# فصل سوم پیاده‌سازی نرم‌افزار

## مقدمه

در این فصل به شرح فرایند پیاده‌سازی نرم‌افزار کنترل دستگاه الکتروانسفالوگرام، ثبت و ذخیره‌سازی داده‌های آن پرداخته شده است. سپس ساختار پایگاه داده که توسط این نرم‌افزار و دستگاه جمع‌آوری شده‌اند مورد برسی قرار گرفته است.

انسان با استفاده از حواس خود درباره جهان اطراف اطلاعاتی کسب می‌کند. بینش انسان یکی از مهم‌ترین بخش‌های ارتباط او با محیط پیرامون است. انسان‌ها در طول زندگی یاد می‌گیرند که با اجسام مختلف ارتباط برقرار کنند. برای برخی از آن‌ها یادگیری چند ثانیه طول می‌کشد (مانند استفاده از دست‌گیره‌ی در)؛ لیکن برخی دیگر به زمان بیشتری نیاز دارند (مانند رانندگی با ماشین).

با پیدایش رایانه انسان‌ها مجبور به تفکر انتزاعی شدند، بدین‌صورت که مجبور شدند با حجم بیشتری از دستوراتی که حتی قادر به یادآوری آن‌ها نبوده، برخورد کنند؛ بنابراین طراحی یک رابط کاربری گرافیکی[[1]](#footnote-1) (GUI) اهمیت پیدا کرد تا تعامل بین رایانه و انسان را به ساده‌ترین شکل، ممکن سازد؛ بنابراین، رابط کاربری گرافیکی یکی از عناصر مهم ارتباط انسان‌ها با دنیای مدرن و تکنولوژی است.

باتوجه‌به این پیش‌زمینه، طراحی یک GUI با کاربری آسان برای نرم‌افزارها از اهمیت بالایی برخوردار است. در این پژوهش باتوجه‌به گزینه‌ها و زبان‌های برنامه‌نویسی موجود، برای طراحی یک GUI برای کنترل‌کننده‌ی هدبند الکتروانسفالوگرام خواب، زبان برنامه‌نویسی پایتون انتخاب شد، زیرا الگوریتم‌های هوش مصنوعی و کتابخانه‌های رسم نمودار پیشرفته و درعین‌حال رایگان برای آن توسعه داده شده است که می‌توان در رابط کاربری گرافیکی هم به نحوی از آن‌ها استفاده کرد.

## معرفی پای‌کیوت

پای‌کیوت[[2]](#footnote-2) یک کتابخانه پایتون است که به کیوت[[3]](#footnote-3) متصل است. کیوت مجموعه‌ای از کتابخانه‌های سی پلاس پلاس و ابزارهای توسعه است که شامل توابع مستقل از نوع سیستم‌عامل برای پیاده‌سازی رابط کاربری گرافیکی، شبکه، ریسمان اجرایی[[4]](#footnote-4)، پایگاه‌های داده اس‌کیوال[[5]](#footnote-5)، نگاره‌سازی برداری مقیاس‌پذیر[[6]](#footnote-6)، اوپن‌جی‌ال[[7]](#footnote-7) و زبان نشانه‌گذاری گسترش‌پذیر[[8]](#footnote-8) است و همچنین بسیاری از ویژگی‌های قدرتمند دیگر را شامل می‌شود. پای‌کیوت توسعه‌یافته توسط شرکت محاسباتی ریوربانک[[9]](#footnote-9)، در دو نسخه موجود است:

1.پای‌کیوت ۴: نسخه‌ای که در برابر Qt 4.x و 5.x ساخته شده است.

2.پای‌کیوت ۵: نسخه‌ای که فقط در برابر Qt 5.x ساخته شده است.

اگرچه پای کیوت 5 بر پایه نسخه 4 ساخته شده، تنها یک زیرمجموعه کوچک از نسخه 5 با نسخه 4 سازگار است و توسط آن پشتیبانی می‌شود. این بدان معناست که در صورت استفاده از پای کیوت 4، برخی از ویژگی‌ها و پیشرفت‌های جدید در نسخه 5 در دسترس نخواهد بود. برای اطلاعات بیشتر در این زمینه به اسناد پای کیوت 4 مراجعه شود [1].

پای‌کیوت 5 مبتنی بر نسخه 5 کیوت است و شامل کلاس‌هایی است که رابط کاربری گرافیکی و همچنین مدیریت XML، ارتباطات شبکه‌ای، عبارات منظم، موضوعات، پایگاه‌های داده SQL ، چندرسانه‌ای، مرور وب و سایر فناوری‌های موجود در کیوت را پوشش می‌دهد. پای‌کیوت 5 بیش از هزار کلاس از کلاس‌های کیوت را در مجموعه‌ای از ماژول‌های پایتون پیاده‌سازی می‌کند که همه آن‌ها در یک بسته سطح بالای پایتون به نام پای‌کیوت 5 قرار دارند [1].

پای کیوت 5 با سیستم عامل‌های ویندوز، یونیکس، لینوکس، مک، اندروید و آی او اس سازگار است. بر اساس این ویژگی، با استفاده از این کتابخانه می‌توان برنامه را روی چند سیستم عامل و دستگاه متفاوت با موفقیت اجرا کرد [1].

