

بسمه تعالی



دانشگاه مهندسی برق

پروژه پایانی

درس سیگنال ها و سیستم ها

دکتر کربلایی

تهیه کنندگان:

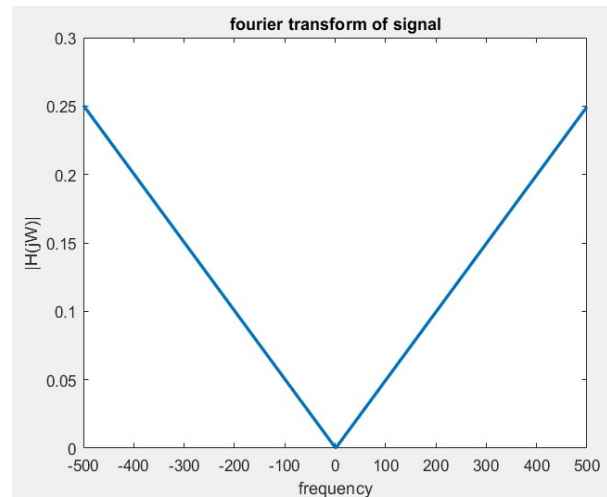
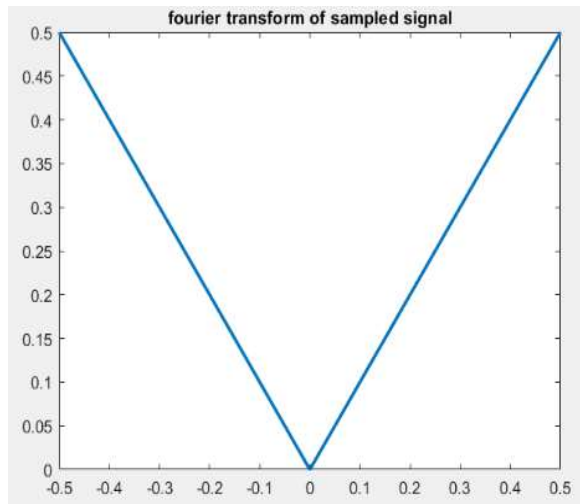
ثنا حریقی 98104047

شقایق عطایی 98102002

بهار 1399-1400

نمونه برداری – sampling

- تابع HalfBandFFt نوشته شده است که با گرفتن سیگنال حقیقی x و فرکانس نمونه برداری f_s ، خروجی y که همان تبدیل فوریه سیگنال در بازه $(-\pi, \pi)$ است را برگردانده و با مقیاس کردن محور فرکانس متناسب با f_s ، تبدیل فوریه سیگنال ورودی را رسم می کند. شکل زیر نمونه ای از خروجی این تابع را به ازای $f_s = 1000$ نشان می دهد:



شکل سمت چپ تبدیل فوریه سیگنال نمونه برداری شده و شکل سمت راست تبدیل فوریه سیگنال اصلی را نشان می دهد. همانطور که می بینیم پهنای باند سیگنال اصلی نسبت به سیگنال نمونه برداری شده f_s برابر شده است.

توجه داریم که برای برقرای انرژی سیگنالها رابطه پارسوال به صورت زیر در می آید. پارسوال برای سیگنالهای گسسته:

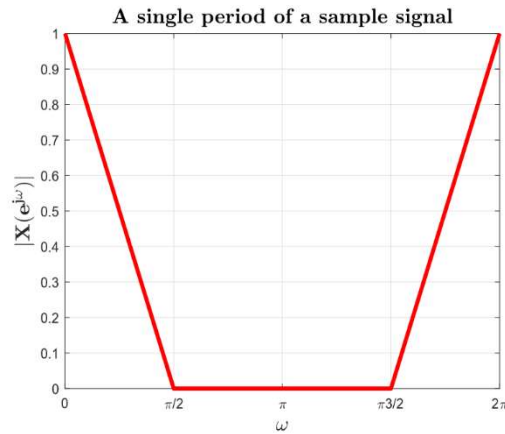
$$\sum_{n=-\infty}^{\infty} |x[n]|^2 = \frac{1}{2\pi} \int_0^{2\pi} |X(e^{j\Omega})|^2 d\Omega \quad \text{رابطه 1}$$

همچنین می دانیم رابطه زیر بین تبدیل فوریه سیگنال اصلی و سیگنال گسسته نمونه برداری شده وجود دارد:

$$X(e^{j\Omega}) = \frac{1}{T} \sum_{r=-\infty}^{\infty} X_c\left(j\left(\frac{\Omega}{T} - \frac{2r\pi}{T}\right)\right) \quad \text{رابطه 2}$$

از آنجایی که در رابطه پارسوال انرژی $X(e^{j\Omega})$ در یک تناوب محاسبه می شود، در رابطه 2 نیز سمت راست معادله را تنها در یک تناوب در نظر گرفته و رابطه 1 به شکل زیر در می آید:

$$\sum_{n=-\infty}^{\infty} |x[n]|^2 = \frac{1}{2\pi} \frac{1}{T^2} \int_0^{2\pi} \left| X_c\left(j\left(\frac{\Omega}{T}\right)\right) \right|^2 d\Omega \xrightarrow{\Omega' = \frac{\Omega}{T}} = \frac{1}{2\pi T} \int_0^{2\pi/T} |X_c(j(\Omega'))|^2 d\Omega'$$

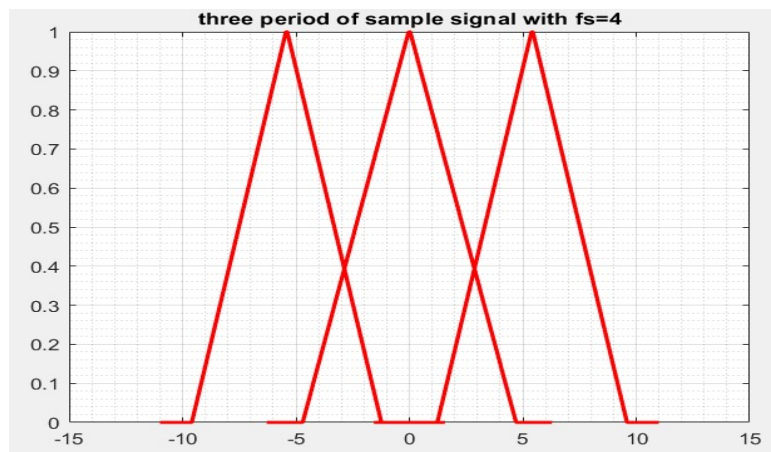


شکل 1-1

- اگر فرض کنیم در شکل 1 فرکانس نمونه برداری f_{s0} بوده باشد، طبق رابطه 1 می دانیم که پهنای باند سیگنال اولیه $\frac{\pi}{2}$ است. از طرفی طبق قضیه نمونه برداری نایکوئیست می دانیم که فرکانس نمونه برداری باید از دو برابر بیشینه فرکانس سیگنال اصلی بزرگتر (در مواردی مساوی) باشد بنابراین:

$$f_{max} = \frac{\omega_{max}}{2\pi} = \frac{\frac{\pi}{2} * f_{s0}}{2\pi} = \frac{f_{s0}}{4} \quad , \quad f_s \geq 2f_{max} \rightarrow f_s \geq \frac{f_{s0}}{2}$$

- اگر f_{s0} برابر 12 باشد، طبق رابطه 1 پهنای باند سیگنال اولیه برابر $\frac{\pi}{2} * f_{s0} = 6\pi$ بدست می آید. بنابراین اگر f_s را به 4 تغییر دهیم، پهنای باند متناظر در تبدیل فوریه سیگنال نمونه برابر $\frac{6\pi}{4} = 1.5\pi$ خواهد شد که با توجه به متناوب بودن تبدیل فوریه گسسته با تناوب 2π منجر به تداخل فرکانسی یا aliasing می شود. در شکل زیر سه تناوب از تبدیل فوریه سیگنال گسسته رسم شده است که همانطور که گفتیم تداخل رخ داده است:



بنابراین با اعمال فیلتر ایده آل با پهنای باند π نمی توان سیگنال را به درستی بازسازی کرد و باید پیشتر از فیلتر آنتی aliasing استفاده کنیم.

