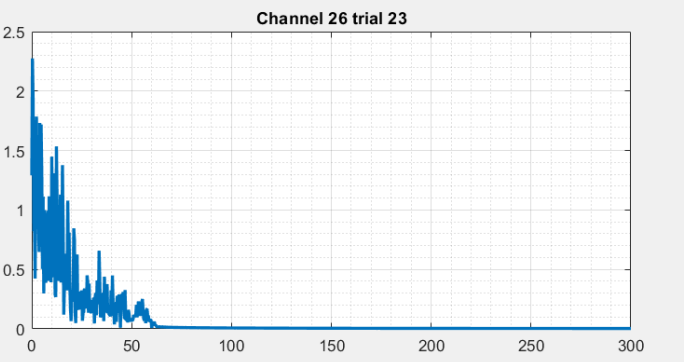
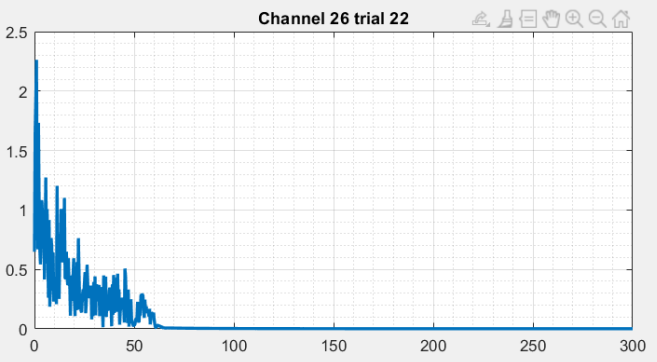
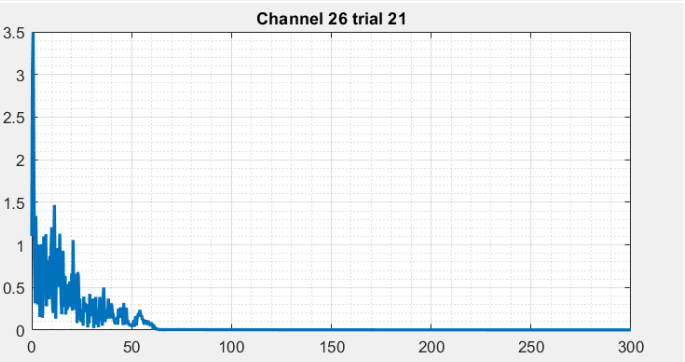
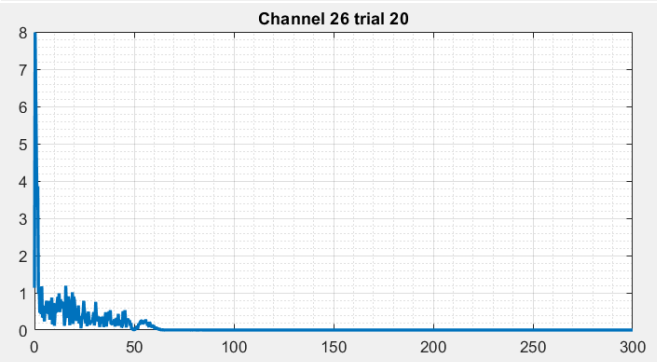
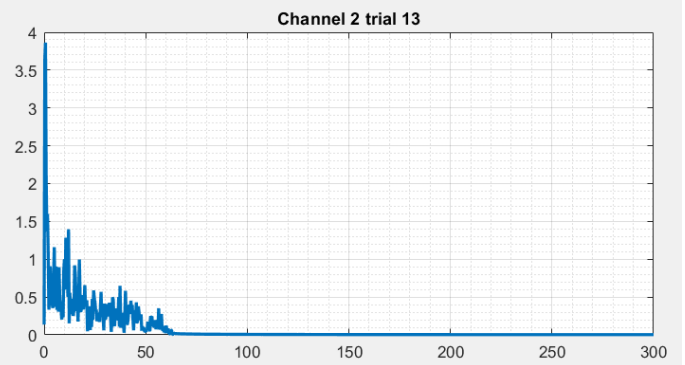
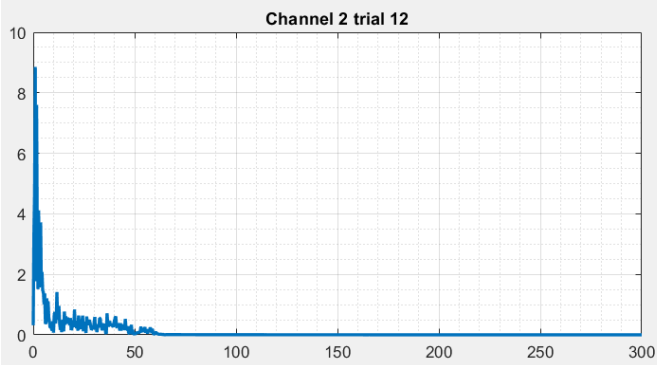
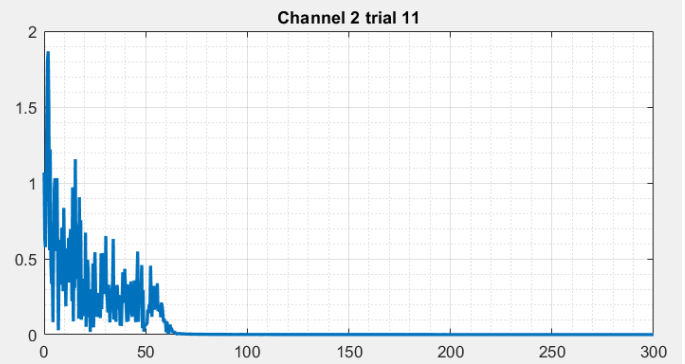
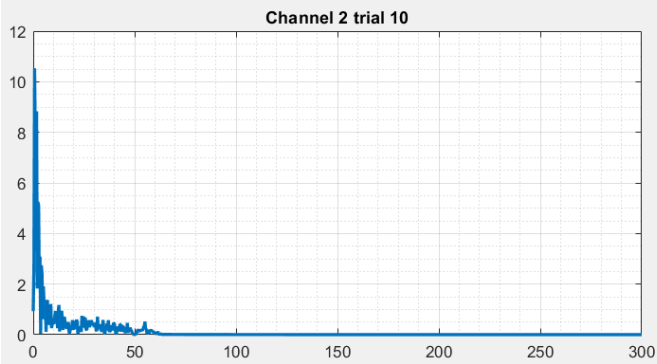
با توجه به مراحل طی شده در بالا می‌دانیم زمانی مقدار r_{XY} برابر با یک خواهد بود که صورت و مخرج عبارت با یکدیگر برابر باشند و این به معنای آن است که:

$$\begin{aligned} \pm \sqrt{\int_{-\infty}^{\infty} X^2(t) dt \int_{-\infty}^{\infty} Y^2(t) dt} &= \int_{-\infty}^{\infty} X(t)Y(t) dt \\ \Rightarrow \int_{-\infty}^{\infty} X^2(t) dt \int_{-\infty}^{\infty} Y^2(t) dt &= \left[\int_{-\infty}^{\infty} X(t)Y(t) dt \right]^2 \\ \Rightarrow \frac{\int_{-\infty}^{\infty} X^2(t) dt}{\int_{-\infty}^{\infty} X(t)Y(t) dt} &= \frac{\int_{-\infty}^{\infty} X(t)Y(t) dt}{\int_{-\infty}^{\infty} Y^2(t) dt} = \alpha \\ \Rightarrow \int_{-\infty}^{\infty} X^2(t) dt &= \alpha \int_{-\infty}^{\infty} X(t)Y(t) dt \\ \alpha \int_{-\infty}^{\infty} Y^2(t) dt &= \int_{-\infty}^{\infty} X(t)Y(t) dt \text{ و} \\ \Rightarrow \int_{-\infty}^{\infty} X^2(t) dt &= \alpha \left[\alpha \int_{-\infty}^{\infty} Y^2(t) dt \right] \\ \Rightarrow \int_{-\infty}^{\infty} X^2(t) dt &= \int_{-\infty}^{\infty} \alpha^2 Y^2(t) dt \\ \Rightarrow \int_{-\infty}^{\infty} [\alpha^2 Y^2(t) - X^2(t)] dt &= 0 \\ \Rightarrow [\alpha^2 Y^2(t) - X^2(t)] &= 0 \Rightarrow X(t) = \alpha Y(t) \end{aligned}$$

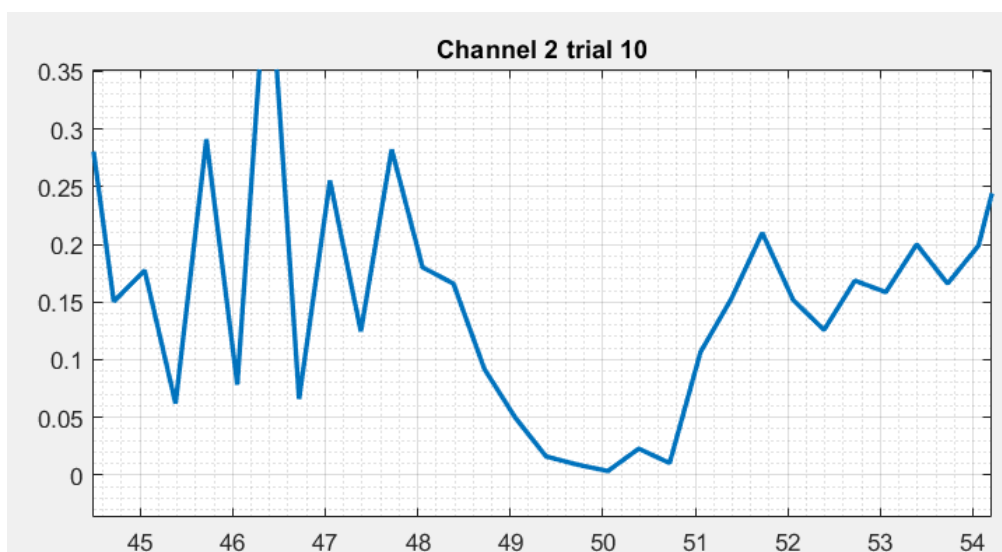
• معیار شباهت:

هر چه توزیع مقادیر دو سیگنال در حوزه زمان به یکدیگر شبیه‌تر باشد انتگرال ضرب دو سیگنال (صورت عبارت کسری) مقدار بیشتری پیدا کرده و در بیشینه حالت مضربی از $\int_{-\infty}^{\infty} X^2(t) dt$ خواهد بود. (حالتی که $X(t) = \alpha Y(t)$) در عین حال مخرج عبارت نیز تنها به انرژی دو سیگنال وابسته است و مستقل از شباهت دو سیگنال است؛ بنابراین هر چه انرژی ضرب دو سیگنال بیشتر باشد مقدار r_{XY} نیز بیشتر خواهد بود. از طرف دیگر می‌دانیم انرژی ضرب دو سیگنال زمانی بیشینه خواهد شد که توزیع مقادیر دو سیگنال بر یکدیگر منطبق شوند، بنابراین این پارامتر تاحدود بسیار خوبی وابسته به همبستگی و شباهت دو سیگنال خواهد بود.

- طیف فرکانسی مربوط به ۴ ترایال از دو کانال ۲ و ۲۶:



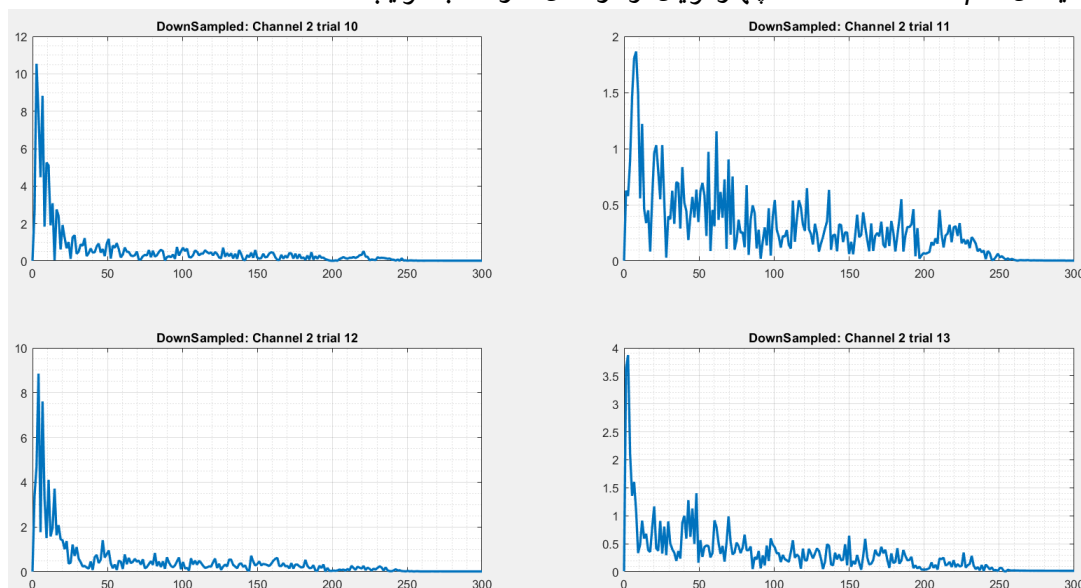
- با بزرگنمایی و پیمایش روی طیف فرکانسی کانال‌ها می‌توان دید که نویزهای حول فرکانس ۵۰ هرتز رفع شده اند و مقادیر در این ناحیه بسیار کوچک است:

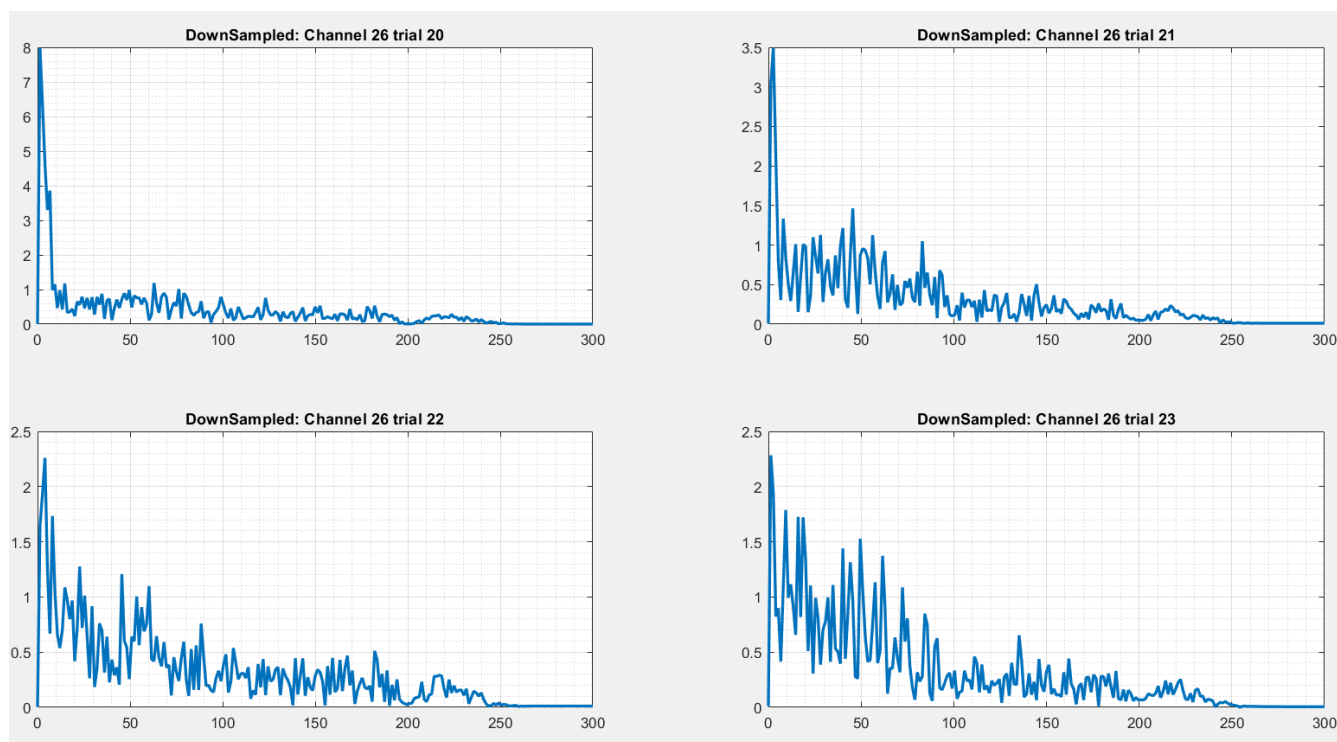


با این حال اما مقادیر ما به ازای فرکانس صفر غیر صفر هستند و نیاز است تا یک بار میانگین داده‌ها را از آن کم کنیم تا مقدار DC رفع شود. این کار را در دو حلقه تو در تو با پیمایش روی همه ترایال‌ها و با بهره‌گیری از تابع `detrend` متلب انجام می‌دهیم.

با توجه به طیف فرکانسی مشهود است که به ازای مقادیر بیش از حدود ۷۰ هرتز برای فرکانس، سیگنال مقدار نسبتاً ناچیزی دارد و با توجه به شرط نایکوئیست می‌توان فرکانس نمونه برداری را تو دو برابر این مقدار پایین آورد. برای این کار سیگنال موجود را یک بار با مرتبه چهار $down\ sample$ می‌کنیم تا فرکانس نمونه برداری به ۱۵۰ هرتز برسد.

سیگنال $down\ sample$ شده چهار ترایال از دو کانال ۲ و ۲۶ به ترتیب:





در این دو نمودار محور افقی (فرکانس) همان محور فرکانس نمودار طیف فرکانسی پیش از کاهش نرخ نمونه برداری است که برای نشان دادن انبساط عرضی سیگنال حفظ شده است.

