

دانشکده مهندسی کامپیوتر

سیگنال‌ها و سیستم‌ها

پروژه امتیازی

نیم‌سال دوم
سال تحصیلی ۱۴۰۳-۱۴۰۴

در انجام تمرین‌ها به نکات زیر توجه نمایید:

- این پروژه شامل دو ریزپروژه می‌باشد که به انتخاب خودتان می‌توانید یکی از آن‌ها را انتخاب و انجام دهید.
- فایل‌های پروژه به همراه یک گزارش به فرمت `Signal_PRJ_student-number.pdf` در قالب یک فایل فشرده با فرمت `Signal_PRJ_student-number.zip` ارسال کنید.
- در گزارش باید منطق پیاده‌سازی بخش‌های مختلف نوشته شود (نیازی به توضیح بخش گرافیکی نیست).
- استفاده از کتابخانه‌های آماده تبدیل فوریه مانند `numpy.fft` مجاز نیست و در صورت استفاده از آن‌ها نمره‌ای به آن تعلق نمی‌گیرد.
- کامنت گذاری در کد جهت افزایش خوانایی انجام شود.
- این پروژه تحویل حضوری یا آنلاین خواهد داشت.
- در صورت هر گونه مشکل می‌توانید به اکانت تلگرام `@parsa22000` (برای زیرپروژه اول) یا `@S_H_1903` (برای زیرپروژه دوم) پیام دهید.
- مهلت انجام این تمرین تا ساعت ۹:۰۰ صبح شنبه ۱۷ خرداد است .

فهرست مطالب

۴	۱ زیرپروژه اول: فیلترها و تاثیر نمونه برداری بر آنها
۴	۱-۱ فاز اول: نمایش سیگنال ها
۵	۲-۱ فاز دوم: نمایش تبدیل فوریه
۶	۳-۱ فاز سوم: فیلتر
۷	۴-۱ فاز چهارم: نمونه برداری
۷	۱-۴-۱ تولید $x_p(t)$
۸	۲-۴-۱ محاسبه $x_p[n]$
۹	۳-۴-۱ تولید $X_p(j\omega)$
۱۰	۲ زیرپروژه دوم: طیف نگار
۱۰	۱-۲ فاز اول: تولید سیگنال های دارای محتوای فرکانسی متغیر
۱۱	۲-۲ فاز دوم: پیاده سازی طیف نگار
۱۲	۳-۲ فاز سوم: تحلیل فایل صوتی واقعی

مقدمه

در درس سیگنال‌ها و سیستم‌ها با مفاهیم اولیه سری فوریه، تبدیل فوریه، فیلترها و تاثیر نمونه‌برداری^۱ بر آن‌ها آشنا شدیم. در این پروژه می‌خواهیم برخی از این مفاهیم را پیاده‌سازی کرده و رفتار آن‌ها را مشاهده کنیم.^۲

این پروژه شامل دو زیرپروژه می‌شود. هر کدام از دانشجویان محترم می‌توانند یکی از این دو زیرپروژه را انتخاب کرده و آن را پیاده‌سازی کنند. در آخر پروژه‌ها به صورت حضوری یا مجازی تحویل گرفته می‌شود. در هر کدام از زیرپروژه‌ها هدف ایجاد یک نرم‌افزار است که بتوان بعضی سیگنال‌ها را نمایش داد، روی آن‌ها عملیاتی انجام داد و نتیجه‌ی اعمال عملیات را در خروجی مشاهده کرد.

زیرپروژه اول: فیلترها و تاثیر نمونه‌برداری بر آن‌ها

زیرپروژه دوم: طیف‌نگار^۳

^۱Sampling

^۲توجه شود که استفاده از توابع کتابخانه‌های آماده برای محاسبه تبدیل فوریه مانند `numpy.fft` مجاز نیست.