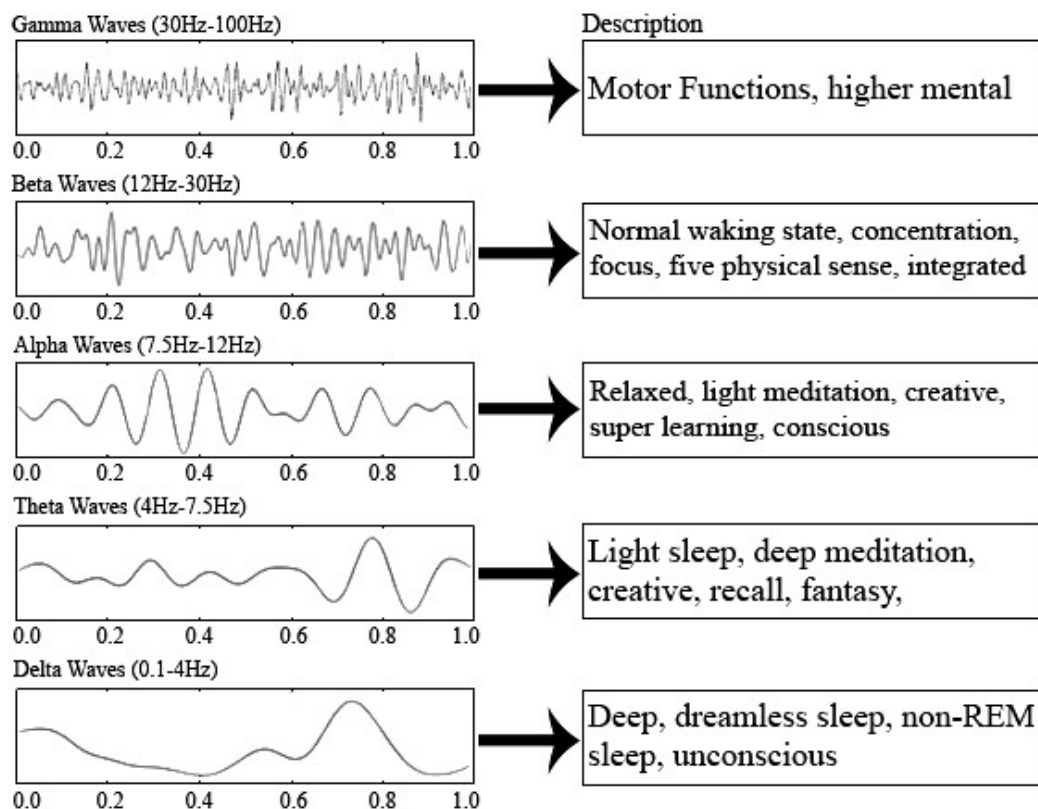
آشنایی با سیگنال های EEG

- ERP یا Event-Related Potentials میدانهای الکتریکی ثبت شده ای هستند که مغز در یک زمان مشخص در پاسخ به یک رویداد تولید می کند. ERP تنها روش غیر تهاجمی ای است که الگوی پویای وقایع مغز را تا حدود میلی ثانیه حل و بررسی می کند. اشکال موج ERP اختلاف پتانسیل بین دو نقطه ضبط را نشان می دهد یکی از مثال های ERP، سیگنال P300 است. سیگنال p300 یک پتانسیل مغزی مربوط به وقایع است که با استفاده از EEG اندازه گیری می شود. این سیگنال به اوج فعالیت تقریباً 300ms پس از اعمال شدن محرک هدف اشاره دارد؛ که با محرک های استاندارد جایگزین شده تا پارادایم "oddball" را ایجاد کند که معمولاً شنوایی است. در آن پارادایم فاعل باید فقط به محرک هدف غیر تناوبی پاسخ دهد (و نه محرک هدف تناوبی). دامنه پاسخ p300 متناسب با میزان پردازش اطلاعات مورد نیاز است. هم چنین مفهوم تاخیر معیاری از سرعت طبقه بندی محرک هاست که ارتباطی با زمان پاسخ رفتاری ندارد.

- در جدول زیر باند های فرکانسی گوناگون شرح داده شده است:

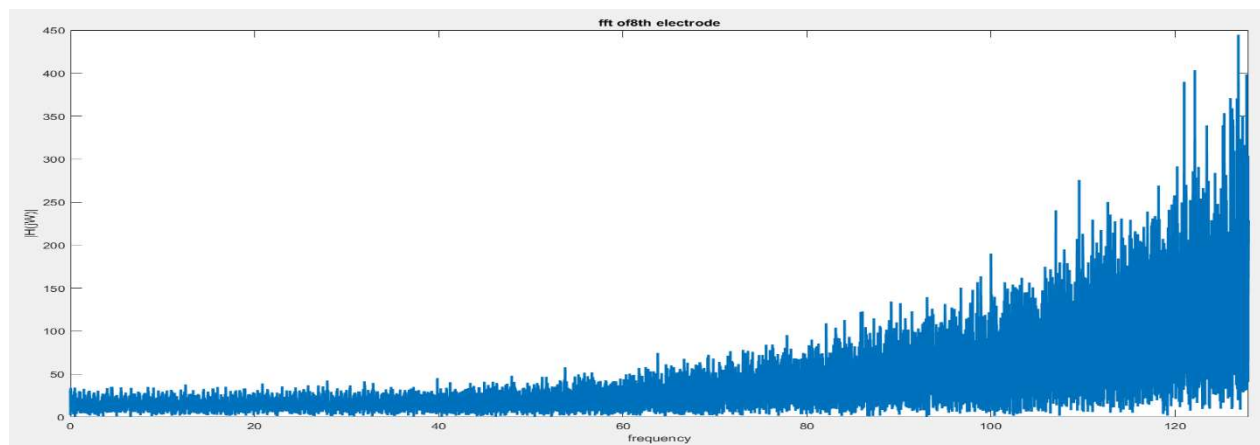
Name	δ	θ	α	β	γ	High γ
frequency	0.1 – 4 Hz	4 – 7 Hz	8 – 13 Hz	15–30 Hz	30–80 Hz	80–150 Hz

جدول 1-2) باند های فرکانسی سیگنال های مغزی

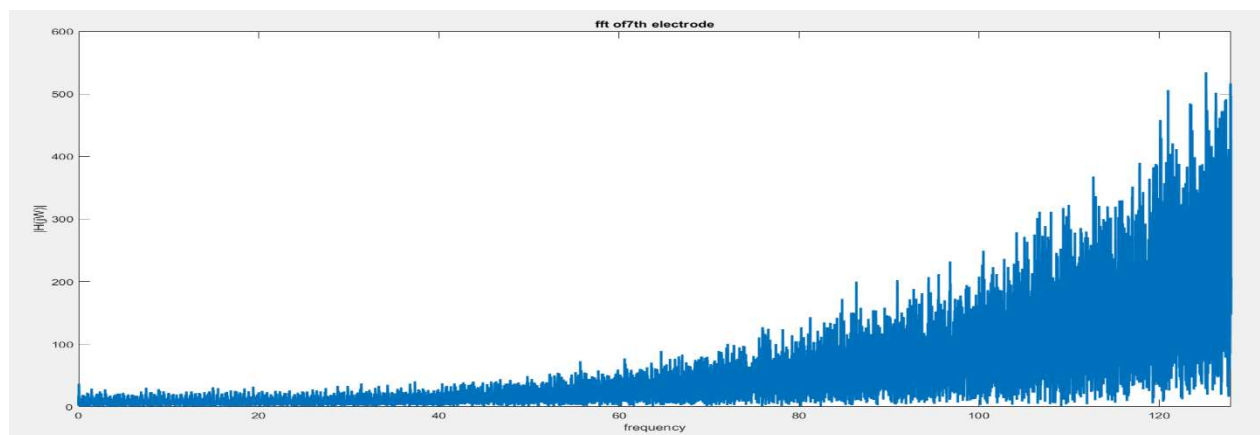


شکل 1-2) فعالیت های متناظر با هر فرکانس از سیگنال

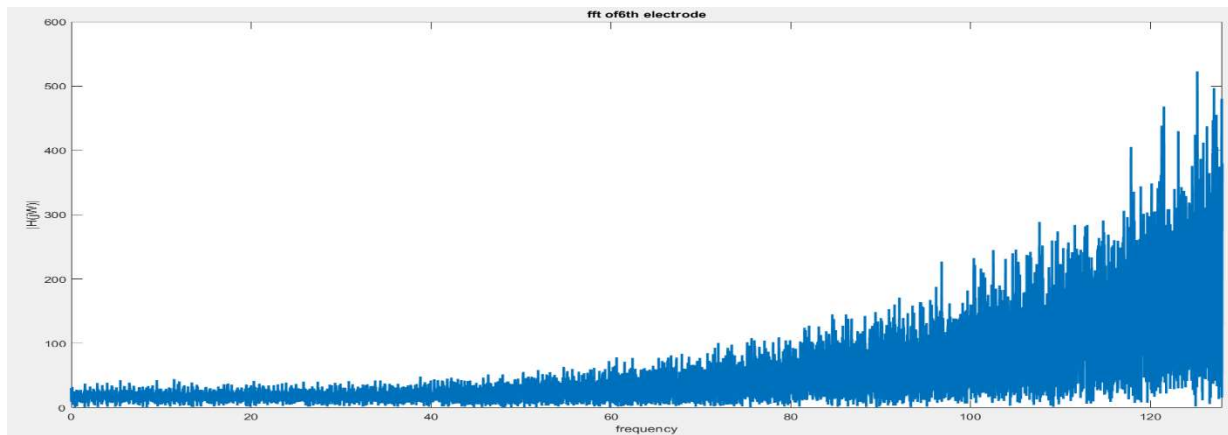
- با توجه به قضیه نایکوئست فرکانس نمونه برداری حداقل دو برابر بیشترین مولفه فرکانسی سیگنال است بنابراین حداقل فرکانس نمونه برداری مناسب بین 160 – 300 Hz است. (به شرطی که فرض کنیم High gamma موجود است)
- با محاسبه فاصله زمانی دو نمونه برداری در سطر اول دوره تناوب و سپس فرکانس نمونه برداری را محاسبه می کنیم در مورد subject1 فرکانس نمونه برداری برابر با 256 Hz است.
- با توجه به جدول باند های فرکانسی گوناگون در مغز (با فرض اینکه داده های تا گاما را خواهیم داشت) بیشترین فرکانس در حدود 50 Hz و کمترین آنها 0.1Hz است پس فرکانس قطع فیلتر باید برابر با این دو فرکانس (به ترتیب از بالا و پایین) باشد.
- با استفاده از تابع HalfBandFft اندازه طیف فرکانسی هشت الکتروود را رسم می کنیم. (فرکانس نمونه برداری را برابر 256 قرار می دهیم). تعدادی از نمودارها در شکلهای زیر نشان داده شده است.