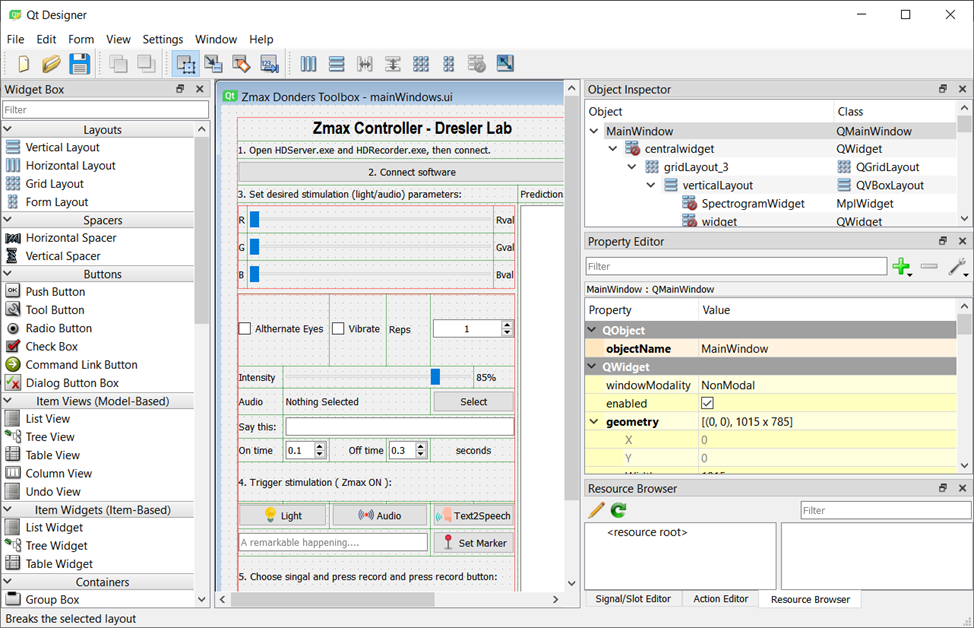
## طراح کیوت[[10]](#footnote-10): نرم‌افزار طراحی رابط گرافیکی

ساخت برنامه با استفاده از پای‌کیوت ممکن است به روش‌های مختلفی صورت گیرد. در هر روش می‌توان بسته به شرایط و علایق شخصی از ابزارهای مختلفی استفاده کرد.

ولی شاید راحت‌ترین و پرکاربردترین روش، استفاده از طراح ‌کیوتی (Qt Designer) باشد. با استفاده از طراح کیوتی می‌توان پیچیده‌ترین پنجره‌ها را به‌سادگی هرچه تمام ایجاد کرد. محیط کاربری رابط این نرم‌افزار در شکل 1 نمایش داده شده است.

طراحی نرم‌افزار در محیط طراح ‌کیوتی شامل مراحل زیر است:

1. ایجاد یک فایل رابط گرافیکی با استفاده از طراح کیوتی (Qt Designer)
2. تخصیص نام و مقداردهی مشخصه‌های مختلف در ویرایشگر صفات طراح کیوتی
3. استفاده از pyuic4 برای ایجاد کلاس پایتون (فایل py) از فایل رابط گرافیکی ایجاد شده
4. ایجاد و ساخت تابع اصلی برنامه و فراخوانی کلاس رابط گرافیکی
5. گسترش کلاس ایجاد شده موجود با توابع و کدهای پایتون



شکل 1 - پیاده‌‌سازی نرم‌افزار در محیط Qt Designer

در این پژوهش باتوجه‌به نیازها و انتظاراتی که از یک کنترل‌کننده‌ی هدبند خواب وجود داشت، یک رابط کاربری برای نرم‌افزار طراحی و پیاده‌سازی شد. هر کلید و دکمه در این رابط به متغیری در کد برنامه متصل است و اطلاعات بین رابط کاربری، کد و هدبند دایم در حال تعامل هستند.

## پیاده‌سازی نرم‌افزار کنترل هدبند

باتوجه‌به نیازها و انتظارات محققین خواب و حافظه مبنی بر امکانات نرم‌افزار کنترلی هدبند الکتروانسفالوگرام، نرم‌افزاری در محیط پای‌کیوت پیاده‌سازی شد. این نیازها شامل:

1. دریافت سیگنال‌های حیاتی از دستگاه و نمایش آن‌ها به‌صورت زمان مستقیم
2. ذخیره داده‌های دریافتی به‌صورت کم‌حجم و قابل بازیابی
3. ارسال دستور شروع محرک صوتی یا نوری به دستگاه و تنظیم شدت و فرکانس محرک
4. ذخیره‌سازی زمان محرک‌های ارسالی هم‌زمان با داده‌های دریافتی
5. پردازش زمان مستقیم داده‌های الکتروانسفالوگرام
6. اجرای الگوریتم‌های مختلف به‌منظور طبقه‌بندی و تفسیر داده‌های دریافتی
7. رسم نمودار چگالی طیف توان به‌صورت زمان مستقیم
8. رسم چگالی طیفی بسامدهای سیگنال مغزی دریافتی به‌صورت زمان مستقیم

نحوه کار هدبند به این صورت است که داده‌هایی که توسط الکترودهای EEG اندازه‌گیری می‌شوند به صورت بی‌سیم و از طریق دانگل و ورودی یواس‌بی به رایانه انتقال داده می‌شوند. این داده‌ها به‌صورت یک‌رشته روی پورت localhost در TCP/IP موجود هستند. نرم‌افزار، این داده‌ها را از روی شبکه می‌خواند، آن را رمزگشایی کرده و تبدیل واحدهای مناسب را برای نمایش و ذخیره‌سازی روی آن‌ها انجام می‌دهد.