^۳Spectrogram

۱ زیر پروژه اول: فیلترها و تاثیر نمونه برداری بر آنها

۱-۱ فاز اول: نمایش سیگنالها

در ابتدای پروژه می‌خواهیم بتوانیم سیگنال‌هایی پیش فرض تعریف کنیم و آنها را بر روی نمودار نمایش دهیم. نام این نمودار را **Time-Representation** بگذارید. سیگنال‌های پیشنهادی ما به صورت زیر است:

$$\sin\left(\frac{2\pi}{T}t\right), \quad \text{Periodic with period } T \bullet$$

$$\cos\left(\frac{2\pi}{T}t\right), \quad \text{Periodic with period } T \bullet$$

$$\text{SQUARE}(t) = \begin{cases} -1, & -T < t < 0 \\ 1, & 0 < t < T \\ 0, & o.w. \end{cases}, \quad \text{Non-Periodic} \bullet$$

$$\text{TRIANGLE}(t) = \begin{cases} 2t + 1, & -T < t < 0 \\ -2t + 1, & 0 < t < T \\ 0, & o.w. \end{cases}, \quad \text{Non-Periodic} \bullet$$

$$\text{SAWTOOTH}(t) = \begin{cases} t, & -T < t < T \\ 0, & o.w. \end{cases}, \quad \text{Non-Periodic} \bullet$$

دو سیگنال اول متناوب با دوره تناوب T و سایر سیگنال‌ها در بازه $-T < t < T$ مقدار غیر صفر دارند و در سایر نقاط صفر هستند. مقدار ثابت T باید قابل تغییر باشد. در رابط کاربری طراحی شده باید بتوان بر روی دکمه‌های مربوط به هر کدام از این سیگنال‌ها کلیک کرد و آن سیگنال را بر روی نمودار مشاهده کرد. پیشنهاد ما برای طراحی رابط کاربری استفاده از کتابخانه `tkinter` و برای نمایش نمودارها استفاده از کتابخانه `matplotlib` پایتون است. می‌توانید از تمرین‌های عملی خود نیز کمک بگیرید.

اگر از کتابخانه `numpy` استفاده می‌کنید، برای نمایش سیگنال باید آن را به صورت زیر نمونه برداری

کنید:

```
# x(t): a defined function
t = np.linspace(start, stop, num)
x_t = x(t)
```

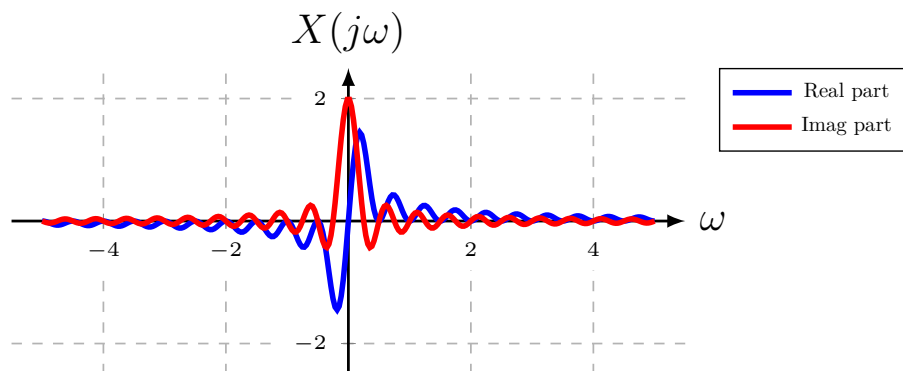
در کد بالا num تعداد نمونه‌ها را مشخص می‌کند و $\frac{\text{stop-start}}{\text{num}}$ فاصله‌های نمونه‌برداری است که آن را با Δ نمایش می‌دهیم. معکوس آن که فرکانس نمونه‌برداری است را با $\frac{1}{\Delta}$ نمایش می‌دهیم. سیگنال نمایش داده شده را با $x[n]$ نمایش می‌دهیم که به صورت زیر تعریف می‌شود:

$$x[n] = x(t = n\Delta)$$

که n همان اندیس مقادیر سیگنال را نشان می‌دهد (توجه کنید که n می‌تواند منفی نیز باشد). فرکانس نمونه‌برداری برای نمایش سیگنال ($\frac{1}{\Delta}$) را با فرکانس نمونه‌برداری در فاز چهارم (ω_s) اشتباه نگیرید.