- در گام بعدی برای محاسبه ماتریس کورولیشن لازم است تا دیتاهای *epoch* شده هر کانال را به طور متوالی در امتداد یکدیگر قرار دهیم و به طور یک‌جا کورولیشن‌های میان داده‌های هر دو کانال را محاسبه کنیم.
- در ادامه، تابع *CorrelationCalculator* را برای محاسبه کورولیشن میان دو سیگنال را در انتهای فایل کد پیاده سازی می‌کنیم. این تابع مطابق فرم گسسته ضریب پیرسون را که در دستورالعمل پروژه شرح داده شد به دست می‌آورد و در خروجی یک مقدار کوچک‌تر از یک را برمی‌گرداند.

$$\rho(a,b) = \frac{\sum_{i=1}^n (X_{a,i} - \bar{X}_a)(Y_{b,i} - \bar{Y}_b)}{\left\{ \sum_{i=1}^n (X_{a,i} - \bar{X}_a)^2 \sum_{j=1}^n (Y_{b,j} - \bar{Y}_b)^2 \right\}^{1/2}}$$

با استفاده از این تابع در دو حلقه تو در تو این مقدار را برای هر انتخاب دو تایی از کانال‌ها محاسبه می‌کنیم، از آنجا که ماتریس کورولیشن یک ماتریس متقارن است، برای کاهش حجم محاسبات این مقدار را تنها برای یک نیمه از آن به دست آورده و باقی درایه‌ها را با توجه به مقدار درایه قرینه‌شان مقداردهی می‌کنیم.

بخش از ماتریس کورولیشن:

63x63 double												
	1	2	3	4	5	6	7	8	9	10	11	12
1	1	0.9088	0.8861	0.9662	0.9568	0.7897	0.9270	0.9307	0.9179	0.6287	0.7390	0.8307
2	0.9088	1	0.9675	0.9297	0.9183	0.7823	0.8003	0.7618	0.7750	0.6281	0.6469	0.6432
3	0.8861	0.9675	1	0.8836	0.9302	0.6708	0.7367	0.7276	0.7736	0.7212	0.5447	0.5868
4	0.9662	0.9297	0.8836	1	0.9291	0.8650	0.9410	0.9049	0.8793	0.5723	0.7970	0.8263
5	0.9568	0.9183	0.9302	0.9291	1	0.7037	0.8541	0.8742	0.9071	0.7217	0.6425	0.7495
6	0.7897	0.7823	0.6708	0.8650	0.7037	1	0.8561	0.7507	0.6742	0.2789	0.8738	0.7363
7	0.9270	0.8003	0.7367	0.9410	0.8541	0.8561	1	0.9522	0.8901	0.4715	0.8737	0.9313
8	0.9307	0.7618	0.7276	0.9049	0.8742	0.7507	0.9522	1	0.9405	0.5221	0.7712	0.9284
9	0.9179	0.7750	0.7736	0.8793	0.9071	0.6742	0.8901	0.9405	1	0.5938	0.6875	0.8576
10	0.6287	0.6281	0.7212	0.5723	0.7217	0.2789	0.4715	0.5221	0.5938	1	0.2516	0.3939
11	0.7390	0.6469	0.5447	0.7970	0.6425	0.8738	0.8737	0.7712	0.6875	0.2516	1	0.8136
12	0.8307	0.6432	0.5868	0.8263	0.7495	0.7363	0.9313	0.9284	0.8576	0.3939	0.8136	1
13	0.8295	0.6350	0.6098	0.7981	0.7827	0.6526	0.8770	0.9289	0.9215	0.4501	0.7187	0.9182
14	0.6790	0.5833	0.6253	0.6318	0.7208	0.4239	0.6208	0.6764	0.7615	0.6083	0.4483	0.5988
15	0.5395	0.4648	0.3496	0.6070	0.4373	0.7933	0.6820	0.5803	0.4872	0.0898	0.7998	0.6389
16	0.7169	0.5740	0.4930	0.7519	0.6274	0.7659	0.8642	0.8022	0.7247	0.2739	0.8802	0.8870
17	0.7054	0.5307	0.4857	0.6954	0.6418	0.6277	0.8014	0.8164	0.7734	0.3320	0.7163	0.8821
18	0.6022	0.5205	0.5185	0.6642	0.6605	0.5471	0.7221	0.7826	0.8310	0.2770	0.6200	0.7886

• Clustering.

در این بخش برای محاسبه فاصله میان دو سیگنال کاندیدهای زیاد وجود دارد؛ [منبع ۱]، منبع [۲]

- فاصله مبتنی بر ضریب پیرسون:

$$d_{cor}(x, y) = 1 - \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 \sum_{i=1}^n (y_i - \bar{y})^2}}$$

- Eisen cosine correlation distance.

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

- Spearman correlation distance.

$$d_{spear}(x, y) = 1 - \frac{\sum_{i=1}^n (x'_i - \bar{x}')(y'_i - \bar{y}')}{\sqrt{\sum_{i=1}^n (x'_i - \bar{x}')^2 \sum_{i=1}^n (y'_i - \bar{y}')^2}}$$

- Kendall correlation distance

$$d_{kend}(x, y) = 1 - \frac{n_c - n_d}{\frac{1}{2}n(n-1)}$$

برای الگوریتم به روز رسانی ماتریس فاصله‌ها نیز چهار کاندیدا داریم که از این میان دو مورد اول پیاده سازی شده اند:

WPGMA -

UPGMA-

Complete-linkage clustering-

Single-linkage clustering-

کلاسترینگ در تابع $CorrelationCluster(InputCorrMat, DistanceMeasure, chNum, disType)$ پیاده سازی شده است که ورودی‌های آن به ترتیب از چپ به راست، ماتریس کورولیشن، حداقل حداقل فاصله میان دو کلاستر، تعداد کانال‌ها و در آخر نوع الگوریتم محاسبه فاصله است که به صورت متن داده می‌شود.

برای کلاسترینگ نیز از الگوریتم مشروح در [این منبع](#) بهره‌گیری شد که توضیحات کلی را می‌توان با مراجعه به آن یافت.

در تابع کلاسترینگ برای شروع، در ایتريشن صفر نیاز به یک ماتریس فاصله اولیه داریم که آن را با رابطه فاصله مبتنی بر ضریب پیرسون محاسبه می‌کنیم.

در قدم بعدی با علم به آنکه همه مقادیر فاصله‌ها میان صفر و یک قرار دارند، مقادیر روی قطر را (که صفر هستند) با ۹ جمع می‌کنیم تا هنگامی که در ادامه می‌خواهیم کوچک‌ترین فاصله را پیدا کنیم، این مقادیر صفر در محاسبات اختلالی ایجاد نکند.