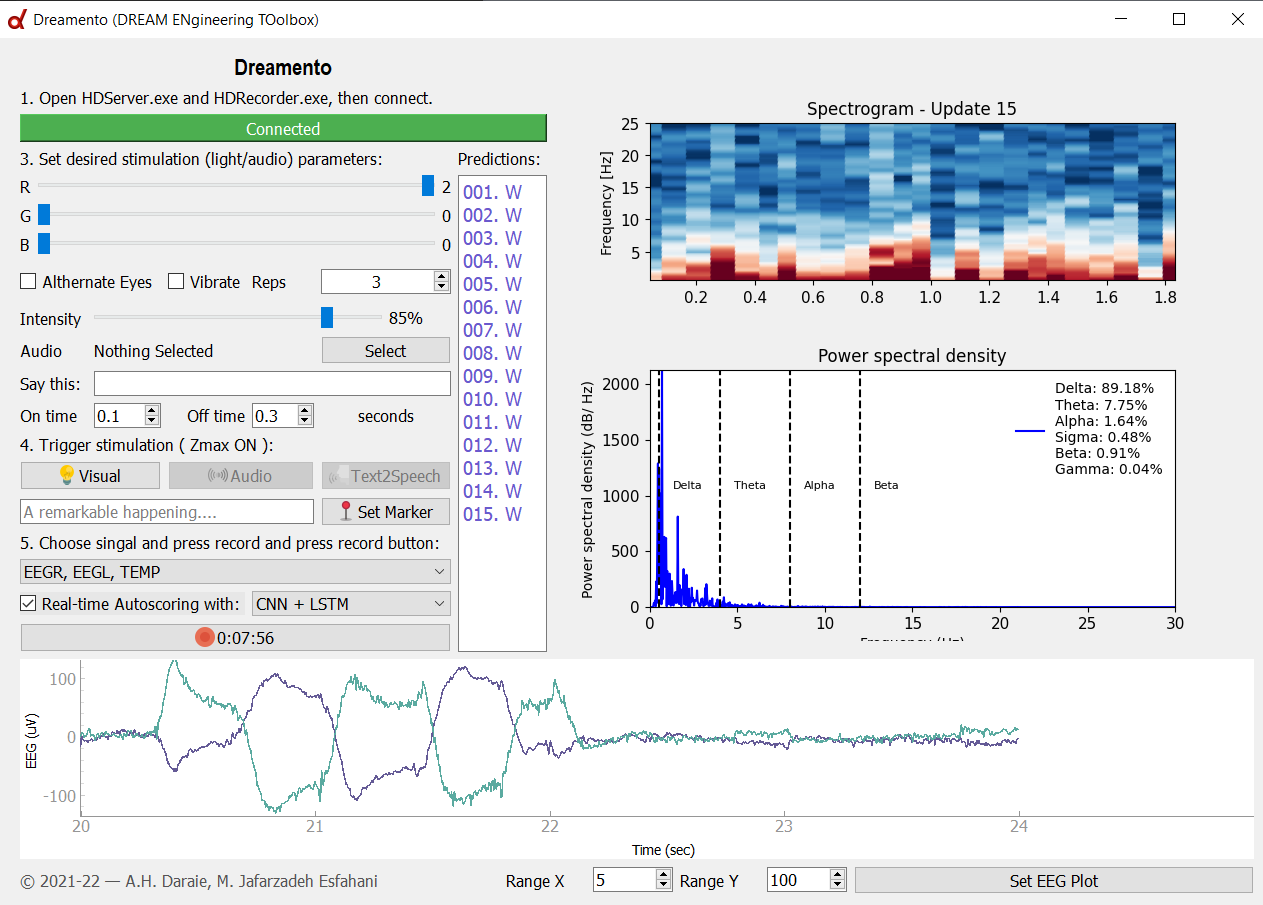
### توابع کنترل و ارتباط با شبکه

تمام قابلیت‌های مربوط به شبکه در فایل ZmaxSocket.py قرار گرفته است. این توابع شامل connect، read (از سوکت)، getbyteat، getwordat، ScaleEEG، ScaleAccel، BatteryVoltage، BodyTemp، hex2dec، dec2hex و در نهایت stimulate است. تابع connect نرم‌افزار را به شبکه متصل می‌کند و تابع read نیز داده‌های موجود روی شبکه را می‌خواند. تابع‌های scale نیز داده‌ها را به مقیاس درست می‌برند زیرا این داده‌ها از ADC میکروکنترلر هستند و باید به واحد مناسب و با معنا تبدیل شوند. برای مثال داده‌های الکتروانسفالوگرام باید به میکرو ولت تبدیل شوند.

تابع stimulate دستوراتی را به شبکه و در نهایت به هدبند فرستاده، پردازنده داخلی هدبند آن‌ها را تفسیر و سپس اجرا می‌کند. این دستورات باید به‌صورت رشته‌هایی از اعداد در مبانی هگزادسیمال کدگذاری شوند و سپس به شبکه ارسال شوند. در این تابع پارامترهایی از قبیل رنگ دیودهای نوری، شدت نور آن‌ها، فرکانس روشن و خاموش شدن آن‌ها و تعداد دوره‌های آن و همچنین پارامترهای مربوط به لرزش هدبند دریافت می‌شوند. سپس آن‌ها به اعدادی در مبانی هگزادسیمال تبدیل می‌شوند و به شبکه و هدبند ارسال می‌شود.

### توابع ارتباط با رابط کاربری

در مرحله بعد با داشتن این توابع کنترل هدبند و شبکه، توابعی به‌منظور کنترل رابط کاربری نرم‌افزار با قابلیت‌های موردنیاز برای آزمایش‌های خواب و حافظه طراحی و ساخته شد. رابط کاربری این نرم‌افزار در شکل 2 قابل‌مشاهده است.



شکل 2 - تصویر رابط کاربری نرم‌افزار

در اینجا با انتخاب کلید Connect، نرم‌افزار به شبکه متصل می‌شود. در این زمان، دانگل هدبند هم باید به پورت USB رایانه متصل باشد. سپس از فهرست انتخاب موجود می‌توان رنگ موردنظر برای دیودهای نوری آن، شدت نوری آن‌ها و نیز امکاناتی مانند لرزش یا پخش صوت را انتخاب کرد.