۲-۱ فاز دوم: نمایش تبدیل فوریه

پس از نمایش هر سیگنال، تبدیل فوریه آن باید قابل مشاهده بر روی نمودار دیگری با نام **Fourier-Representation** باشد. با انتخاب هر سیگنال پیش‌فرض باید سیگنال فوریه آن نمایش داده شود. از آنجایی که تبدیل فوریه سیگنالی مختلط به ما می‌دهد، در نمودار تبدیل فوریه بخش real سیگنال را با یک رنگ و بخش imaginary را با یک رنگ دیگر نمایش دهید. شکل زیر یک نمونه از این نمودار است:



در کنار سیگنال‌های زمانی که در فاز قبل به آن‌ها اشاره شد، همان سیگنال‌ها را باید بتوان در حوزه فرکانس نیز نمایش داد و تبدیل فوریه معکوس آن‌ها را در نمودار زمانی مشاهده کرد. یعنی سیگنال‌های زیر هم باید قابل مشاهده باشند (مانند بخش قبل باید بتوان W را نیز تغییر داد):

$$\sin\left(\frac{2\pi}{W}\omega\right), \quad \text{Periodic with period } W \bullet$$

$$\cos\left(\frac{2\pi}{W}\omega\right), \quad \text{Periodic with period } W \bullet$$

$$\text{SQUARE}(t) = \begin{cases} -1, & -W < \omega < 0 \\ 1, & 0 < \omega < W \\ 0, & o.w. \end{cases}, \quad \text{Non-Periodic} \bullet$$

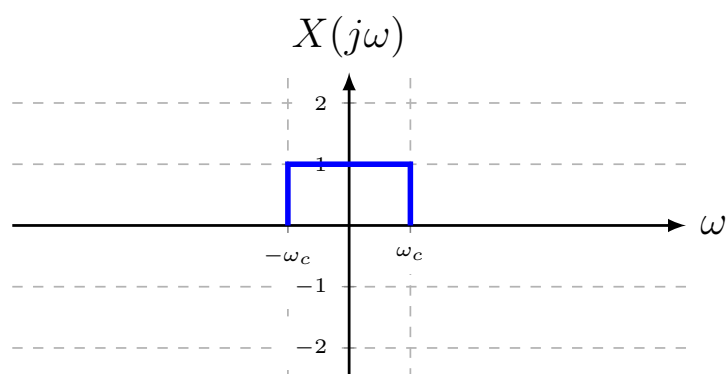
$$\text{TRIANGLE}(\omega) = \begin{cases} 2\omega + 1, & -W < \omega < 0 \\ -2\omega + 1, & 0 < \omega < W \\ 0, & o.w. \end{cases}, \quad \text{Non-Periodic} \bullet$$

$$\text{SAWTOOTH}(\omega) = \begin{cases} \omega, & -W < \omega < W \\ 0, & o.w. \end{cases}, \quad \text{Non-Periodic} \bullet$$