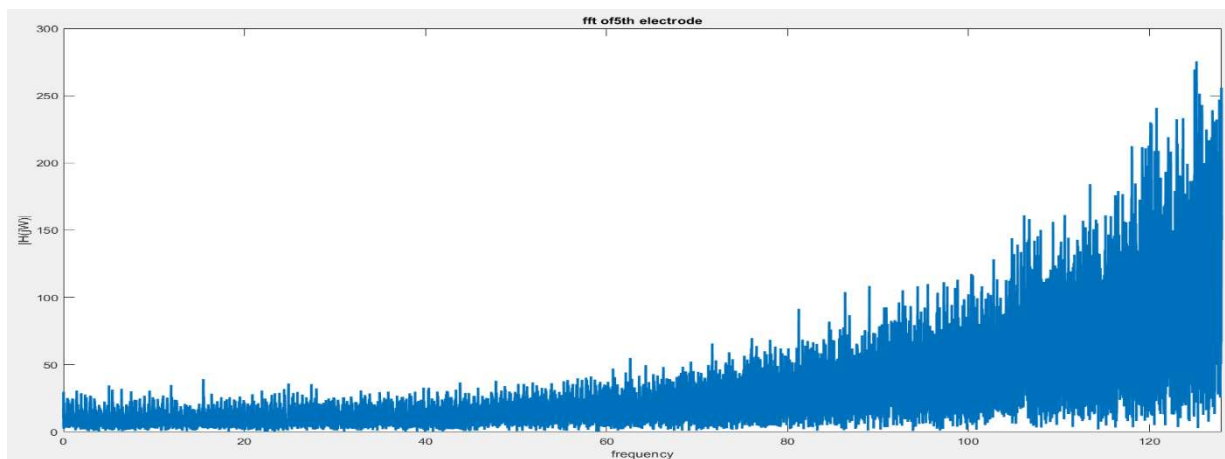
شکل 2-2) طیف فرکانسی الکتروود هشتم



شکل 2-3) طیف فرکانسی الکتروود هفتم



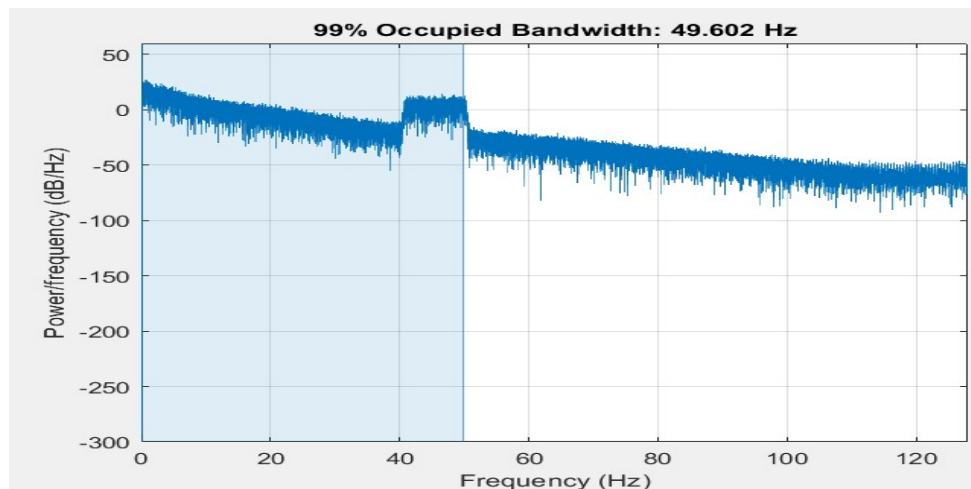
شکل 2-4) طیف فرکانسی الکترود ششم



شکل 2-5) طیف فرکانسی الکترود پنجم

سایر الکترودها نیز طیف فرکانسی مشابهی دارند. با توجه به طیف فرکانسی فرکانس قطع مناسب 40 تا 60 هرتز می باشد.

- با استفاده از تابع `obw` متلب و با وارد کردن داده های کانال و فرکانس نمونه برداری (که در قسمت های قبلی محاسبه شده) محدوده فرکانسی با بیشترین انرژی را می یابیم. همچنین برای صرف نظر کردن از مولفه DC کافی است `fft` سیگنال را محاسبه کنیم؛ مولفه اول این بردار برابر با مولفه DC سیگنال است؛ آنرا برابر با صفر گذاشته و `ifft` می گیریم. از آنجایی که مقدار فرکانس برای هر 8 کانال بسیار به هم نزدیک است در اینجا نمودار کانال اول در دیتاست اول آورده شده است :



البته این نکته حائز اهمیت است که این فرکانس در دیتاست های دیگر کمتر از 50Hz است اما به صورت کلی برای فیلترینگ از همین فرکانس به عنوان فرکانس قطع بالا استفاده شده است (زیرا به این صورت نیز هنوز سیگنال تمیزی داریم) بنابراین با توجه به نمودار های مرسوم، فرکانس قطع در حدود 50 Hz است.

- برای کم کردن مولفه DC از سیگنال کافی است میانگین سیگنال را از آن کم کنیم سپس سیگنال را از یک فیلتر بالاگذر با فرکانس قطع کم مثلا در حدود 1 Hz عبور دهیم.
- کم کردن میانگین سیگنال از آن برای از بین بردن بخش DC (به شرط آنکه قبل از فیلترینگ صورت گیرد) بهینه نیست زیرا در این صورت بخش DC نویز همراه با سیگنال نیز از بین رفته و در ادامه نتیجه درستی از فیلترینگ دریافت نمی کنیم. بنابراین در کنار کم کردن میانگین، سیگنال را از یک فیلتر بالاگذر با فرکانس قطع بسیار اندک نیز می گذرانیم تا بخش DC از بین برود. (از آنجایی که فرکانس بخش DC، 0 است باید فرکانس قطع فیلتر بالاگذر را اندک انتخاب کنیم.) بنابراین در نهایت می توانیم از یک فیلتر میان گذر با فرکانس قطع بالای 50Hz و فرکانس قطع پایین 1Hz استفاده کنیم. برای فیلتر میان گذر از دستور bandpass استفاده می کنیم.
- با توجه به طول فیلتر که 50 در نظر گرفته شده اگر فرکانس نمونه برداری جدید را دو برابر این مقدار یعنی 100 Hz انتخاب کنیم؛ همچنین با توجه به فرکانس نمونه برداری قدیمی می توانیم از تابع downsample استفاده کنیم. برای تعیین مقدار n در این تابع، کافی است سقف حاصل تقسیم فرکانس نمونه برداری بر 100 را بیابیم. به این ترتیب برای همه ردیف های دیتا، downsampling را انجام داده و ماتریس جدیدی از دیتاها می سازیم و به تابع epoching می دهیم.

- با توجه به اصل ناکوئیست و پدیده aliasing، چرا نمی توانیم فرکانس نمونه برداری را قبل از فیلتر کردن کاهش دهیم؟
زیرا به علت حضور نویز ممکن است فرکانس قطع کمتر از فرکانس لازم را انتخاب کنیم و به این ترتیب aliasing رخ دهد و تشخیص ما در نهایت با خطای فاحشی روبه رو شود.

- این کار که ابتدا epoch کنیم و سپس فیلتر، چه فرقی خواهد داشت با این که ابتدا فیلتر کنیم و سپس پنجره را انتخاب کرده و جدا کنیم؟
با توجه به مثال داده شده از آنجایی که ما طول سیگنال نامحدود زمانی را محدود می کنیم؛ در حال ایجاد اثرات لبه ای هستیم که فیلتر وظیفه رفع و نرم کردن آنرا دارد؛ بنابراین بهتر است ابتدا فیلتر کنیم سپس با epoching سیگنال را در پنجره های معین بپیماییم.
- چرا لازم است در ابتدای داده گیری حتماً مدت زمانی پس از شروع داده گیری توسط کلاه EEG، هنوز آزمایش را شروع نکنیم؟
با قرار دادن کلاه EEG و متصل کردن کانال ها یک محرک خارجی برای نوروها قرار می دهیم بنابراین باید صبر کنیم تا پاسخ گذرای حاصل از قرار گرفتن این محرک خارجی سپری شود و سپس آزمایش اصلی را شروع کنیم و محرک های مورد نظر را اعمال کنیم. در غیر این صورت پاسخ های نادرستی به عنوان نویز با سیگنال حاصل از پاسخ به تحریک/عدم تحریک مورد نظر ما جمع میشود و داده ها خطا خواهند داشت.

خوشه بندی بر مبنای همبستگی

- نشان دهید $r_{XY} = \frac{\int_{-\infty}^{\infty} X(t)Y(t)dt}{\sqrt{\int_{-\infty}^{\infty} X^2(t)dt \int_{-\infty}^{\infty} Y^2(t)dt}}$ ، عددی بین -1 و 1 است و $|r_{XY}|$ برابر یک تنها در حالتی رخ می دهد که دو سیگنال تنها در یک ضریب تفاوت داشته باشند.

عدد حقیقی α را در نظر گرفته و می دانیم که عبارت زیر همواره نامنفی است:

$$\int_{-\infty}^{\infty} (X(t) - \alpha Y(t))^2 dt \geq 0 \rightarrow \int_{-\infty}^{\infty} X^2(t)dt + \alpha^2 \int_{-\infty}^{\infty} Y^2(t)dt - 2\alpha \int_{-\infty}^{\infty} X(t)Y(t)dt \geq 0 \quad \text{نامساوی 1}$$

ابتدا نامساوی 1 را در حالت اکیدا مثبت در نظر می گیریم. اگر بخواهیم معادله درجه دو فوق به ازای هر α ، مثبت باشد، باید دلتای معادله منفی شود: (ضریب α^2 ، $\int_{-\infty}^{\infty} Y^2(t)dt$ ، عددی مثبت است و تنها شرط همواره مثبت بودن رابطه، منفی بودن دلتای معادله است.)

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

$$\Delta = 4 \left(\int_{-\infty}^{\infty} X(t)Y(t)dt \right)^2 - 4 \int_{-\infty}^{\infty} Y^2(t)dt \int_{-\infty}^{\infty} X^2(t)dt < 0$$

$$\rightarrow \left(\int_{-\infty}^{\infty} X(t)Y(t)dt \right)^2 < \int_{-\infty}^{\infty} Y^2(t)dt \int_{-\infty}^{\infty} X^2(t)dt$$

$$\rightarrow \frac{\left(\int_{-\infty}^{\infty} X(t)Y(t)dt \right)^2}{\int_{-\infty}^{\infty} Y^2(t)dt \int_{-\infty}^{\infty} X^2(t)dt} < 1$$

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

حال اگر نامساوی را در حالت برابر با صفر بررسی کنیم، خواهیم داشت:

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

$$\Delta = 4 \left(\int_{-\infty}^{\infty} X(t)Y(t)dt \right)^2 - 4 \int_{-\infty}^{\infty} Y^2(t)dt \int_{-\infty}^{\infty} X^2(t)dt \geq 0$$

$$\rightarrow \left(\int_{-\infty}^{\infty} X(t)Y(t)dt \right)^2 \geq \int_{-\infty}^{\infty} Y^2(t)dt \int_{-\infty}^{\infty} X^2(t)dt \quad (1)$$

طبق نامساوی کوشی شوارتز در حالت پیوسته، باید داشته باشیم:

$$\left(\int_{-\infty}^{\infty} X(t)Y(t)dt \right)^2 \leq \int_{-\infty}^{\infty} Y^2(t)dt \int_{-\infty}^{\infty} X^2(t)dt \quad (2)$$

طبق (1) و (2) تنها زمانی معادله جواب صفر خواهد داشت که :

$$\begin{aligned} \left(\int_{-\infty}^{\infty} X(t)Y(t)dt \right)^2 &= \int_{-\infty}^{\infty} Y^2(t)dt \int_{-\infty}^{\infty} X^2(t)dt \\ \rightarrow \frac{\left(\int_{-\infty}^{\infty} X(t)Y(t)dt \right)^2}{\int_{-\infty}^{\infty} Y^2(t)dt \int_{-\infty}^{\infty} X^2(t)dt} &= 1 \end{aligned}$$

در نتیجه در این حالت r_{XY} طبق تعریف برابر ± 1 خواهد بود.