در یک حلقه شرطی با بررسی آنکه حداقل فاصله موجود در ماتریس فاصله‌ها از مقدار حداقلی که در ورودی به تابع داده شده بیشتر است یا خیر تا زمانی که این فاصله از فاصله حداقلی ورودی تابع بیشتر شود محاسبات را ادامه می‌دهیم.

در داخل این حلقه، در هر گام دو ستون متناظر با درایه‌ای که کمترین مقدار متناظر را در ماتریس فاصله دارد را با یکدیگر ادغام می‌کنیم و در یک کلاستر قرار می‌دهیم. سپس با توجه به الگوریتم انتخاب شده برای تجدید ماتریس فاصله‌ها، سطر و ستون متناظر با کلاستر تازه تاسیس را با مقادیر مجموع وزن‌دار یا بدون وزن فاصله از دو کلاستر پیش از ادغام جایگزین می‌کنیم.

در نهایت خروجی تابع برداری خواهد بود که هر درایه آن شامل شماره کلاستر کانال متناظر با اندیس آن درایه است.

بخشی از مقدار اولیه ماتریس فاصله‌ها:

63x63 double														
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1	9	0.0912	0.1139	0.0338	0.0432	0.2103	0.0730	0.0693	0.0821	0.3713	0.2610	0.1693	0.1705	0.3210
2	0.0912	9	0.0325	0.0703	0.0817	0.2177	0.1997	0.2382	0.2250	0.3719	0.3531	0.3568	0.3650	0.4167
3	0.1139	0.0325	9	0.1164	0.0698	0.3292	0.2633	0.2724	0.2264	0.2788	0.4553	0.4132	0.3902	0.3747
4	0.0338	0.0703	0.1164	9	0.0709	0.1350	0.0590	0.0951	0.1207	0.4277	0.2030	0.1737	0.2019	0.3682
5	0.0432	0.0817	0.0698	0.0709	9	0.2963	0.1459	0.1258	0.0929	0.2783	0.3575	0.2505	0.2173	0.2792
6	0.2103	0.2177	0.3292	0.1350	0.2963	9	0.1439	0.2493	0.3258	0.7211	0.1262	0.2637	0.3474	0.5761
7	0.0730	0.1997	0.2633	0.0590	0.1459	0.1439	9	0.0478	0.1099	0.5285	0.1263	0.0687	0.1230	0.3792
8	0.0693	0.2382	0.2724	0.0951	0.1258	0.2493	0.0478	9	0.0595	0.4779	0.2288	0.0716	0.0711	0.3236
9	0.0821	0.2250	0.2264	0.1207	0.0929	0.3258	0.1099	0.0595	9	0.4062	0.3125	0.1424	0.0785	0.2385
10	0.3713	0.3719	0.2788	0.4277	0.2783	0.7211	0.5285	0.4779	0.4062	9	0.7484	0.6061	0.5499	0.3917
11	0.2610	0.3531	0.4553	0.2030	0.3575	0.1262	0.1263	0.2288	0.3125	0.7484	9	0.1864	0.2813	0.5517
12	0.1693	0.3568	0.4132	0.1737	0.2505	0.2637	0.0687	0.0716	0.1424	0.6061	0.1864	9	0.0818	0.4012
13	0.1705	0.3650	0.3902	0.2019	0.2173	0.3474	0.1230	0.0711	0.0785	0.5499	0.2813	0.0818	9	0.3299
14	0.3210	0.4167	0.3747	0.3682	0.2792	0.5761	0.3792	0.3236	0.2385	0.3917	0.5517	0.4012	0.3299	9
15	0.4605	0.5352	0.6504	0.3930	0.5627	0.2067	0.3180	0.4197	0.5128	0.9102	0.2002	0.3611	0.4561	0.7111
16	0.2831	0.4260	0.5070	0.2481	0.3726	0.2341	0.1358	0.1978	0.2753	0.7261	0.1198	0.1130	0.1964	0.5010
17	0.2946	0.4693	0.5143	0.3046	0.3582	0.3723	0.1986	0.1836	0.2266	0.6680	0.2837	0.1179	0.1212	0.4411
18	0.3067	0.4805	0.4915	0.3259	0.3705	0.4530	0.2670	0.2164	0.1600	0.6320	0.2701	0.2114	0.1000	0.3000

- همانطور که بیان شد برای کارکرد صحیح الگوریتم درایه‌های صفر قطر فرعی با ۹ جایگزین شده‌اند.