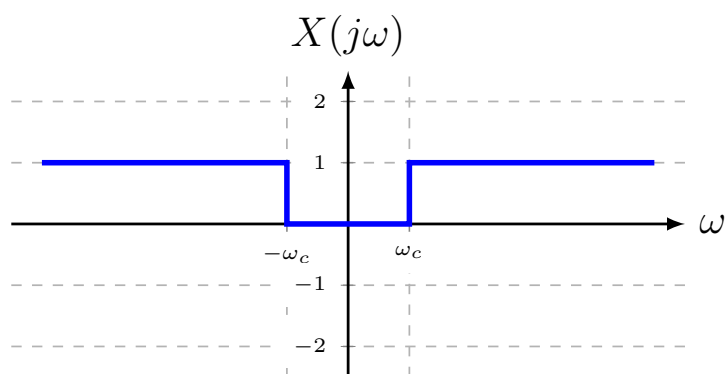
۳-۱ فاز سوم: فیلتر

پس از نمایش یک سیگنال (در حوزه زمان و فرکانس به طور همزمان) باید بتوان بر روی آن انواع فیلتر را اعمال کرد. پس از انتخاب ω_c (یا ω_{c1}, ω_{c2} در فیلتر میان گذر) و کلیک بر روی یکی از فیلترها، نمودار زمان و فرکانس باید متناسب با نوع فیلتر تغییر کنند. فیلترها به صورت زیر می باشند:

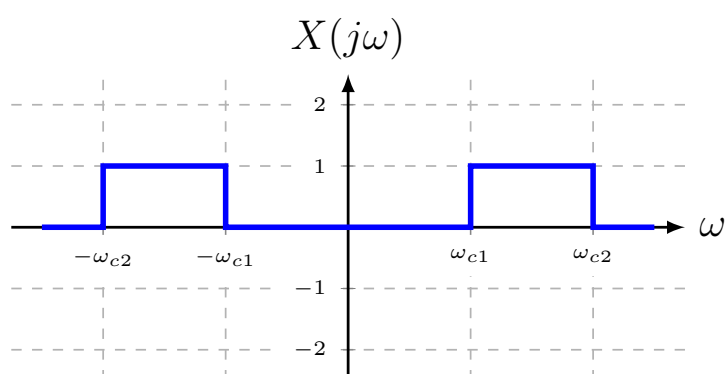
• فیلتر پایین گذر



• فیلتر بالا گذر



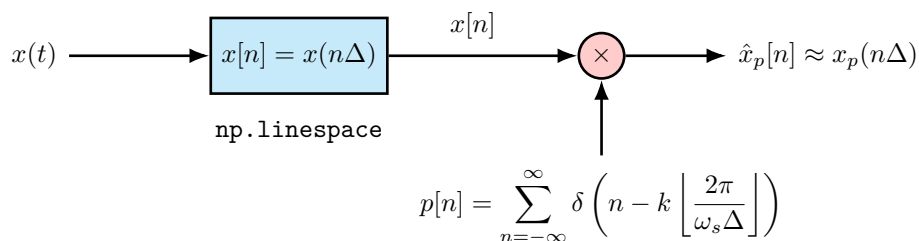
• فیلتر میان گذر



۴-۱ فاز چهارم: نمونه برداری

۱-۴-۱ تولید $x_p(t)$

برای مشاهده تاثیر نمونه برداری بر فرکانس سیگنال، پس از انتخاب sampling rate که آن را با ω_s نمایش می دهیم و کلیک بر روی دکمه sample، نمودار حوزه زمان باید با آن فرکانس نمونه برداری شود. شکل زیر فرایند تولید نمودار نمونه برداری شده از $x(t)$ را نشان می دهد:



شکل ۱: فرایند نمایش سیگنال بر روی نمودار تا نمونه برداری از آن

در ادامه نحوه محاسبه $\hat{x}_p[n]$ که تقریبی از $x_p(n\Delta)$ است را به طور کامل تشریح می کنیم.

از آنجایی که تابع $\delta(t)$ به صورت حدی تعریف می‌شود، می‌توانید از تقریب زیر به جای آن استفاده کنید:

$$\delta(t) \approx \begin{cases} \frac{1}{\Delta}, & 0 \leq t < \Delta \\ 0, & o.w. \end{cases}$$

که در رابطه بالا Δ همان فاصله زمانی است که با آن نمودارها را در حوزه زمان نشان می‌دهید. مثلاً اگر تابع $\sin(t)$ را با $\text{sample_rate}=1000$ در numpy نمایش می‌دهید، مقدار $\Delta = \frac{1}{1000}$ خواهد بود (sample_rate استفاده شده برای نمایش نمودار را با ω_s استفاده شده برای نمونه برداری آن سیگنال اشتباه نگیرید).