همچنین در این حالت دلتا صفر شده و تنها به ازای یک مقدار α نامساوی صفر می شود:

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

و با توجه به نامنفی بودن $(X(t) - \alpha Y(t))^2$ ، رابطه زیر حاصل می شود:

$$X(t) = \alpha Y(t)$$

- چرا این معیار، معیار مناسبی برای سنجش شباهت دوسیگنال است.

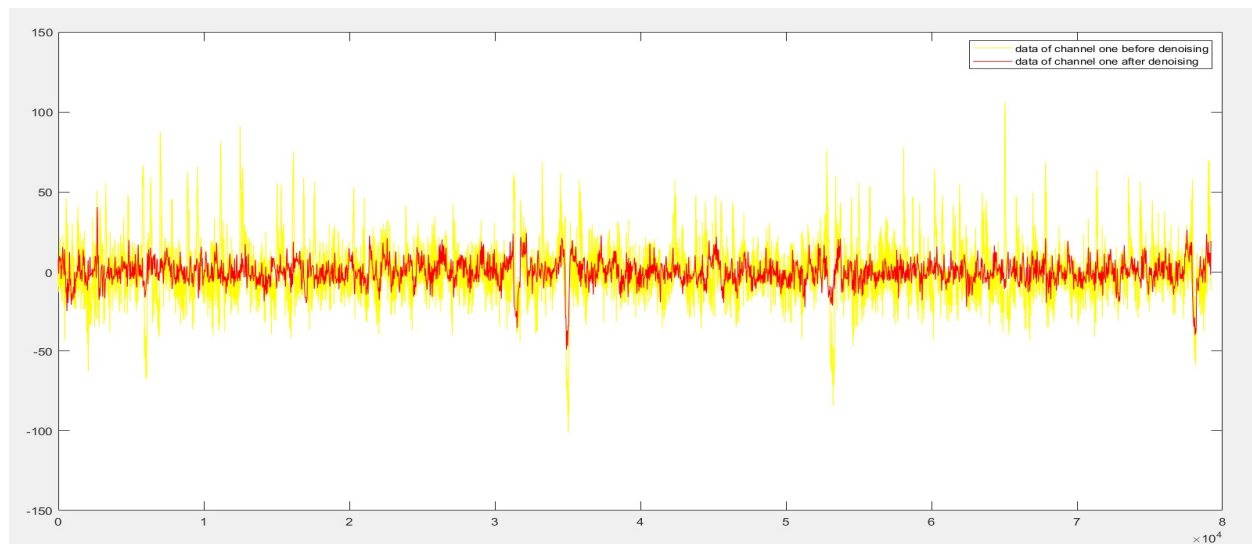
ضریب همبستگی خطی (r_{XY} - linear correlation coefficient) قدرت و جهت رابطه خطی بین دو متغیر X و Y را اندازه گیری می کند؛ علامت ضریب همبستگی خطی جهت رابطه خطی بین X و Y را نشان می دهد. همچنین وقتی r_{XY} نزدیک 1 یا -1 باشد، رابطه خطی قوی است و می توان از X برای تخمین Y استفاده کرد اما وقتی این پارامتر نزدیک 0 باشد، رابطه خطی بین دو بردار ضعیف و عملاً از هم مستقل اند.

همچنین می توان رابطه r_{XY} را به گونه ای ضرب داخلی دو بردار X و Y در نظر گرفت زیرا همانطور که دیدیم زمانیکه دو بردار با یکدیگر رابطه خطی داشته باشند، اندازه ضریب همبستگی آنها 1 شده و اگر از هم مستقل باشند، ضریب همبستگی صفر می شود.

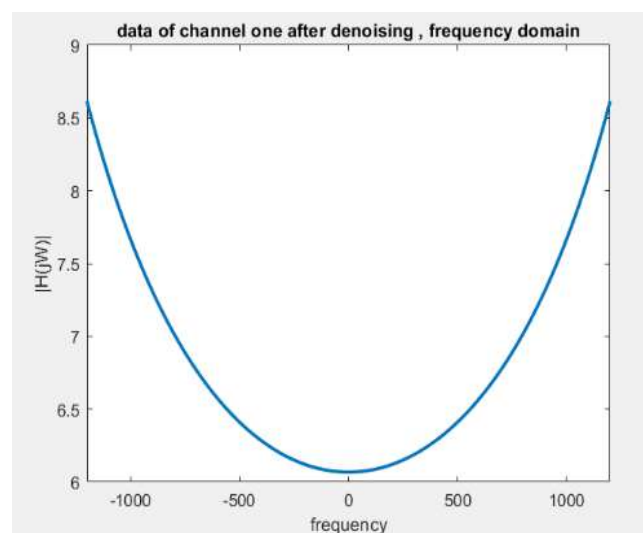
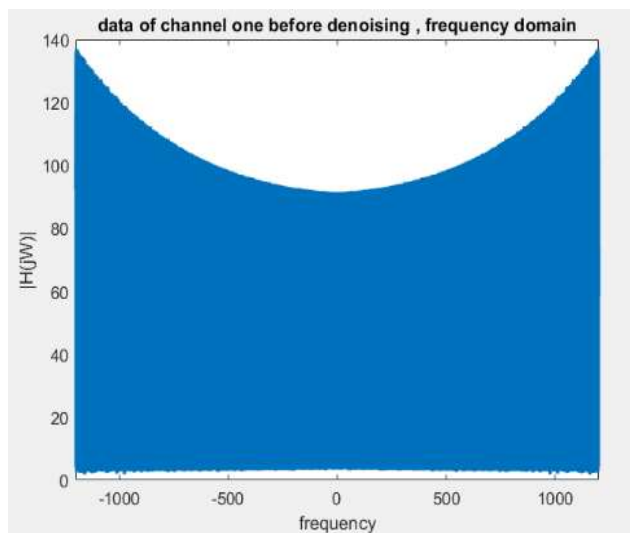
- با مشاهده طیف فرکانسی، در صورت لزوم فیلتری را بر داده ها اعمال کنید تا نویز در نتایج حاصل اثر هر چه کم تری داشته باشد.

در بخش آشنایی با سیگنالهای EEG دیدیم که فرکانس قطع بالا تقریباً 50 هرتز و فرکانس قطع پایین 0.1 می باشد بنابراین میتوانیم از فیلتری میان گذر با این فرکانس های قطع استفاده کنیم. برای طراحی فیلتر از `fdatool` متلب استفاده می کنیم که

فیلتر استفاده شده تحت عنوان '*filter.mat*' ضمیمه شده است و درنهایت با استفاده از تابع *filtfilt* آن را اعمال می کنیم. به عنوان مثال دیتای کانال یک پیش از و پس از اعمال فیلتر رفع نویز به صورت زیر می باشد:



دو شکل زیر به ترتیب از راست به چپ، دیتای رفع نویز شده و نویزی کانال اول را در حوزه فرکانس نشان می دهد: (F_s را برابر 2400 هرتز در نظر می گیریم).



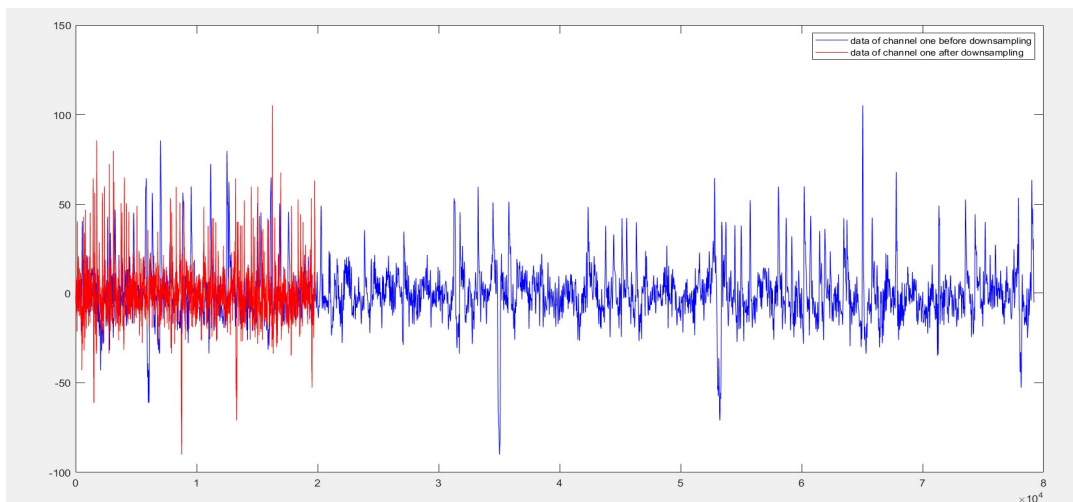
- فرکانس نمونه برداری را تا حد معقولی کاهش دهید.

برای کاهش فرکانس نمونه برداری، داده ها را با نرخ M مطابق رابطه زیر حذف می کنیم. در این رابطه x سیگنال اصلی و x_d سیگنال کاهش داده شده است:

$$x_d[n] = x[Mn]$$

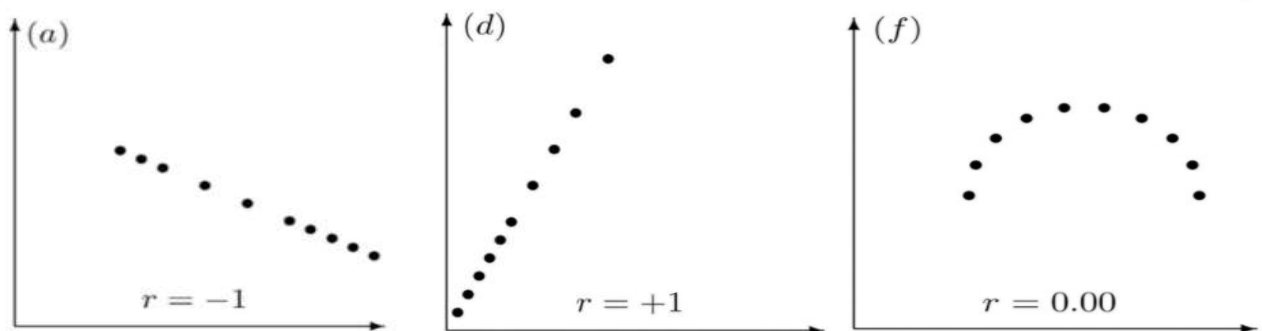
برای این منظور تابع **downsampler** نوشته شده است که x (سیگنال اولیه) و M (نرخ کاهش فرکانس نمونه برداری) را به عنوان ورودی می گیرد.