خروجی تابع کلاسترینگ با الگوریتم WPGMA و حد آستانه ۰/۲۸:

```
cluster_w =

Columns 1 through 20

    1     1     1     1     1     2     3     3     3     4     2     3     3     1     2     3     3     3     5     6

Columns 21 through 40

    6     3     6     7     6     6     6     8     7     8     7     8     1     1     2     3     3     1     2     3

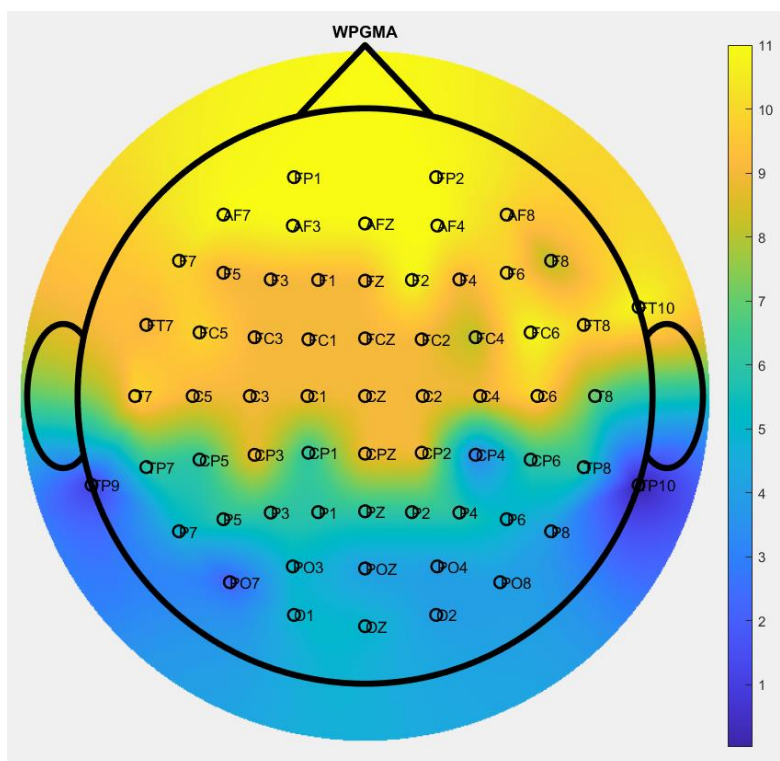
Columns 41 through 60

    3     3     4     2     3     3     3     2     6     3     3     9     6     6     6     6     7    10     8     8

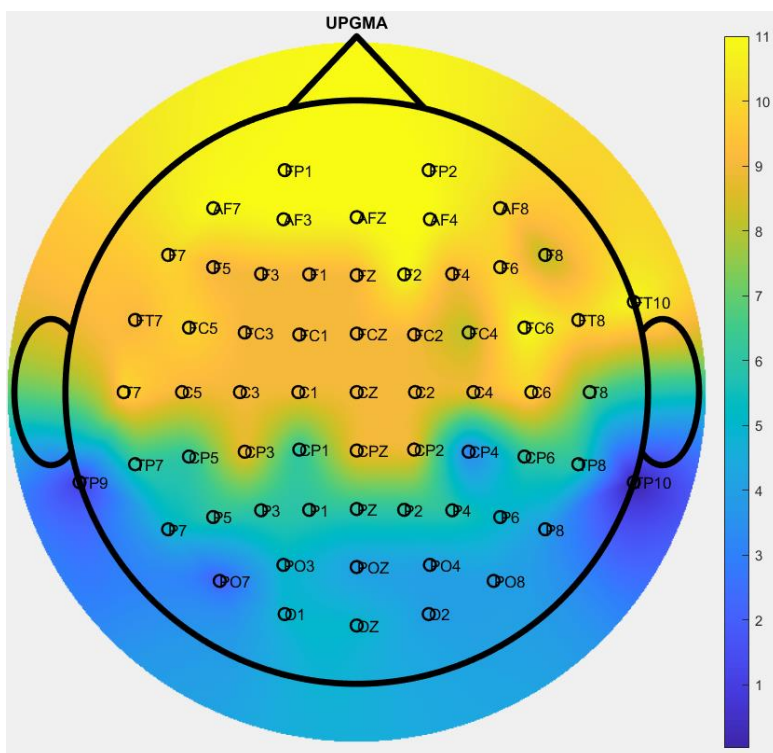
Columns 61 through 63

    7     11     12
```

نقشه توپوگرافی کلاسترها به ازای ورودی‌ها مذکور:



نقشه توپوگرافی کلاسترها با بهره‌گیری از الگوریتم *UPGMA*:



- برای آنکه ترکیب رنگی هر دو خروجی به نسبت یکسان و به طور واضح‌تری باشد، در هر دو مورد، بردارها را از تعداد کل کلاسترها کم کردیم تا کلاسترهای اولیه که شلوغ‌تر هستند رنگ‌های گرم‌تری داشته باشند.
در نهایت همانطور که مشهود است خروجی هر دو الگوریتم بسیار نزدیک به یکدیگر هستند و خروجی توپوگرافیک آن‌ها اختلاف مشهودی ندارد، با این حال خروجی تابع به ازای الگوریتم *UPGMA* با حد آستانه 0.28 در نهایت به 10 کلاستر تقسیم می‌شود در حالی که در الگوریتم دیگر 12 کلاستر به دست می‌دهد.
- کانال‌های الکترودهای نزدیک به یکدیگر در به طور متقارن در کلاسترهای یکسان قرار گرفته‌اند و همین انتظار نیز با توجه به آن که *ERP*ها متناسب با عملکرد جمعی نوروهای اطراف هر الکتروود بروز پیدا می‌کنند وجود داشت.
- برای کلاستریک دیتای هشت کاناله نیز همین فرآیند را به ازای $chNum = 8$ طی می‌کنیم و در نهایت به خروجی زیر دست پیدا می‌کنیم:

`cluster_for8 =`

1 1 2 2 2 3 3 4

طراحی فیلتر

اگر اندازه فیلتر که مقداری ثابت است را یک در نظر بگیریم خواهیم داشت:

$$H(\omega) = e^{j\phi(\omega)} = e^{\frac{-j\pi}{3}\text{sgn}(\omega)\omega}$$

تبدیل فوریه سیگنال ورودی نیز به صورت زیر است:

$$x(t) = \cos(\omega_0 t) + \cos(2\omega_0 t) \rightarrow x(t) = \frac{1}{2} * (e^{j\omega_0 t} + e^{-j\omega_0 t} + e^{j2\omega_0 t} + e^{-j2\omega_0 t})$$

$$\rightarrow X(\omega) = \frac{1}{2} * (\delta(\omega - \omega_0) + \delta(\omega + \omega_0) + \delta(\omega - 2\omega_0) + \delta(\omega + 2\omega_0))$$

با ضرب تابع تبدیل ورودی در تابع تبدیل سیستم تابع تبدیل خروجی به دست خواهد آمد (در این جا فرض می کنیم ω_0 مثبت است) :

$$Y(\omega) = H(\omega)X(\omega) = \frac{1}{2} * e^{\frac{-j\pi}{3}\omega.\text{sgn}(\omega)} * (\delta(\omega - \omega_0) + \delta(\omega + \omega_0) + \delta(\omega - 2\omega_0) + \delta(\omega + 2\omega_0))$$

$$\rightarrow Y(\omega) = \frac{1}{2} * (e^{\frac{-j\pi}{3}\omega_0}(\delta(\omega - \omega_0) + \delta(\omega + \omega_0)) + e^{\frac{-j\pi}{3}2\omega_0}(\delta(\omega - 2\omega_0) + \delta(\omega + 2\omega_0)))$$

$$y(t) = \cos\left(\omega_0 t \mp \frac{\pi}{3} \omega_0\right) + \cos\left(2\omega_0 t \mp \frac{4\pi}{3} \omega_0\right)$$