سیگنال نمونه‌برداری شده ($x_p(t)$) با تقریب بالا به صورت زیر حاصل می‌شود:

$$\begin{aligned} x_p(t) &= \sum_{k=-\infty}^{\infty} x(t) \delta\left(t - k \frac{2\pi}{\omega_s}\right) \\ &\approx \begin{cases} \frac{1}{\Delta} x(t), & k \frac{2\pi}{\omega_s} \leq t < k \frac{2\pi}{\omega_s} + \Delta \quad (\exists k \in \mathbb{Z}) \\ 0, & o.w. \end{cases} \end{aligned}$$

اگر سیگنال نمایش داده بر روی نمودار را با $x[n]$ نمایش دهیم، $x(t)$ و $x[n]$ رابطه‌ی زیر را دارند:

$$x[n] = x(t = n\Delta)$$

لذا $\hat{x}_p[n]$ به صورت زیر محاسبه می‌شود:

$$\begin{aligned} x_p[n] = x_p(n\Delta) &\approx \begin{cases} \frac{1}{\Delta} x(n\Delta), & k \frac{2\pi}{\omega_s} \leq n\Delta < k \frac{2\pi}{\omega_s} + \Delta \quad (\exists k \in \mathbb{Z}) \\ 0, & o.w. \end{cases} \\ &= \begin{cases} \frac{1}{\Delta} x[n], & k \frac{2\pi}{\omega_s \Delta} \leq n < k \frac{2\pi}{\omega_s \Delta} + 1 \quad (\exists k \in \mathbb{Z}) \\ 0, & o.w. \end{cases} \\ &= \hat{x}_p[n] \end{aligned}$$

اگر از نامعادله $k \frac{2\pi}{\omega_s \Delta} \leq n < k \frac{2\pi}{\omega_s \Delta} + 1$ کف بگیریم نامعادله زیر حاصل می‌شود:

$$\left\lfloor k \frac{2\pi}{\omega_s \Delta} \right\rfloor \leq n < \left\lfloor k \frac{2\pi}{\omega_s \Delta} \right\rfloor + 1$$

که این نامعادله برا معادله زیر برابر است:

$$n = \left\lfloor k \frac{2\pi}{\omega_s \Delta} \right\rfloor$$

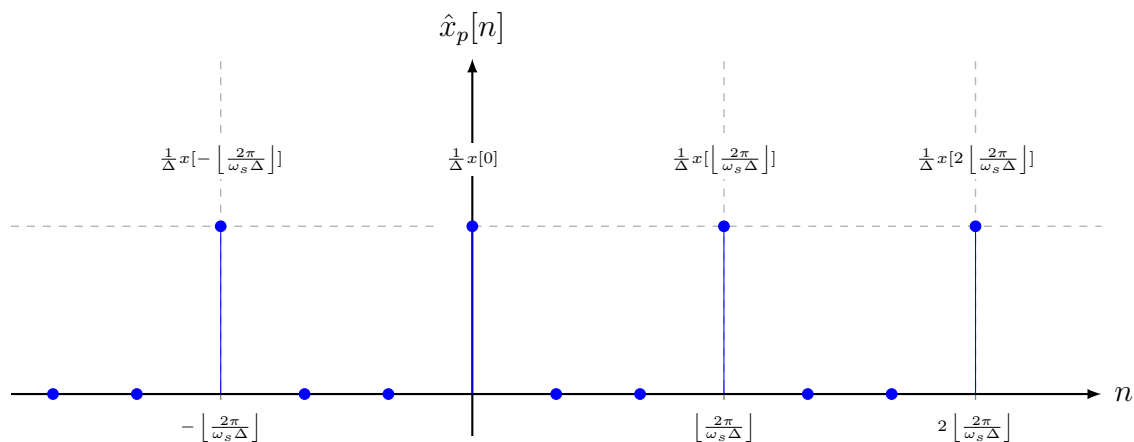
و $\hat{x}_p[n]$ به صورت زیر در می‌آید: از آنجایی که $\text{sample_rate} = \frac{1}{\Delta}$ است $x(t)$ نمایش داده شده مقدار ثابتی در فواصل Δ دارد) می‌توان فرمول بالا را به صورت زیر نوشت:

$$x_p[n] \approx \hat{x}_p[n] \begin{cases} \frac{1}{\Delta} x[n], & n = \left\lfloor k \frac{2\pi}{\omega_s \Delta} \right\rfloor \quad (\exists k \in \mathbb{Z}) \\ 0, & o.w. \end{cases}$$

که طبق تقریبی که برای $\delta(t)$ زدیم، $\hat{x}_p[n]$ که در شکل ۱ بدست آمده بود حاصل می‌شود:

$$\begin{aligned}\hat{x}_p[n] &= x[n] \sum_{k=-\infty}^{\infty} \delta \left(n - \left\lfloor k \frac{2\pi}{\omega_s \Delta} \right\rfloor \right) \\ &\approx x[n] \sum_{k=-\infty}^{\infty} \delta \left(n - k \left\lfloor \frac{2\pi}{\omega_s \Delta} \right\rfloor \right)\end{aligned}$$

این یعنی برای تولید سیگنال نمونه‌برداری شده، کفایست مقادیر $x[n]$ را در مضارب $\left\lfloor \frac{2\pi}{\omega_s \Delta} \right\rfloor$ در $\frac{1}{\Delta}$ ضرب کرده و در باقی نقاط صفر کنید.



۳-۴-۱ تولید $X_p(j\omega)$

پس از تولید $x_p[n]$ که همان $x_p(t)$ تقریبی است، باید تبدیل فوریه آن در نمودار فوریه نمایش داده شود. برای اینکار کافیهست $x_p[n]$ تولید شده را به همان تابع تبدیل فوریه که از قبل نوشته‌اید بدهید تا تبدیل فوریه آن را نمایش دهد. بعد از نمونه‌برداری سیگنال‌های مختلف چه چیزی مشاهده می‌کنید؟

۲ زیر پروژه دوم: طیف‌نگار

تبدیل فوریه یکی از ابزارهای بنیادی در تحلیل سیگنال‌ها به شمار می‌رود و اطلاعات ارزشمندی درباره‌ی محتوای فرکانسی سیگنال‌ها ارائه می‌دهد. با این حال، این تبدیل تنها فرکانس‌های موجود در سیگنال را بدون در نظر گرفتن تغییرات آن‌ها در طول زمان مشخص می‌نماید. این در حالی است که در بسیاری از کاربردهای عملی، از جمله پردازش گفتار، موسیقی و سیگنال‌های زیستی، دانستن این که چه فرکانس‌هایی در چه زمان‌هایی حضور دارند، بسیار حائز اهمیت است.

طیف‌نگار ابزاری گرافیکی است که برای تحلیل زمانی-فرکانسی سیگنال‌ها مورد استفاده قرار می‌گیرد. در نمایش طیف‌نگار:

- محور افقی نمایانگر زمان،

- محور عمودی نمایانگر فرکانس،

- و شدت رنگ، نمایانگر قدرت (دامنه) هر مؤلفه فرکانسی در زمان مشخص است.

برای رسم طیف‌نگار، سیگنال به قطعه‌های زمانی کوچکی تقسیم می‌شود و از هر قطعه تبدیل فوریه گرفته می‌شود. این روش تحلیل را اغلب به عنوان **تبدیل فوریه پنجره‌ای (STFT)** می‌شناسند.