در صورتیکه M را برابر 4 در نظر بگیریم، دیتای کانال یک پیش و پس از **downsampling** به صورت زیر خواهد بود:



- برای فاصله ی دوکانال که زیاد بودن آن نشان دهنده عدم شباهت دوکانال است، چه رابطه ای را پیشنهاد می کنید؟

با توجه به قسمت قبل، هر چه اندازه r_{xy} به یک نزدیکتر باشد دو سیگنال رابطه خطی و در نتیجه شباهت بیشتری با هم دارند بنابراین برای تعریف فاصله دو کانال که نشان دهنده ی شدت تفاوت دو سیگنال است می توان از تعریف $1 - |r_{xy}|$ استفاده کرد. سه شکل زیر رابطه بین دو سیگنال را برای سه حالت خاص r نشان می دهد.



برای تعیین فاصله دو خوشه چهار الگوریتم رایج وجود دارد که به توضیح هر یک می پردازیم:

1. (unweighted pair group method with arithmetic mean) upgma

در این الگوریتم فاصله بین هر دو خوشه A و B که اندازه های آنها به ترتیب با $|A|$ و $|B|$ نشان داده می شود به صورت زیر تعیین می شود که در آن x اشیای موجود در خوشه A و y اشیای موجود در خوشه B است:

$$\frac{1}{|A| \cdot |B|} \sum_{x \in A} \sum_{y \in B} d(x, y)$$

به عبارت دیگر در هر مرحله خوشه بندی، فاصله بین دو خوشه ادغام شده A و B با خوشه جدید x طبق رابطه زیر تغییر پیدا می کند:

$$d_{(A \cup B), X} = \frac{|A| \cdot d_{A, X} + |B| \cdot d_{B, X}}{|A| + |B|}$$

حال به توضیح این الگوریتم با یک مثال می پردازیم:

- در قدم اول ماتریس فاصله تشکیل داده شده و دو خوشه با کمترین فاصله باهم ادغام می شوند که در اینجا a و b هستند:

	a	b	c	d	e
a	0	17	21	31	23
b	17	0	30	34	21
c	21	30	0	28	39
d	31	34	28	0	43
e	23	21	39	43	0

$$\delta(a, u) = \delta(b, u) = \frac{17}{2} = 8.5 \rightarrow \text{first branch length estimation}$$

فواصل چنانچه گفته شد به صورت زیر بروزرسانی می شوند:

$$D_2((a, b), c) = (D_1(a, c) \times 1 + D_1(b, c) \times 1) / (1 + 1) = (21 + 30) / 2 = 25.5$$

$$D_2((a, b), d) = (D_1(a, d) + D_1(b, d)) / 2 = (31 + 34) / 2 = 32.5$$

$$D_2((a, b), e) = (D_1(a, e) + D_1(b, e)) / 2 = (23 + 21) / 2 = 22$$

- قدم دوم:

	(a,b)	c	d	e
(a,b)	0	25.5	32.5	22
c	25.5	0	28	39
d	32.5	28	0	43
e	22	39	43	0

$$\delta(a, v) = \delta(b, v) = \delta(e, v) = \frac{22}{2} = 11 \rightarrow \text{second branch length estimation}$$

$$D_3(((a, b), e), c) = (D_2((a, b), c) \times 2 + D_2(e, c) \times 1) / (2 + 1) = (25.5 \times 2 + 39 \times 1) / 3 = 30$$

$$D_3(((a, b), e), d) = (D_2((a, b), d) \times 2 + D_2(e, d) \times 1) / (2 + 1) = (32.5 \times 2 + 43 \times 1) / 3 = 36$$

• قدم سوم:

	$((a,b),e)$	c	d
$((a,b),e)$	0	30	36
c	30	0	28
d	36	28	0

$$D_4((c,d),((a,b),e)) = (D_3(c,((a,b),e)) \times 1 + D_3(d,((a,b),e)) \times 1) / (1 + 1) = (30 \times 1 + 36 \times 1) / 2 = 33$$

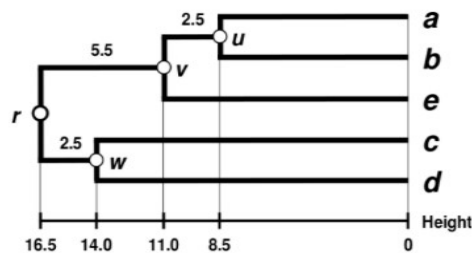
$$\delta(c,w) = \delta(d,w) = \frac{28}{2} = 14 \rightarrow \text{third branch length estimation}$$

• قدم آخر:

	$((a,b),e)$	(c,d)
$((a,b),e)$	0	33
(c,d)	33	0

$$\delta((c,d),r) = \delta(((a,b),e),r) = \frac{33}{2} = 16.5$$

بنابراین خوشه بندی نهایی به صورت مقابل در می آید:

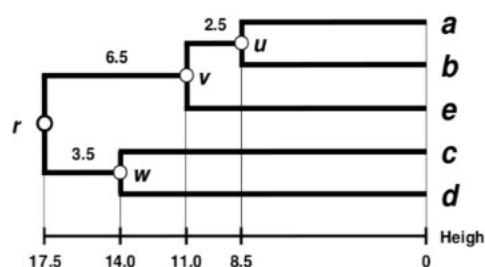


2. wpgma (weighted pair group method with arithmetic mean)

این الگوریتم مشابه upgma است با این تفاوت که فاصله بین دو خوشه ادغام شده i و j با خوشه k از رابطه زیر تعیین می شود و مستقل از اندازه i و j می باشد:

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

اگر ماتریس مثال الگوریتم upgma را طبق این تعریف خوشه بندی کنیم به نتیجه نهایی زیر می رسم:



3. complete linkage clustering

در این الگوریتم فاصله بین دو خوشه X و Y به صورت زیر تعیین می شود که X و Y به ترتیب اعضای موجود در خوشه X و Y هستند:

$$D(X, Y) = \max_{x \in X, y \in Y} d(x, y)$$

4. single linkage clustering

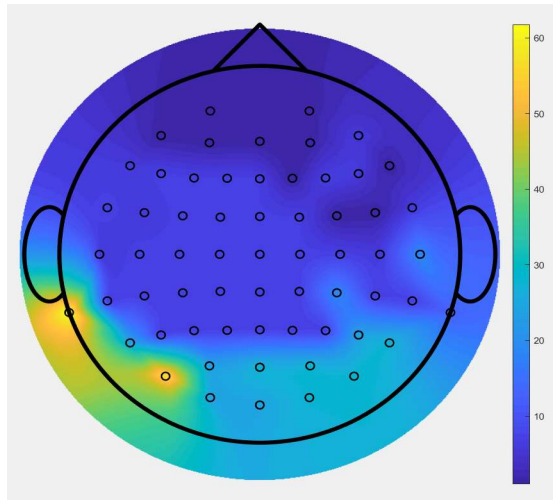
در این الگوریتم فاصله بین دو خوشه X و Y به صورت زیر تعیین می شود که X و Y به ترتیب اعضای موجود در خوشه X و Y هستند:

$$D(X, Y) = \min_{x \in X, y \in Y} d(x, y)$$

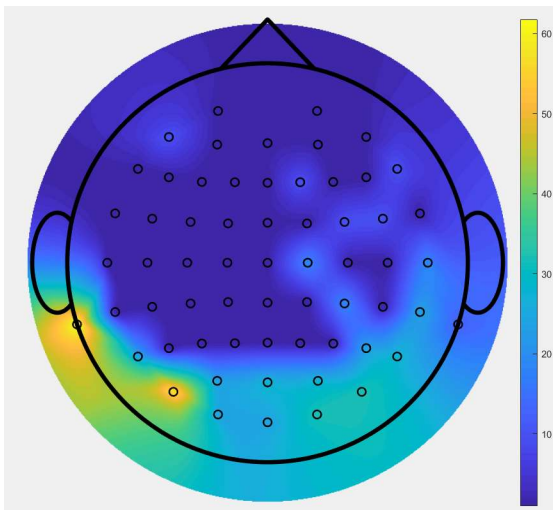
- برای پیاده سازی تابع `CorrelationCluster`، از دو الگوریتم `wpgma` و `upgma` استفاده می کنیم، بنابراین علاوه بر دو ورودی `InputCorrMat` و `DistanceMeasure` که به ترتیب ماتریس فاصله و معیار خوشه بندی را تعیین می کنند از ورودی دیگری برای تعیین نوع الگوریتم نیز استفاده می کنیم.
- می توان معیار را عددی کوچک در نظر گرفت؛ به نحوی که هرگاه فاصله بین دو خوشه از این مقدار کمتر شد الگوریتم متوقف شود. همچنین می توان آن را به عنوان تعداد خوشه در نظر گرفت. بدیهی است که در این صورت این عدد باید بین 1 و 63 باشد. یک معیار دیگر می تواند تعداد مراحل باشد و در هر مرحله بررسی کنیم که آیا خوشه های ایجاد شده همبستگی را به خوبی نشان می دهند یا خیر و تعداد مراحل را بر این اساس تعیین می کنیم.
- در این تابع ما `DistanceMeasure` را تعداد مراحل در نظر می گیریم. با انجام خوشه بندی به ازای دو الگوریتم گفته شده در می یابیم که الگوریتم `wpgma` در مقایسه با `upgma` نتایج بهتری به ما می دهد. (خوشه ها ایجاد شده کانالهای نزدیک به هم را بهتر پوشش می دهند.) علت این امر می تواند وزن دار بودن محاسبات در این الگوریتم باشد.
- باشهودی که خود دارید و یا با جست و جو در اینترنت به دنبال این بردید که چه ارتباط بین الکترودها میتواند برقرار باشد؟ آیا مثلا الکترودهای سمت چپ از سمت راست مستقل هستند؟

انتظار داریم که الکترودهای موجود در نواحی مجاور عملکرد مشابهی داشته باشند و در یک خوشه قرار بگیرند. همچنین نمی توان گفت الکترودهای سمت چپ و راست کاملا از هم مستقلند بلکه ارتباط آنها با یکدیگر به کار صورت گرفته توسط شخص بستگی دارد اما در حالت کلی انتظار نداریم که تک الکترودی از سمت چپ با الکترودی از سمت راست در یک خوشه قرار بگیرد و انتظار داریم خوشه ها الکتروهای نزدیکتر به هم از نظر عملکرد و مکان را پوشش دهند.