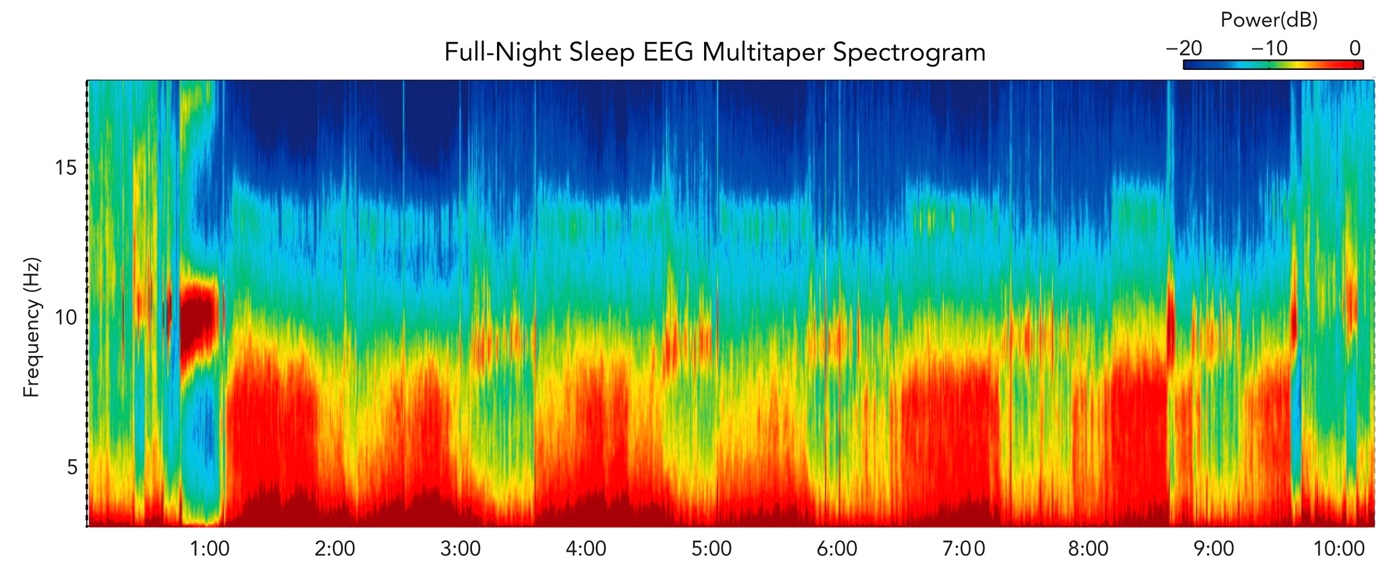
با زدن کلید Record، ثبت آغاز شده و داده‌های حس‌گرهای هدبند از قبیل الکتروانسفالوگرام، شتاب‌سنج و دما را ذخیره می‌کند. در انتها با زدن کلید پایان ضبط، این عملیات متوقف می‌شود و تمامی داده‌ها در فایلی با فرمت txt در مسیر نرم‌افزار ذخیره می‌شوند.

### رسم اسپکتروگرام

در طول خواب، شبکه‌های قشری و زیر قشری در مغز پویایی نوسانی ساختاریافته‌ای را از خود نشان می‌دهند که می‌توان آن را در سیگنال الکتروانسفالوگرام مشاهده کرد. توانایی توصیف دقیق تغییرات در حین خواب ناشی از این نوسانات، هدف اصلی در علم خواب بوده است.

در این راستا، یک ابزار قدرتمند برای تجزیه‌وتحلیل ساختار نوسانی EEG خواب، تجزیه‌وتحلیل طیفی بوده است که یک رویکرد کمی برای توصیف یک سیگنال از نظر نوسانات زیرین آن (سینوسوئیدها) در فرکانس‌های مختلف است. به این منظور با استفاده از تجزیه‌وتحلیل طیفی چندمخروطی[[11]](#footnote-11) نمودار طیفی سیگنال خواب به‌صورت زمان مستقیم محاسبه و نمایش داده می‌شود. وجود این نمودار می‌تواند به تجزیه‌وتحلیل و طبقه‌بندی خواب کمک شایانی کند.

طیف توان تابعی است که قدرت نوسانات را در هر فرکانس نشان می‌دهد. تجزیه‌وتحلیل طیفی همچنین شامل تجزیه‌وتحلیل نوسانات متغیر با زمان با استفاده از طیف‌نگار[[12]](#footnote-12) است که قدرت سیگنال را به‌عنوان تابعی از زمان و فرکانس ترسیم می‌کند و مشاهده چگونگی تغییر ساختار فرکانس در طول زمان را ممکن می‌سازد (مانند شکل 3)؛ بنابراین طیف‌نگارها برای توصیف تکامل زمانی نوسانات عصبی مرتبط با خواب در فرکانس‌های مختلف در طول یک شب ‌خواب مناسب هستند.



شکل 3 - نمونه طیف‌نگار چند مخروطی از دادگان مغزی خواب. برگرفته از [2]

برای نمایش داده‌ها در حوزه زمان فرکانس، می‌توان با تخمین طیف توان در زمان‌های مختلف با استفاده از یک پنجره متحرک داده، یک طیف‌نگار ساخت. در این روش اندازه پنجره داده طوری تنظیم می‌شود که ساختار داده‌ها در آن پنجره تقریباً ثابت باشد. یک طیف‌نگار با محور x نشان‌دهنده زمان، محور y نشان‌دهنده فرکانس، و توان طیفی نشان‌داده‌شده با رنگ در هر نقطه زمان فرکانس به تصویر کشیده می‌شود.