اما در صورتی که از فاز خطی استفاده کنیم:

$$y(t) = \cos\left(\omega_0 t - \frac{\pi}{3} \omega_0\right) + \cos\left(2\omega_0 t - \frac{4\pi}{3} \omega_0\right)$$

$$\begin{aligned} \frac{d}{d\omega} H(j\omega) &= \frac{d}{d\omega} A(j\omega) e^{j\phi(j\omega)} + j \frac{d}{d\omega} \phi(j\omega) A(j\omega) e^{j\phi(j\omega)} \\ &= e^{j\phi(j\omega)} \left(\frac{d}{d\omega} A(j\omega) + j A(j\omega) \frac{d}{d\omega} \phi(j\omega) \right) \end{aligned}$$

با توجه به فرض ثابت بودن اندازه فیلتر می توان گفت که مشتق A برابر با ۰ است.

$$\frac{d}{d\omega} H(j\omega) = e^{j\phi(j\omega)} \left(\frac{d}{d\omega} A(j\omega) + j A(j\omega) \frac{d}{d\omega} \phi(j\omega) \right)$$

$$\frac{\frac{d}{d\omega} H(j\omega)}{H(j\omega)} = \frac{\frac{d}{d\omega} A(j\omega)}{A(j\omega)} + j \frac{d}{d\omega} \phi(j\omega)$$

$$\frac{j \frac{d}{d\omega} H(j\omega)}{H(j\omega)} = -\frac{d}{d\omega} \phi(j\omega) + j \frac{\frac{d}{d\omega} A(j\omega)}{A(j\omega)}$$

$$\operatorname{Re} \left\{ \frac{j \frac{d}{d\omega} H(j\omega)}{H(j\omega)} \right\} = -\frac{d}{d\omega} \phi(j\omega) = g_d(\omega) \quad \therefore$$

هر چه N بزرگتر باشد نرخ نمونه برداری از تبدیل فوریه بیشتر خواهد بود و احتمال بیشتری وجود دارد که طول سیگنال کمتر از تناوب N باشد. در این صورت می توان سیگنال اصلی را FFT بازایی کرد و رخداد نامطلوب جمع مقادیر در فواصل N تایی رخ نخواهد داد.

- h یک سیگنال گسسته است که H تبدیل فوریه DTFT آن می باشد. برای پیاده کردن تابع مورد نظر در مطلب می توان از خواص تبدیل فوریه استفاده کرد. می دانیم که :

$$j \frac{dH(\omega)}{d\omega} \leftrightarrow nh[n]$$

در این جا به جای استفاده از تبدیل فوریه گسسته زمان DTFT از تابع fft که تبدیل فوریه گسسته DFT را خروجی می دهد استفاده می کنیم. در نتیجه کافی است که n هر مقدار را متناظرا در مقدار سیگنال در همان n ضرب کنیم، fft سیگنال حاصل را محاسبه کنیم و آن را تقسیم بر fft سیگنال اصلی کنیم و در نهایت جز حقیقی آن را به عنوان پاسخ نهایی اتخاذ کنیم.

- بله چنین فیلتری وجود دارد. در حقیقت اگر فیلتر تنها یک گین یا یک سیم مستقیم باشد به طوری که پاسخ ضریه فیلت برابر باشد با:

$$h[n] = K\delta[n]$$

- این فیلتر به وضوح حقیقی و علی است. تبدیل فوریه آن نیز عدد ثابت K می باشد در نتیجه فاز آن ثابت و برابر صفر است.

شناسایی کلمات

از آنجایی که کل جدول حروف ۶ ستون و ۶ سطر دارد برای مشخص کردن اعداد در RC paradigm نیاز است که ۱۲ عدد مشخص شوند و انتظار داریم که از اعداد ۱ تا ۱۲ استفاده شود. اما برای SC paradigm نیاز به ۳۶ عدد است تا بتوان همه کاراکترها را مشخص کرد. در نتیجه می‌توانیم با نگاه کردن مقادیری که در سطر ۱۰ train هر سابجکت متوجه شویم که متعلق به کدام پارادایم است. نتیجه به صورت زیر است:

a1 =

Columns 1 through 24

01234567891011121314151617181920212223

Columns 25 through 37

24252627282930313233343536

a2 =

Columns 1 through 24

01234567891011121314151617181920212223

Columns 25 through 37

24252627282930313233343536

a3 =

0123456789101112

a5 =

0123456789101112

a6 =

0123456789101112

a7 =

0123456789101112

a8 =

0123456789101112

a9 =

0123456789101112

در نتیجه سابجکت اول و دوم در SC paradigm و بقیه سابجکت‌ها در RC paradigm آزمایش شده‌اند.

برای دریافتن نحوه اندیس‌گذاری خانه‌های جدول در دو پارادایم از دو سطر ۱۰ و ۱۱ train data استفاده می‌کنیم. به این منظور بردار x را می‌سازیم که حاصل ضرب نقطه‌ای این دو سطر برای یک سابجکت دلخواه است. نتیجه این ضرب نقطه‌ای برداری است که متشکل از اندیس‌های درست و مرتب کاراکترهای کلمه مورد نظر است. از آنجا که کلمه مشخص است می‌توان به نحوه اندیس‌گذاری دست پیدا کرد.

در پارادایم SC از داده سابعکت دوم استفاده می کنیم. حاصل ضرب نقطه ای پس از حذف مقادیر تکراری که ناشی از ۱۵ بار نمایش دادن و همچنین بازه نمونه برداری هستند، ترتیب خانه های نمایش داده شده به صورت زیر هستند:

```
x2 =
      0      12      21      11      1      19
>>
```

به نظر می رسد که شماره گذاری به صورت زیر می باشد.

<u>1</u>	<u>2</u>	<u>3</u>	<u>4</u>	<u>5</u>	<u>6</u>
<u>7</u>	<u>8</u>	<u>9</u>	<u>10</u>	<u>11</u>	<u>12</u>
<u>13</u>	<u>14</u>	<u>15</u>	<u>16</u>	<u>17</u>	<u>18</u>
<u>19</u>	<u>20</u>	<u>21</u>	<u>22</u>	<u>23</u>	<u>24</u>
<u>25</u>	<u>26</u>	<u>27</u>	<u>28</u>	<u>29</u>	<u>30</u>
<u>31</u>	<u>32</u>	<u>33</u>	<u>34</u>	<u>35</u>	<u>36</u>

زیرا اولین حرف ۱۲ می باشد که مطابق با حرف L است. دومین حرف شماره ۲۱ U می باشد که صحیح است. سومین حرف ۱۱ که مطابق با حرف K است و صحیح است. چهارمین حرف ۱ است که مطابق با A است و صحیح است. آخرین حرف نیز ۱۹ که مطابق با S است و صحیح است. اگر شماره گذاری را به صورت دیگر انجام می دادیم یعنی در هر ستون با حرکت به بالا پایین شماره ها زیاد می شد، هیچ کدام از حروف درست مشخص نمی شدند.