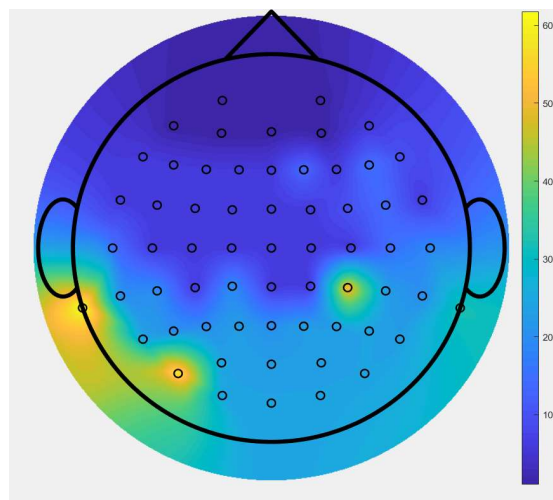
- با انجام خوشه بندی به ازای تعداد مراحل مختلف و با استفاده از هردو الگوریتم، می بینیم که در 57 مرحله داده ها تنها به یک خوشه تبدیل می شوند که مطلوب نیست. 55 مرحله بهینه ترین تعداد مراحل است زیرا با مشاهده خوشه ها با استفاده از تابع `topography`، می بینیم که کانالهای مجاور در یک خوشه قرار گرفته اند. شکل 1-3 و 2-3 نتایج خوشه بندی توسط تابع `correlationCluster` نشان می دهد.



شکل 3-1) خروجی ('wpgma', 55, $1 - \text{abs}(r)$)



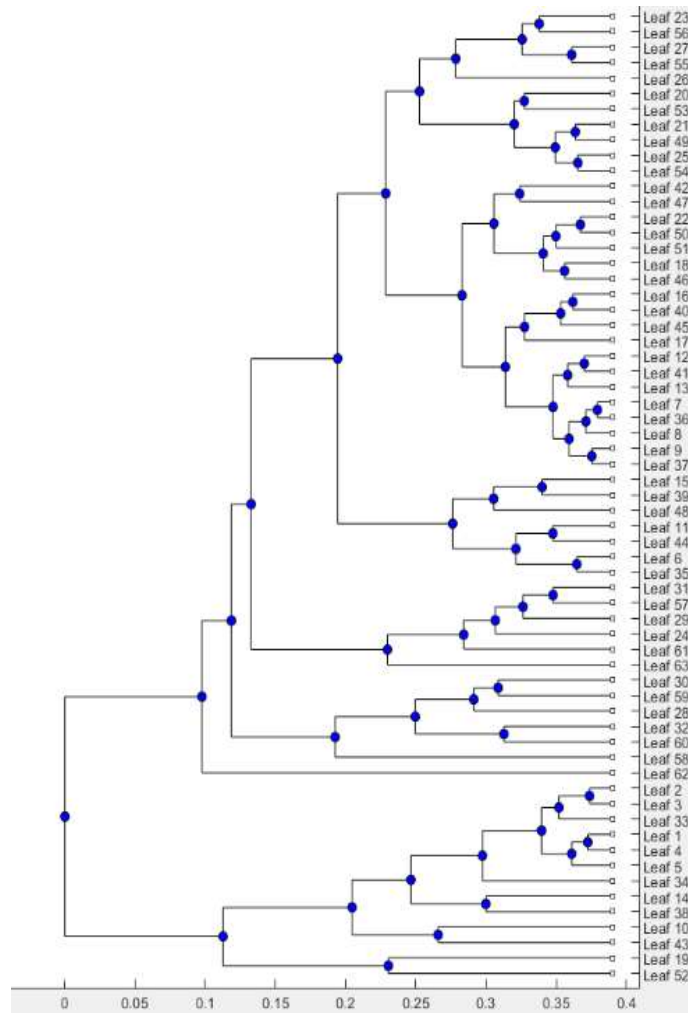
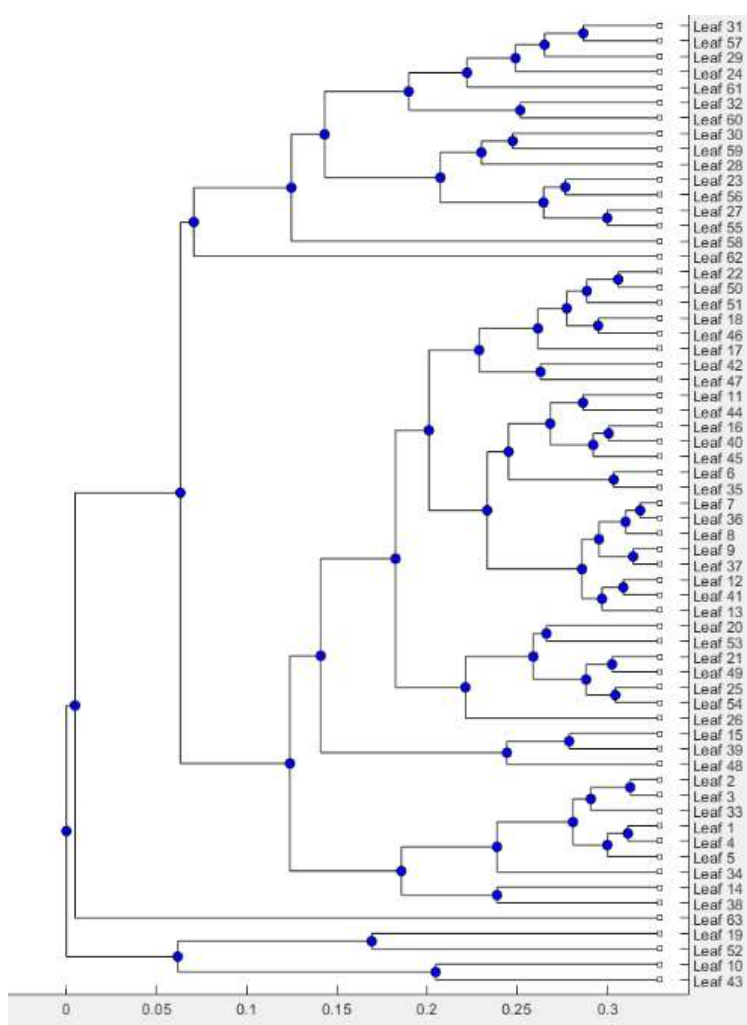
شکل 3-2) خروجی ('upgma', 55, $1 - \text{abs}(r)$)



شکل 3-3) خروجی ('upgma', 50, $1 - \text{abs}(r)$)

نتایج حاصل از دو الگوریتم همانطور که در شکلهای 1-3 و 2-3 می بینیم مشابه است. در شکل 3-3 خوشه بندی تا 50 مرحله اتفاق افتاده که پیداست این تعداد مرحله مطلوب نیست زیرا الکتروود 52 با الکترودهای سمت راست در یک خوشه قرار گرفته است در حالیکه در عملکرد و ناحیه اشتراکی با آنها ندارد و الکترودهای مجاور الکتروود 52 خود در یک خوشه قرار گرفته اند.

برای بررسی صحت نتایج با استفاده از تابع `seqlinkage` متلب، نمودار درختی 63 الکتروود به صورت زیر می باشد شکل سمت راست خروجی تابع را به ازای الگوریتم `upgma` و شکل سمت چپ خروجی تابع به ازای الگوریتم `wpgma` را نشان می دهد:



طراحی فیلتر

- فرض می کنیم اندازه فیلتر برابر ثابت k باشد و می دانیم که برای سیستم LTI با تابع تبدیل $H(s)$ رابطه زیر برقرار است:

$$\cos(\omega t) \longrightarrow \boxed{H(s)} \longrightarrow |H(j\omega)| \cos(\omega t + \angle H(j\omega))$$

بنابراین برای فیلتر با فاز $\phi(\omega) = -\frac{\pi}{3} \text{sgn}(\omega)$ خواهیم داشت:

$$x(t) = \cos(\omega_0 t) + \cos(2\omega_0 t) \rightarrow y(t) = k \cos\left(\omega_0 t - \frac{\pi}{3} \text{sgn}(\omega_0)\right) + k \cos\left(2\omega_0 t - \frac{\pi}{3} \text{sgn}(\omega_0)\right)$$

$$\blacksquare \text{ if } \omega_0 > 0 : y(t) = k \cos\left(\omega_0 t - \frac{\pi}{3}\right) + k \cos\left(2\omega_0 t - \frac{\pi}{3}\right)$$

$$= k \left[\frac{1}{2} \cos(\omega_0 t) + \frac{\sqrt{3}}{2} \sin(\omega_0 t) + \frac{1}{2} \cos(2\omega_0 t) + \frac{\sqrt{3}}{2} \sin(2\omega_0 t) \right]$$

$$= k \left[\frac{1}{2} \cos(\omega_0 t) + \frac{\sqrt{3}}{2} \sin(\omega_0 t) + \cos^2(\omega_0 t) - \frac{1}{2} + \sqrt{3} \sin(\omega_0 t) \cos(\omega_0 t) \right]$$

$$= k \left[\cos(\omega_0 t) \left(\frac{1}{2} + \cos(\omega_0 t) \right) + \sqrt{3} \sin(\omega_0 t) \left(\frac{1}{2} + \cos(\omega_0 t) \right) - \frac{1}{2} \right]$$

$$= k \left[(\cos(\omega_0 t) + \sqrt{3} \sin(\omega_0 t)) \left(\frac{1}{2} + \cos(\omega_0 t) \right) - \frac{1}{2} \right]$$

$$= k \left[\left(\frac{1}{2} \cos(\omega_0 t) + \frac{\sqrt{3}}{2} \sin(\omega_0 t) \right) (1 + 2\cos(\omega_0 t)) - \frac{1}{2} \right] = k \sin\left(\omega_0 t + \frac{\pi}{6}\right) (1 + 2\cos(\omega_0 t)) - k/2 \blacksquare$$