برخی از تکنیک‌های رایج برای تخمین طیفی EEG تخمین‌های نادرستی از طیف توان را تولید می‌کنند که تفسیر طیف‌نگار حاصل را دشوار می‌کند. در این پژوهش، از تخمین طیفی چندمخروطی که وضوح تخمین‌های طیفی را نسبت به روش‌های استاندارد تا حد زیادی افزایش می‌دهد، استفاده شده است. این روش در اوایل دهه ۱۹۸۰ توسط دیوید تامسون [3] توسعه یافت. نشان داده شده است که این روش دارای خواص آماری برتر در مقایسه با تخمین‌های طیفی تک‌مخروطی است [4]. نوآوری روش چندمخروطی این است که به جای استفاده از یک تابع تک‌مخروطی برای محاسبه طیف توان، از چندین تابع مخروطی برای محاسبه طیف‌های تک‌مخروطی استفاده می‌کند که با هم میانگین گرفته می‌شوند. این مخروطی‌ها از دسته خاصی از توابع به نام توالی کروی پرولات گسسته[[13]](#footnote-13) (DPSS) می‌آیند.

روش تخمین طیفی چندمخروطی شامل مراحل زیر است:

۱) مجموعه‌ای از مخروطی‌های DPSS را باتوجه‌به فرضیات داده ایجاد کنید.

۲) برای هر یک از مخروطی‌های DPSS، یک طیف تک‌مخروطی برای داده‌ها تخمین بزنید.

۳) تخمین طیفی چندمخروطی را تشکیل دهید.

درحالی‌که این روش ساده است، درک تصمیمات عملی که باید هنگام استفاده از تخمین طیفی چندتایی برای داده‌های واقعی مانند EEG گرفته شود، مهم است، به طوری که تجزیه و تحلیل‌ها اصولی بوده و از مفروضات منطقی پیروی کنند. در عمل، تخمین‌های طیفی چندمخروطی با چندین پارامتر تعریف می‌شوند که تعداد مخروطی‌های DPSS و خواص آن‌ها را کنترل می‌کنند. این پارامترها شامل N (اندازه هر بخش داده در واحد ثانیه)، TW (حاصل‌ضرب نیمه باند زمان[[14]](#footnote-14)) و L (تعداد مخروطی‌ها) می‌شوند. با درک این سه پارامتر، می‌توان به طور صریح ویژگی‌های زیربنای تخمین چندتایی را کنترل کرد.

برای انتخاب مقادیر این پارامترها، باید دو چیز را تعیین کرد: دوره زمانی که تصور می‌شود داده‌ها از نظر خواص فرکانسی و آماری ثابت هستند و وضوح طیفی موردنظر. ابتدا، اندازه بخش داده N باید به‌عنوان بیشینه مدت زمانی (بر حسب ثانیه) که در آن داده‌ها ثابت هستند، تعریف شود. سپس، وضوح طیفی () پهنای باند (بر حسب هرتز) لوب اصلی در تخمین طیفی است که کمینه فاصله بین قله‌هایی را که می‌توان در حوزه فرکانس داشت، کنترل می‌کند. در عمل، یک بزرگ قله‌های هموار و با وضوح پایین ایجاد می‌کند، در حالی که یک کوچک قله‌هایی با وضوح بالاتر و جزئیات بیشتر ایجاد می‌کند. با داشتن و N، اکنون می‌توان را محاسبه کرد (رابطه‌ی 1).

|  |  |
| --- | --- |
| (رابطه ۱) |  |

در نهایت، نیاز است ، یا تعداد مخروطی‌های استفاده شده در برآورد تعیین شود. با توجه به ، مقدار براساس رابطه ۲ بدست می‌آید.

|  |  |
| --- | --- |
| (رابطه ۲) |  |

روش انتخاب پارامترهای طیف چند مخروطی شامل 4 مرحله است: ۱) اندازه پنجره N را با تعیین مدت زمانی که تصور می‌شود سیگنال از نظر آماری و محتوای فرکانسی ثابت است، تنظیم کنید. ۲) وضوح فرکانسی موردنظر را باتوجه‌به ساختار نوسانی داده‌ها تنظیم کنید (تعیین ) ۳) حاصل‌ضرب نیمه پهنای باند زمان را محاسبه کنید. ۴) تعداد مخروطی‌ها را محاسبه کنید.

در این پژوهش، تصور می‌شود که سیگنال الکتروانسفالوگرام خواب در بازه‌های زمانی ۳۰ ثانیه‌ای ایستا است [2], [5]. بدین ترتیب مقدار N برابر با ۳۰ ثانیه تنظیم می‌شود. همچنین وضوح فرکانسی مطلوب برابر ۱ هرتز است، یعنی است. با توجه به رابطه 1، و در نتیجه بر اساس رابطه‌ی ۲، .

در سمت راست برنامه، قابی[[15]](#footnote-15) گذاشته شده است که در آن به‌صورت زمان مستقیم اسپکتروگرام داده‌های الکتروانسفالوگرام دریافتی نمایش داده می‌شود. برای رسم این اسپکتروگرام فرض شده است که فرکانس نمونه‌برداری 256 هرتز است.

در صفحه این نمودار، می‌توان اسپکتروگرام مربوط به 2 دقیقه گذشته را مشاهده کرد. درعین‌حال، با دریافت 30 ثانیه داده الکتروانسفالوگرام جدید، این نمودار به‌روزرسانی می‌شود و آخرین بخش از نمودار داده‌های گذشته به سمت منفی محور زمان جابجا می‌شود.

طیف‌نگار چندمخروطی نمایش واضحی از پویایی طیفی EEG خواب که در مقیاس‌های زمانی از چند ثانیه تا چند ساعت رخ می‌دهد، ارائه می‌کند. به‌این‌ترتیب، امید است که بتوان یک چارچوب برای توصیف فعالیت عصبی در طول خواب از نظر پویایی نوسانی و فعالیت شبکه به‌صورت زمان مستقیم با هدف تجزیه‌وتحلیل‌های تجربی و بالینی ارائه داد.

برای نشان‌دادن خروجی نتایج الگوریتم‌های طبقه‌بندی خواب، یک قاب در رابط کاربری برنامه تعریف شده و نتایج را به‌صورت شماره ایپاک (هر 30 ثانیه داده الکتروانسفالوگرام یک ایپاک است) و در کنار آن، نتیجه پیش‌بینی‌شده توسط الگوریتم نشان داده می‌شود.