۱-۲ فاز اول: تولید سیگنال‌های دارای محتوای فرکانسی متغیر

در این بخش یک یا چند سیگنال با محتوای فرکانسی غیرثابت تولید نمایم. نمونه‌هایی از این نوع سیگنال‌ها به شرح زیر است:

۱. سیگنال چیرپ خطی (Linear Chirp)

فرکانس سیگنال به صورت خطی از مقدار اولیه به مقدار نهایی تغییر می‌کند.

$$x(t) = \sin\left(2\pi\left(f_0 t + \frac{k}{2}t^2\right)\right)$$

که در آن f_0 فرکانس ابتدایی و $k = \frac{f_1 - f_0}{T}$ نرخ تغییر فرکانس است.

مثال: فرکانس از 100 هرتز به 1000 هرتز در بازه زمانی 5 ثانیه تغییر کند.

۲. سیگنال چیرپ نمایی (Exponential Chirp)

در این نوع، فرکانس به صورت نمایی تغییر می‌کند.

$$x(t) = \sin\left(2\pi f_0 \frac{r^t - 1}{\ln(r)}\right)$$

$$r = \left(\frac{f_1}{f_0}\right)^{1/T}$$

مثال: فرکانس از 200 هرتز به 2000 هرتز به صورت نمایی در ۴ ثانیه افزایش یابد.

۳. سیگنال سینوسی قطعه‌ای با فرکانس‌های متفاوت

سیگنال در بازه‌های زمانی مجزا دارای فرکانس‌های مختلف است.

$$x(t) = \begin{cases} \sin(2\pi \cdot 300t), & 0 \leq t < 2 \\ \sin(2\pi \cdot 500t), & 2 \leq t < 4 \\ \sin(2\pi \cdot 400t), & 4 \leq t < 6 \end{cases}$$

۴. ترکیب سینوسی با فرکانس‌های مختلف در زمان‌های مختلف

در هر بازه زمانی ترکیبی از چند فرکانس داریم.

$$x(t) = \begin{cases} \sin(2\pi \cdot 300t) + \sin(2\pi \cdot 700t), & 0 \leq t < 2 \\ \sin(2\pi \cdot 500t) + \sin(2\pi \cdot 900t), & 2 \leq t < 4 \\ \sin(2\pi \cdot 400t) + \sin(2\pi \cdot 1200t), & 4 \leq t < 6 \end{cases}$$

سیگنال تولیدشده باید بر حسب زمان رسم گردد تا رفتار آن در حوزه زمان به‌وضوح قابل مشاهده باشد. استفاده از کتابخانه matplotlib برای رسم نمودارها توصیه می‌شود.

۲-۲ فاز دوم: پیاده‌سازی طیف‌نگار

در این بخش، باید با استفاده از یکی از روش‌های موجود طیف‌نگار سیگنال را محاسبه و نمایش دهید. در طراحی برنامه، باید امکان کنترل پارامترهای زیر برای کاربر فراهم باشد:

- اندازه پنجره (Window Length)
- میزان هم‌پوشانی بین پنجره‌ها (Overlap)
- نوع پنجره (مانند Hann, Hamming یا Rectangular)

طیف‌نگار باید به گونه‌ای نمایش داده شود که تغییرات فرکانسی سیگنال در طول زمان به صورت دقیق و قابل تفسیر باشد.

۲-۳ فاز سوم: تحلیل فایل صوتی واقعی

در این مرحله، یک فایل صوتی واقعی (با فرمت WAV) را بارگذاری نموده، آن را تحلیل کرده و طیف‌نگار آن را نمایش دهید.
وظایف مورد انتظار:

- خواندن فایل صوتی با استفاده از تابع `wavfile.read` از کتابخانه `scipy.io`
- نمایش شکل موج فایل صوتی در حوزه زمان
- رسم طیف‌نگار سیگنال صوتی
- افزودن قابلیت پخش فایل صوتی
- شناسایی بازه‌های زمانی با انرژی بالا
- مقایسه طیف‌نگار دو سیگنال صوتی مختلف