$$\blacksquare \text{ if } \omega_0 < 0 : y(t) = k \cos\left(\omega_0 t + \frac{\pi}{3}\right) + k \cos\left(2\omega_0 t + \frac{\pi}{3}\right)$$

$$= k \left[\frac{1}{2} \cos(\omega_0 t) - \frac{\sqrt{3}}{2} \sin(\omega_0 t) + \frac{1}{2} \cos(2\omega_0 t) - \frac{\sqrt{3}}{2} \sin(2\omega_0 t) \right]$$

$$\begin{aligned}
&= k \left[\frac{1}{2} \cos(\omega_0 t) - \frac{\sqrt{3}}{2} \sin(\omega_0 t) + \cos^2(\omega_0 t) - \frac{1}{2} - \sqrt{3} \sin(\omega_0 t) \cos(\omega_0 t) \right] \\
&= k \left[\cos(\omega_0 t) \left(\frac{1}{2} + \cos(\omega_0 t) \right) - \sqrt{3} \sin(\omega_0 t) \left(\frac{1}{2} + \cos(\omega_0 t) \right) - \frac{1}{2} \right] \\
&= k \left[(\cos(\omega_0 t) - \sqrt{3} \sin(\omega_0 t)) \left(\frac{1}{2} + \cos(\omega_0 t) \right) - \frac{1}{2} \right] \\
&= k \left[\left(\frac{1}{2} \cos(\omega_0 t) - \frac{\sqrt{3}}{2} \sin(\omega_0 t) \right) (1 + 2 \cos(\omega_0 t)) - \frac{1}{2} \right] = k \sin \left(\omega_0 t + \frac{5\pi}{6} \right) (1 + 2 \cos(\omega_0 t)) - k/2 \blacksquare
\end{aligned}$$

در صورتیکه $\phi(\omega) = -\frac{\pi}{3}\omega$ ، داریم:

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

■ if $\omega_0 > 0$ or $\omega_0 < 0$:

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

در فیلتر اول با تغییر علامت فرکانس، می بینیم که فاز خروجی به سرعت تغییر پیدا می کند. از طرفی تغییرات زمان در ورودی منجر به تغییرات زمانی مشابه در خروجی نمی شود و این منجر به غیر خطی شدن رفتار خروجی و اعوجاج آن خواهد شد:

$$x(t) = \cos(\omega_0 t) + \cos(2\omega_0 t) \rightarrow \begin{cases} \omega_0 > 0 : k \sin \left(\omega_0 t + \frac{\pi}{6} \right) (1 + 2 \cos(\omega_0 t)) - k/2 \\ \omega_0 < 0 : k \sin \left(\omega_0 t + \frac{5\pi}{6} \right) (1 + 2 \cos(\omega_0 t)) - k/2 \end{cases}$$

$$x'(t) = x(t + t_0) \xrightarrow{\times} y'(t) = y(t + t_0)$$

در فیلتر دوم نسبت تغییرات زاویه به تغییرات فرکانس عددی ثابت بوده و خروجی با آهنگی یکنواخت در زمان شیفت پیدا می کند. همچنین هر تک فرکانس رفتار خطی دارد و شیفت زمانی ورودی و خروجی با هم متناسب است.

$$x_1(t) = \cos(\omega_0 t) : x'_1(t) = x_1(t + t_0) \rightarrow y_1(t) = k \cos \left(\omega_0 \left(t - \frac{\pi}{3} \right) \right) : y'_1(t) = y_1(t + t_0)$$

$$x_2(t) = \cos(2\omega_0 t) : x'_2(t) = x_2(t + t_0) \rightarrow y_2(t) = k \cos \left(\omega_0 \left(2t - \frac{\pi}{3} \right) \right) : y'_2(t) = y_2(t + t_0)$$

- مطابق تعریف group delay، درستی گزاره ی "برای محتوای هر فرکانس، نشان گر این است که پس از عبور از فیلتر، چه مقدار تأخیر (برحسب سمپل یا ثانیه) خواهد یافت." را بررسی می کنیم.

$$gd_1 = -\frac{d\phi_1(\omega)}{d\omega} = \frac{2\pi}{3}\delta(\omega) \quad , \quad gd_2 = -\frac{d\phi_2(\omega)}{d\omega} = \frac{\pi}{3}$$

ورودی شامل دو تک فرکانس است و برای نشان دادن مقدار تأخیر بر حسب ثانیه، فرض می کنیم ورودی نیز به اندازه t_0 تأخیر پیدا کرده است و خروجی را بدست می آوریم:

$$x'(t) = x(t + t_0) = \cos(\omega_0(t + t_0)) + k \cos(\omega_0(2t + 2t_0))$$

در فیلتر اول داریم:

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

$$\rightarrow y'(t) = y(t + t_0) = k \cos\left(\omega_0(t + t_0) \pm \frac{\pi}{3}\right) + k \cos\left(\omega_0(2t + 2t_0) \pm \frac{\pi}{3}\right)$$

تا زمانی که علامت ω_0 تغییر نکند تأخیر فاز ثابت $\pm\pi/3$ است اما در $\omega = 0$ به دلیل تغییر علامت فاز به اندازه $2\pi/3$ جهش داشته و مستقل از تأخیر زمانی ورودی است.

در فیلتر دوم داریم:

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

$$\rightarrow y'(t) = y(t + t_0) = k \cos\left(\omega_0(t + t_0 - \frac{\pi}{3})\right) + k \cos\left(\omega_0(2t + 2t_0 - \frac{\pi}{3})\right) = kx\left(t + t_0 - \frac{\pi}{3}\right)$$

در این حالت می بینیم که خروجی دقیقاً به اندازه $\frac{\pi}{3}$ شیفت داده شده است.

- به اثبات رابطه داده شده برای تأخیر گروه می پردازیم:

$$H(w) = A(w)e^{j\phi(w)} \rightarrow \frac{d[H(w)]}{dw} = A(w) \cdot \frac{d[e^{j\phi(w)}]}{dw} + e^{j\phi(w)} \cdot \frac{d[A(w)]}{dw}$$

$$\xrightarrow{\text{تقسیم طرفین بر } A(w)} \frac{d[H(w)]/dw}{A(w)} = \frac{d[e^{j\phi(w)}]}{dw} + \frac{e^{j\phi(w)}}{A(w)} \cdot \frac{d[A(w)]}{dw}$$

$$= j e^{j\phi(w)} \frac{d[\phi(w)]}{dw} + \frac{e^{j\phi(w)}}{A(w)} \cdot \frac{d[A(w)]}{dw}$$

$$\xrightarrow{\text{تقسیم طرفین بر } e^{j\phi(w)} \text{ و ضرب در } j} j \frac{d[H(w)]/dw}{A(w)e^{j\phi(w)}} = -\frac{d[\phi(w)]}{dw} + j \frac{1}{A(w)} \cdot \frac{d[A(w)]}{dw} \quad \text{رابطه (4)}$$

با توجه به روابط $gd(w) = -\frac{d[\phi(w)]}{dw}$ و (4) خواهیم داشت:

$$gd(w) = \text{Re} \left\{ j \frac{d[H(w)]/dw}{A(w)e^{j\phi(w)}} \right\} = \text{Re} \left\{ j \frac{d[H(w)]/dw}{H(w)} \right\} \quad \text{رابطه (5)}$$

از طرفی می دانیم که :

$$j \frac{d[H(w)]}{dw} = \text{DTFT}\{nh[n]\}$$

$$H(w) = A(w)e^{j\phi(w)} = \text{DTFT}\{h[n]\}$$

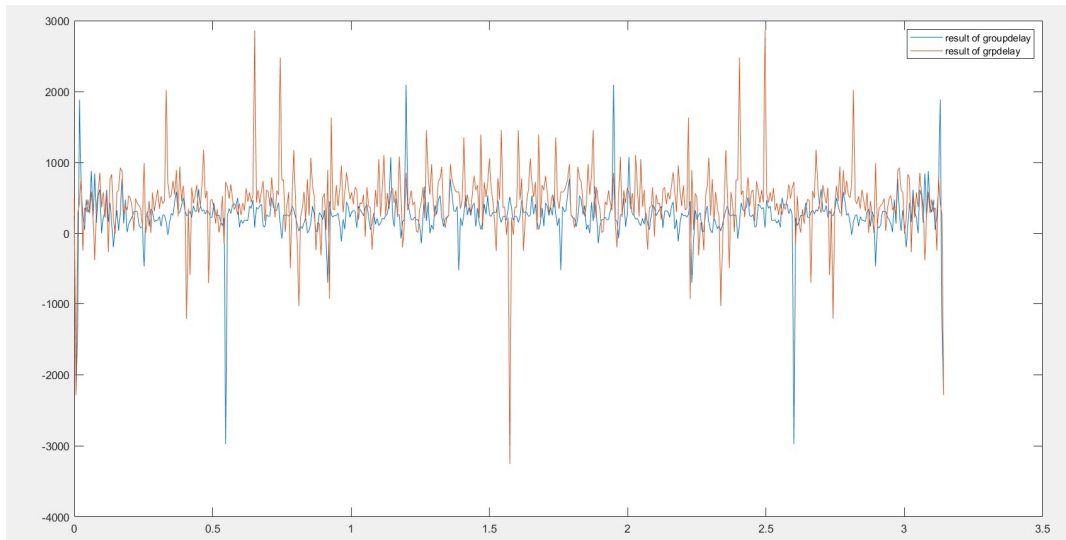
بنابراین می توان رابطه 4 و در نتیجه رابطه 5 را به صورت زیر نوشت:

$$\frac{\text{DTFT}\{nh[n]\}}{\text{DTFT}\{h[n]\}} = gd(w) + j \frac{1}{A(w)} \cdot \frac{d[A(w)]}{dw}$$

$$gd(w) = \text{Re} \left\{ \frac{\text{DTFT}\{nh[n]\}}{\text{DTFT}\{h[n]\}} \right\} = \text{Re} \left\{ \frac{\text{DFT}\{nh[n]\}}{\text{DFT}\{h[n]\}} \right\}$$

- با استفاده از رابطه بدست آمده برای تاخیر گروه برحسب fft، تابع groupdelay را می نویسیم که ورودی های آن h به عنوان سیگنال ورودی و N جهت نمونه برداری است.