## جداسازی پردازش‌ها به‌وسیله چندریسمانی (Multi Threading)

چندریسمانی به طور عمده در سیستم‌عامل‌های چند وظیفه‌ای پیدا می‌شود. چندریسمانی یک مدل گسترده در برنامه‌نویسی و اجرا است که اجازه می‌دهد چند ریسمان در یک فرایند اجرا شوند. برنامه‌ها همچنین می‌توانند باهدف بیشینه‌ی بهره‌گیری از قابلیت پردازش در سیستم‌های چندهسته‌ای طراحی شوند. چنین برنامه‌هایی می‌توانند چندین کار پردازنده‌ای سنگین را به طور موازی در چندین هسته محاسباتی انجام دهند. این ریسمان‌ها منابع فرایند را به اشتراک می‌گذارند، اما می‌توانند به طور مستقل اجرا شوند. مدل برنامه‌نویسی ریسمانی توسعه‌پذیری خوبی را توسط اجرای هم‌زمان ایجاد می‌کند. چندریسمانی همچنین می‌تواند در یک فرایند تنها برای رایانش موازی در سیستم چند پردازشی به کار رود.

برنامه‌های کاربردی چندریسمانی مزیت‌های زیر را دارا هستند [6], [7]:

1. **پاسخ‌دهی**: چندریسمانی کردن یک برنامه تعاملی به برنامه اجازه می‌دهد که حتی اگر بخشی از آن مسدود شده باشد یا عملیاتی طولانی را انجام دهد، به اجرا ادامه دهد و در نتیجه پاسخگویی به کاربر را افزایش دهد. این خاصیت به‌ویژه در طراحی رابط‌های کاربری بسیار مفید است. به‌عنوان‌مثال، در نظر بگیرید که چه اتفاقی می‌افتد وقتی کاربر دکمه‌ای را کلیک می‌کند که منجر به انجام یک عملیات وقت‌گیر می‌شود. یک برنامه تک ریسمانی تا زمانی که عملیات به پایان نرسد، پاسخگوی کاربر نخواهد بود. در مقابل، اگر عملیات زمان بر در یک ریسمان جداگانه و ناهمگام انجام شود، برنامه همچنان پاسخگو به کاربر است.

برای مثال در این برنامه، به‌منظور جلوگیری از تداخلات این‌چنینی رابط کاربری در زمان دریافت داده‌های الکتروانسفالوگرام، دستورات دریافت داده‌ از دستگاه در ریسمانی مجزا از ریسمان اجراکننده برنامه اصلی اجرا می‌شود.

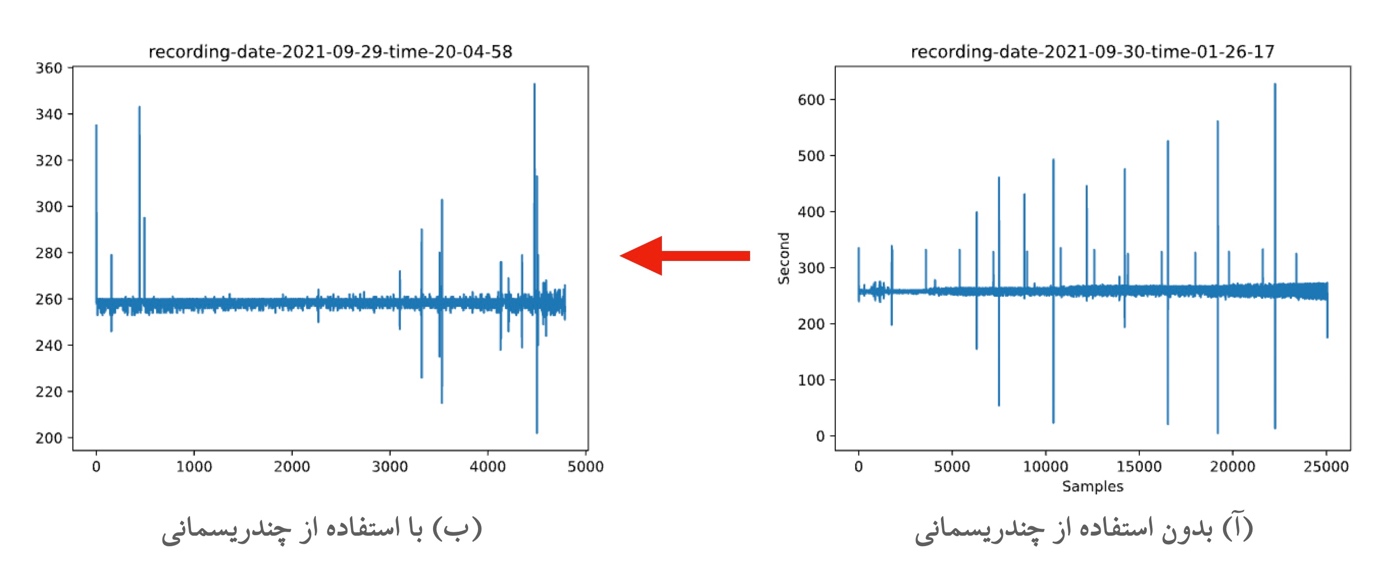
1. **اشتراک منابع**: فرایندها می‌توانند منابع را فقط از طریق تکنیک‌هایی مانند حافظه مشترک و انتقال پیام به اشتراک بگذارند. چنین تکنیک‌هایی باید به طور صریح توسط برنامه‌نویس تدارک دیده شوند. بااین‌حال، ریسمان‌ها به طور پیش‌فرض حافظه و منابع فرایند را به اشتراک می‌گذارند. فایده اشتراک کد و داده این است که به یک برنامه اجازه می‌دهد چندین ریسمان فعالیتی مختلف در فضای آدرس یکسان داشته باشد.
2. **اقتصادی بودن**: اختصاص حافظه و منابع برای ایجاد فرایند هزینه بر است. ازآنجاکه ریسمان‌ها منابع فرایندی را که به آن تعلق دارند به اشتراک می‌گذارند، ایجاد و تغییر ریسمان‌ها اقتصادی‌تر است. ارزیابی تجربی تفاوت در سربار ممکن است دشوار باشد، اما به‌طورکلی ایجاد ریسمان، زمان و حافظه کمتری نسبت به ایجاد فرایند می‌برد. علاوه بر این، تعویض زمینه بین ریسمان‌ها معمولاً سریع‌تر از پردازش‌ها است.
3. **اجرای سریع‌تر**: این مزیت چندریسمانی به برنامه اجازه می‌دهد که در سیستم رایانه‌هایی که دارای چندین پردازنده یا پردازنده‌های چندهسته‌ای هستند؛ سرعت عملکرد را افزایش دهد، زیرا به طور طبیعی ریسمان‌های یک برنامه در هم‌روندی اجرا خود را قرض می‌دهند.
4. **موازی کردن**: برنامه‌های کاربردی به دنبال بهره‌برداری از سیستم‌های چندهسته‌ای و چندپردازنده‌ای که توانایی استفاده از چندریسمانی برای تقسیم داده‌ها و وظایف را دارند، هستند. هرکدام از این وظایف در یک هسته جدا یا به‌صورت موازی در چندین هسته اجرا می‌شود.