حال برای RC کار مشابه انجام می دهیم. بعد از حذف صفر ها و حذف مقادیر تکراری که مکرر پشت هم ثبت شده اند به بردار زیر می رسیم:

Columns 1 through 24

8 6 6 8 8 6 6 8 8 6 6 8 8 6 6 8 8 6 6 8 8 6 8 6

Columns 25 through 48

6 8 6 8 8 6 10 3 3 10 3 10 3 10 10 3 3 10 10 3 10 3 3 10

Columns 49 through 72

10 3 3 10 10 3 10 3 3 10 3 10 5 8 8 5 5 8 8 5 5 8 8 5

Columns 73 through 96

8 5 8 5 5 8 8 5 5 8 8 5 8 5 5 8 5 8 1 7 7 1 1 7

Columns 97 through 120

7 1 1 7 7 1 1 7 7 1 1 7 7 1 1 7 7 1 7 1 7 1 1 7

Columns 121 through 144

10 1 10 1 10 1 1 10 10 1 10 1 10 1 10 1 1 10 1 10 1 10 1 10

با توجه به کلمه LUKAS شماره گذاری به ترتیبی که در ادامه مشاهده می کنید انجام شده است. ابتدا ۸ و ۶ که نشان دهنده حرف L هستند. پس از آن ۱۰ و ۳ که نشان دهنده U هستند. بعد از آن ۵ و ۸ که نشان دهنده k هستند. سپس ۷ و ۱ که نشان دهنده A هستند و در نهایت ۱۰ و ۱ که نشان دهنده s هستند.

	1	2	3	4	5	6
7						
8						
9						
10						
11						
12						

پیاده سازی الگوریتم ماشین لرنینگ

ساختار داده:

داده ها متشکل از یک ماتریس سه بعدی epoch شده برای هر شخص می باشد که بعد اول آن تعداد تریال ها، بعد دوم آن تعداد کانال ها و بعد سوم آن زمان را مشخص می کند. از آن جا که سیگنال downsample شده است ۸۵ نقطه زمانی در هر تریال وجود دارد که با ضرب در ۸ نتیجه ۶۸۰ می شود. در نتیجه ماتریس X که همان train features می باشد یک ماتریس ۲۷۰۰ در ۶۸۰ در دو آزمایش اول است. در آزمایش های بعدی تعداد تریال ها کمتر است و در نتیجه طول بعد اول آن ۹۰۰ می باشد.

بردار y یک بردار تحقق رخداد های مطلوب یا همان تارگت است. در ابتدای هر تریال اگر اندیس متناظر در سطر ۱۱ یک باشد به این معناست که این تریال مطلوب و تارگت است. در نتیجه بردار y نیز مشخص می شود. فرق این کار با تابع index extraction در آن است که در مورد دوم همه خانه های تارگت برگردانده می شوند نه فقط اولین خانه ی شروع تریال.

با اعمال الگوریتم svm نتایج حاصل می شود. مدل مورد طراحی ما برای همه training set ها به غیر از شماره پنج، ۱۰۰ درصد صحیح عمل می کند. همچنین در داده های شماره ۵ با خطای حدود ۲ درصد عملکرد قابل قبولی دارد. برای کاهش احتمال دسته بنی اشتباه و پیش بینی ۰ به جای ۱، ماتریس هزینه شده است که هزینه این خطا را به صورت قابل توجهی بالاتر از سایر خطا ها تنظیم می کند.

برای تشخیص کلمات روش کلی آن است که داده های ایپاک شده در سطر ۱۰ را در prediction به صورت المان به المان ضرب کنیم و در نهایت صفر های زائد را حذف کنیم و می توان اعداد را دیکود کنیم.

برای بررسی خطای مدل ساخته شده بر اساس هر الگوریتم، از تابع loss که در تولباکس ماشین لرنینگ قرار دارد استفاده کردیم. خطا ها در دو بردار errorSVM و errorLDA ذخیره شده اند.

```
errorLDA =
    0.0070    0.0087    0.0050         0    0.0017    0.0017    0.0017    0.0006    0.0006
...

errorSVM =
         0         0         0         0    0.0216    0.0001         0         0         0
```

مشاهده می کنید که الگوریتم SVM عملکرد بسیار بهتری نسبت به LDA در این مسئله دارد.

همچنین برای قسمت پیش بینی ابتدا همان طور که ذکر شد با ضرب داخلی ایپاک سطر ۱۱ و ۱۰ به اعداد مورد نظر می رسیم. ابتدا این کار را روی همان دیتای قسمت ترین انجام می دهیم:

ans =

12 21 11 1 19

ans =

12 21 11 1 19

به طور مثال روی سه داده اول قسمت ترین می توانیم ببینیم که اعداد درست استخراج شده اند. ۶ و ۸ مربوط به L، ۱۰ و ۳ مربوط به U، ۸ و ۵ مربوط به K، ۱ و ۷ مربوط به A، ۱۰ و ۱ مربوط به S هستند.

همچنین تابع دیگری به نام pred2word نوشته شده است که با گرفتن یک رشته عدد در پارادایم sc آن را به کلمه تبدیل می کند که می توانید عملکرد آن را در کد مشاهده کنید.

```

8 6 8 6 8 6 8 6 8 6 8 6 8 6 8 6 8 6 8
Columns 20 through 38
6 8 6 8 10 3 10 3 10 3 10 3 10 3 10 3 10 3 10
Columns 39 through 57
3 10 3 10 3 10 3 8 5 8 5 8 5 8 5 8 5 8 5
Columns 58 through 76
8 5 8 5 8 5 8 5 8 5 8 5 7 1 7 1 7 1 7
Columns 77 through 95
1 7 1 7 1 7 1 7 1 7 1 7 1 7 1 7 1 10 1
Columns 96 through 114
10 1 10 1 10 1 10 1 10 1 10 1 10 1 10 1 10 1 10

```