هر چه تعداد نمونه ها بیشتر باشد DFT با دقت بهتری تغییرات فرکانس را نشان داده و تقریب بهتری از تبدیل فوریه پیوسته به ما می دهد. شکل زیر نمونه ای از خروجی این تابع و مقایسه آن با تابع grpdelay متلب را نشان می دهد.



شکل 4-1) خروجی تابع groupdelay به ازای N=512 و h=rand(1000,1)

- تابع $\text{zphasefilter}(h,x,N)$ نوشته شده است که با استفاده از تابع groupdelay ابتدا تاخیر گروه فیلتر h را محاسبه کرده و براساس رابطه :

$$x[n - n_0] \leftrightarrow e^{-j\Omega n_0} X(\Omega)$$

عمل می کند. بدین صورت که n_0 ها را همان groupdelay در نظر می گیریم و سپس از این رابطه ifft می گیریم.

- آیا فیلتری علی و حقیقی وجود دارد که فاز آن در تمام فرکانس ها صفر باشد؟
بله، در این صورت کافی است فیلتری با اندازه ثابت و فرکانس صفر را در نظر بگیریم که خروجی همان ورودی بدون تاخیر فاز خواهد بود و تنها در اندازه فیلتر ضرب می شود.

شناسایی کلمات

- در حالت کلی از آنجایی که اطلاعات ردیف دهم هر ماتریس، نشان دهنده سطر، ستون و یا شماره حرف روشن شده است؛ بنابراین اگر در هر دیتاست، در سر دهم عددی بزرگتر از 12 (مجموع تعداد سطر ها و ستون ها حالت RC) داشتیم به این معناست که این دیتاست در حالت SC گرفته شده است.

با توجه به توضیحات مذکور پارادایم ها برای هر یک از نفرات به شرح جدول زیر است :

Person's No.	1	2	3	5	6	7	8	9
paradigm	SC	SC	RC	RC	RC	RC	RC	RC

جدول 1-5) نوع آزمایش هر فرد

- در حالت اول و دوم که پارادایم SC است کافی است بررسی کنیم سطر یازدهم هر ماتریس train در پاسخ به کدام عدد ها در سطر دهم، یک شده است؛ با حذف داده های تکراری به کمک دستور unique ماتریس حرف های مورد نظر را می یابیم. برای نفر اول و دوم داریم :

letter	L	U	K	A	S
No.	12	21	11	1	19

جدول 2-5) شماره متناظر با هر حرف

بنابراین شماره ها از 1 تا 36 به ترتیب مربوط به حروف الفبا و اعداد 0 تا 9 هستند.

برای پارادایم RC باید جفت سطر و ستون را برای هر حرف جدا کنیم؛ اگر شماره های 6-1 را برای سطر و شماره های

12-8 را برای ستونها در نظر بگیریم، برای نفر سوم تا نهم داریم :

letter	row	column
L	6	8
U	3	10
K	5	8
A	1	7
S	1	10

جدول 3-5) شماره سطر و ستون متناظر با هر حرف

به این ترتیب حروف و اعداد در این پارادایم، شماره های سطر و ستونی به شرح زیر دارند :

Row\Column	7	8	9	10	11	12
1	A	G	M	S	Y	4
2	B	H	N	T	Z	5
3	C	I	O	U	0	6
4	D	J	P	V	1	7
5	E	K	Q	W	2	8
6	F	L	R	X	3	9

جدول (4-5) شماره سطر و ستون متناظر با حروف و اعداد در آزمایش RC

- برای نوشتن تابع IndesExtraction کافی است با استفاده از دستور find ، مکان تحریکات را در سطر دهم داده های train بیابیم و در سطر بردار index قرار دهیم. اگر بردار index را به عنوان ورودی به سطر اول داده های train بدهیم؛ زمان تحریکات محاسبه میشود یا زمان ها را در سطر اول ماتریس time ذخیره می کنیم. همچنین اگر به سطر یازدهم داده ها، بردار index را به عنوان ورودی بدهیم؛ target یا non-target بودن تحریک مشخص میشود. این داده ها را در سطر دوم ماتریس time و روبه روی هر تحریکی ذخیره می کنیم.

پیاده سازی الگوریتم یادگیری ماشین

- حال برای هر یک از افراد با استفاده از دو الگوریتم SVM و LDA، دو مدل را به صورت زیر آموزش میدهم :

• SVM (Support Vector Machines)

برای آموزش مدل از دستور fitcsvm، استفاده می کنیم. به عنوان ورودی باید یک ماتریس از ویژگی ها و یک بردار به عنوان نتیجه حاصل شده از این ویژگی را به دستور بدهیم. ماتریس ویژگی ها را در اینجا از طریق ماتریس epoch شده داده ها محاسبه می کنیم. به این ترتیب به عنوان ویژگی ها (ستون های ماتریس) زمان _طبق پنجره بندی ای که از زمان داشتیم یعنی 200 میلی ثانیه قبل از هر تحریک و 800 میلی ثانیه پس از هر تحریک _ و به عنوان مشاهدات (observations / سطر ها) داده های سیگنال هر کانال در این بازه زمانی را قرار میدهم. به این ترتیب ماتریس سه بعدی epoch را برای استفاده در این بخش به یک ماتریس دو بعدی تبدیل کردیم. به عنوان بردار label (نتایج) نیز نتیجه متناظر با هر زمان که در سطر یازدهم داده های train ذکر شده است را قرار می دهیم.

دو ورودی ساخته شده را به دستور fitcsvm داده و مدل را آموزش می دهیم.

• LDA (Discriminant Analysis Classification)

برای آماده سازی داده ها نسبت به الگوریتم قبلی، نیاز به انجام مرحله اضافه تری نداریم. کافس است ورودی های آماده شده را به دستور `fittediscr` بدهیم و مدل را آموزش دهیم.

- از آنجایی که در `classification` حاضر، تعداد اعضای یک کلاس (`non-target`) بسیار بیشتر از تعداد اعضای کلاس دیگر (`target`) که از قضا کلاس مهمتری نیز هست؛ است بنابراین کلاس بندی ما برای آموزش مدل بسیار نامتعادل است. معمولا در مواقع این چنینی تابع `predict` همه حالات را به عنوان `non-target` تشخیص می دهد ولی چون تعداد خطا در تشخیص اندک است (به خاطر تعداد اندک `target` ها نسبت به کل) درصد صحت پیش بینی بالا تلقی می شود.
- یکی از راه های پیشگیری از این امر، تعریف ماتریس هزینه به گونه ای است که هزینه تشخیص غلط بالاتر باشد؛ تعریف هر یک از درایه های ماتریس هزینه به صورت مقابل است.

تشخیص نادرست <code>target</code> به عنوان <code>non-target</code>	تشخیص درست <code>target</code> به عنوان <code>target</code>
تشخیص درست <code>non-target</code> به عنوان <code>non-target</code>	تشخیص نادرست <code>non-target</code> به عنوان <code>target</code>

جدول 1-6) ماتریس هزینه

بنابراین درایه های روی قطر را برابر با 0 و هزینه درایه (1,2) را بیشتر از درایه (2,1) تنظیم می کنیم. این ماتریس را نیز به عنوان ورودی به دو دستور `fitsvm` و `fittediscr` می دهیم.

- در نهایت باید از روی پیش بینی انجام شده برای نتایج کلمه گفته شده را تشخیص دهیم. برای این منظور تابع `WordDetection` را به صورت مقابل می نویسیم :

باید برای تشخیص کلمه در روش `SC` و `RC` به دو روش مختلف عمل کنیم؛ برای روش `SC`، ابتدا یک `string` از حروف از `A-Z` و اعداد از `0-9` تشکیل می دهیم. با دستور `split` این رشته را به بردار تبدیل می کنیم. سپس اندیس هایی را که در بردار نتایج پیش بینی شده؛ برابر با 1 هستند با دستور `find` می یابیم. مکان این تحریک ها را با استفاده از بردار `stimuliOnset` که در گذشته محاسبه کردیم (مکان تحریکات `target`، که تحریک ماقبل آنها `non-target` است)

پیدا می کنیم. حال کافی است در سطر دهم دیتاهای `train`، شماره روشن شده معادل با این مکان ها را بیابیم. این مکان ها را در بردار `word` ذخیره می کنیم. از آنجایی که اعداد به ترتیب مربوط به `A-Z` و `0-9` هستند کافی است درایه های متناظر با این شماره ها را از برداری که از حروف و اعداد تشکیل دادیم در کنار هم دهیم و کلمه را بازیابی کنیم.

برای روش `RC`، 6 رشته متناظر با اعضای ستون های جدول ... می سازیم و به بردار تبدیل می کنیم. از آنجایی که هر حرف 15 بار تکرار می شود کافی است برای به دست آوردن سطر و ستون هر حرف بردار شماره های `target` را (که در بالا به دست آوردیم) با پنجره های 30 تایی بپیماییم و با دستور `unique` دو درایه نشانگر سطر و ستون را بیابیم (با این کار المانهای تکراری را حذف می

کنیم). سپس دو داده را sort میکنیم و داده بزرگتر را اول و داده کوچکتر را بعد از آن در word قرار می دهیم. در نهایت با توجه به 6 برداری که از حروف و اعداد تعریف کردیم و با استفاده از یک switch/case روی شماره ستونها، کلمه را بازیابی می کنیم.

نتایج تشخیص کلمه برای داده های train، مطابق انتظار برابر با 'LUKAS' است.

- در کد پروژه سعی شد که با استفاده از تغییر ماتریس هزینه، oversampling داده ها با الگوریتم smote و مقابله با مشکل overfitting به کمک cross validation و دستور crossval، کلمه گفته شده در دیتاست test نیز تشخیص داده شود. گرچه حاصل تابع predict برای این دیتاست (با و بدون اعمال الگوریتم های مذکور) حاوی شماره هایی به عنوان شماره های target بود؛ اما در بردار حاصل، حرفی با تکرار منظم 15 بار (ذکر شده در سوال) وجود نداشت و بنابراین کلمه صحیح تشخیص داده نشد. به هر روی کد الگوریتم smote استفاده شده نیز در متن کد پروژه قرار داده شده اما به علت جلوگیری از نادرست شدن بقیه پیش بینی ها از اعمال آن صرف نظر شده است.