برنامه‌های کاربردی رابط گرافیکی دارای یک‌رشته اصلی اجرا هستند که حلقه رویداد و رابط کاربری را اجرا می‌کند. اگر یک کار طولانی‌مدت را در این موضوع راه‌اندازی کنید، رابط کاربری آن تا پایان کار منجمد می‌شود. در این مدت، کاربر نمی‌تواند با برنامه ارتباط برقرار کند و در نتیجه تجربه کاربری ناخوشایندی را در پی خواهد داشت. خوشبختانه، کلاس QThread PyQt به محققین این امکان را داده است تا این موضوع را حل کنند [8].

یکی از چالش‌هایی که حین انتقال داده‌های الکتروانسفالوگرام از هدبند به رایانه از طریق شبکه وجود داشت، یکنواخت نگه داشتن تعداد نمونه‌هایی بود که در ثانیه توسط برنامه دریافت می‌شد. هم‌زمان با ارسال داده‌ها از سوی هدبند به رایانه، عوامل متغیر و تداخل‌زای ذیل باعث می‌شوند که تعداد نمونه‌هایی که در ثانیه دریافت می‌شود ثابت نباشد:

* تداخلات و نویزهای موجود در محیط
* تغییر مداوم باتری هدبند
* سرعت پردازش متفاوت رایانه‌ها
* اجرای هم‌زمان دستورات مختلف شامل:
  + تعداد زیادی الگوریتم هوش مصنوعی برای طبقه‌بندی خواب (Sleep Stage Classification)
  + رسم اسپکتروگرام و پریودوگرام بعد از هر 30 ثانیه
  + رسم ولتاژ اندازه‌گیری شده توسط دو الکترود به‌صورت زمان‌مستقیم
  + فیلترکردن این ولتاژها به‌صورت زمان‌مستقیم و عبور بین بازه 0.5 تا 40 هرتز
  + محاسبه تبدیل فوریه پس از تعداد مشخصی نمونه دریافتی به‌منظور رسم نمودار پریودوگرام ولش

برای حل این مشکل، یک ریسمان جداگانه در نظر گرفته شد. به این معنا که یکی از هسته‌های CPU را که مسئولیت دریافت داده‌ها را دارد فقط به دریافت داده مشغول کرده و از انجام هرگونه پردازش دیگری مربوط به رابط کاربری نرم‌افزار در آنجا جلوگیری شد. به‌طور مشخص سیستم بعد از دریافت 128 نمونه، آن‌ها را ذخیره کرده و داده‌ها را برای رسم و نمایش در اختیار هسته دیگری از CPU یا ریسمان قرار می‌دهد. به‌این‌ترتیب اگر نرخ نمونه‌برداری و ارسال داده 256 نمونه در ثانیه باشد، نمودار مربوطه هر ثانیه 2 بار به‌روزرسانی می‌شود. نمودار شکل 4 این مشکل را قبل و پس از حل شدن نشان می‌دهد. در هر نمودار محور افقی ثانیه و محور عمودی تعداد نمونه‌های دریافت شده است. در قسمت آ قبل از استفاده از چندریسمانی پس از هر ۳۰ ثانیه پرش هایی ناشی از انبوه دادگان دریافتی در ورودی وجود داشت. این مشکل با چندریسمانی در قسمت ب حل شد. در عمل فرآیند داده‌گیری به یک ریسمان متفاوت منتقل گردید.



شکل 4 – تعداد نمونه‌های دریافتی در ثانیه توسط نرم‌افزار (آ) قبل و (ب) بعد از جداسازی رابط کاربری و عملیات داده‌گیری بر روی سی‌پی‌یو

## فرم و ساختار داده ثبت شده

به‌منظور دسترسی دوباره به داده‌های الکتروانسفالوگرام ثبت‌شده حین خواب، این داده‌ها در انتهای فرایند داده‌گیری بر روی فضای ذخیره‌سازی رایانه ذخیره می‌شوند. با زدن کلید Record و شروع داده‌گیری و ثبت، دادگان حس‌گرهای هدبند از قبیل الکتروانسفالوگرام، شتاب‌سنج و دما به رایانه ارسال می‌شوند و در متغیرهای داخلی نرم‌افزار ذخیره می‌شوند. در انتها با زدن کلید پایان ضبط، این عملیات متوقف می‌شود و تمام داده‌های ذخیره شده را در فایلی با فرمت txt در مسیر نرم‌افزار ذخیره می‌کند. همراه این فایل دو فایل دیگر نیز در آن آدرس ذخیره می‌شوند. این فایل‌ها بصورت کلی شامل اطلاعات زمانی و ویژگی‌های فیزیکی محرک‌های ارسالی توسط دستگاه در حین خواب و همچنین پیش‌بینی‌های الگوریتم طبقه‌بندی خواب است.

فایل اول شامل دادگان دریافتی از دستگاه ثبت است. هر خط حاوی دادگان دریافتی در یک نمونه است. دادگان هر خط توسط یک ویرگول از هم جدا می‌شوند. در هر خط به ترتیب ولتاژ ثبت شده توسط الکترود چپ، الکترود راست، دمای بدن، شماره نمونه و ثانیه نمونه دریافتی ذخیره شده است. فایل دوم حاوی اطلاعات زمانی و ویژگی‌های فیزیکی محرک‌های ارسالی توسط دستگاه در حین خواب است. این محرک‌ها شامل نور، صوت، و یا لرزش می‌باشند. فایل سوم شامل تمام پیش‌بینی‌های الگوریتم طبقه‌بندی خواب است. درصورتی‌که الگوریتمی فعال نباشد، محتویات این فایل خالی خواهد بود.

## جمع بندی

در این فصل ابتدا دلایل انتخاب و فرآیند استفاده از پای‌کیوت به عنوان نرم‌افزار استفاده‌شده برای طراحی رابط کاربری توضیح داده شد. سپس عملکرد توابع شرح داده شد. این توابع شامل توابع تنظیم‌کننده (مانند تابع ارتباط نرم‌افزار با سرور هدبند)، توابع گرافیکی (مانند تابع رابط کاربری، کلیدها و نمودارها)، توابع کنترلی (مانند تابع تنظیم و ارسال دستورات اجرای محرک به هدبند توسط سرور) و توابع و کلاس‌های مربوط به چندریسمانی می‌شوند. سپس نحوه رسم اسپکتروگرام توسط طیف چندمخروطی توضیح داده شد. علاوه بر این نشان داده شد که چگونه استفاده از چندریسمانی مشکلات موجود در دریافت دادگان از سرور را حل می‌کند. در نهایت خروجی‌های نرم افزار معرفی و توضیح داده شدند. در فصل بعد پس از معرفی دستگاه ثبت دادگان الکتروانسفالوگرام و معرفی دادگان تجربی ثبت شده در پژوهش به بررسی ساختار پایگاه دادة موجود پرداخته می‌شود. سپس به ترتیب فرایند طبقه‌بندی بصری این دادگان، نحوه دسته‌بندی و نمایش این دادگان، روش پیش‌پردازش شامل حذف خط پایه، اعمال یک فیلتر میان‌گذر بر دادگان، دوره‌بندی زمانی، و در نهایت مراحل پردازش و استخراج ویژگی موردبحث قرار می‌گیرد. پس از آن نیز به بررسی روش‌های مختلف انتخاب ویژگی و روش طبقه‌بندی استفاده شده یعنی SVM پرداخته می‌شود.

# مراجع

[1] “Python and PyQt: Building a GUI Desktop Calculator – Real Python.” https://realpython.com/python-pyqt-gui-calculator/ (accessed Jun. 02, 2022).

[2] M. J. Prerau, R. E. Brown, M. T. Bianchi, J. M. Ellenbogen, and P. L. Purdon, “Sleep Neurophysiological Dynamics Through the Lens of Multitaper Spectral Analysis,” *Physiology*, vol. 32, no. 1, pp. 60–92, Jan. 2017, doi: 10.1152/physiol.00062.2015.

[3] D. J. Thomson, “Spectrum estimation and harmonic analysis,” *Proceedings of the IEEE*, vol. 70, no. 9, pp. 1055–1096, Sep. 1982, doi: 10.1109/PROC.1982.12433.

[4] T. P. Bronez, “On the performance advantage of multitaper spectral analysis,” *IEEE Transactions on Signal Processing*, vol. 40, no. 12, pp. 2941–2946, Dec. 1992, doi: 10.1109/78.175738.

[5] A. K. Patel, V. Reddy, and J. F. Araujo, “Physiology, Sleep Stages,” in *StatPearls*, Treasure Island (FL): StatPearls Publishing, 2022. Accessed: Jun. 02, 2022. [Online]. Available: http://www.ncbi.nlm.nih.gov/books/NBK526132/

[6] “The Benefits of Multithreaded Programming.” https://www.tutorialspoint.com/the-benefits-of-multithreaded-programming (accessed Jun. 02, 2022).

[7] “Benefits of Multithreading in Operating System - GeeksforGeeks.” https://www.geeksforgeeks.org/benefits-of-multithreading-in-operating-system/ (accessed Jun. 02, 2022).

[8] “Use PyQt’s QThread to Prevent Freezing GUIs – Real Python.” https://realpython.com/python-pyqt-qthread/ (accessed Jun. 02, 2022).

1. Graphical User Interface [↑](#footnote-ref-1)
2. PyQt [↑](#footnote-ref-2)
3. Qt [↑](#footnote-ref-3)
4. Thread [↑](#footnote-ref-4)
5. SQL [↑](#footnote-ref-5)
6. SVG [↑](#footnote-ref-6)
7. OpenGL [↑](#footnote-ref-7)
8. XML [↑](#footnote-ref-8)
9. RiverBank Computing Ltd [↑](#footnote-ref-9)
10. Qt Designer [↑](#footnote-ref-10)
11. Multitaper Spectral Analysis [↑](#footnote-ref-11)
12. Spectrogram [↑](#footnote-ref-12)
13. Discrete prolate spheroidal sequence [↑](#footnote-ref-13)
14. Time-half-bandwidth product [↑](#footnote-ref-14)
15. Panel [↑](#footnote-ref-15)