سروین نامی 9931103

سارا روحانی 9827030

گزارش پروژه‌ی سیگنال

در ابتدا، مقدار ثابتی به نام SAMPLING\_RATE تعریف می‌شود که نرخ نمونه‌برداری را نشان می‌دهد. سپس توابعی برای خواندن سیگنال ورودی از فایل، اعمال فیلتر پهن باند به سیگنال و تغییر فرکانس سیگنال تعریف می‌شوند.

1. تابع read\_input\_file(file\_path):

این تابع مسئول خواندن سیگنال ورودی از یک فایل مشخص است. آرگومان file\_path مسیر فایل ورودی را دریافت کرده و سیگنال را خوانده و به صورت لیستی از اعداد بازمی‌گرداند.

1. تابع apply\_bandpass\_filter(signal\_data, cutoff\_freq):

این تابع فیلتر پهن باند را بر روی سیگنال ورودی اعمال می‌کند. با دریافت دو آرگومان signal\_data (سیگنال ورودی) و cutoff\_freq (فرکانس قطع فیلتر)، ابتدا فرکانس قطع را به صورت مقدار نرمال شده محاسبه می‌کند. سپس از فیلتر Butterworth با استفاده از مقادیر b و a استفاده می‌کند تا فیلتر لوپاس را بسازد. در نهایت، سیگنال ورودی را با استفاده از تابع lfilter از کتابخانه scipy.signal فیلتر می‌کند و سیگنال فیلتر شده را به عنوان خروجی بازمی‌گرداند.

1. تابع change\_frequency(signal\_data, freq\_shift):

این کد فرکانس سیگنال ورودی را تغییر می‌دهد. این تابع دو آرگومان را می‌پذیرد: `signal\_data` که یک آرایه‌ی NumPy حاوی سیگنال ورودی است و `freq\_shift` که میزان تغییر فرکانس را نشان می‌دهد.

تابع ابتدا یک آرایه‌ی زمان ایجاد می‌کند (`time`) که به ازای هر نمونه در سیگنال ورودی، مقدار زمان متناظر را در آرایه ذخیره می‌کند. این آرایه زمان بر اساس نرخ نمونه‌برداری (`SAMPLING\_RATE`) محاسبه می‌شود.

سپس متغیرهای `start\_freq` و `end\_freq` تعیین می‌شوند که نشان دهنده فرکانس شروع و پایان برای تغییر فرکانس هستند. `end\_freq` برابر با ضرب `freq\_shift` در ۱۰۰۰ است.

سیگنال جدید (`shifted\_signal`) با ضرب سیگنال ورودی در یک موج برخیپ (`chirp`) محاسبه می‌شود. تابع `signal.chirp` یک موج برخیپ تولید می‌کند که فرکانس آن از `start\_freq` به `end\_freq` با توجه به آرایه‌ی زمان تغییر می‌کند. سپس سیگنال ورودی با این موج برخیپ ضرب می‌شود.

در نهایت، سیگنال جدید (`shifted\_signal`) با سیگنال ورودی (`signal\_data`) ضرب می‌شود و به عنوان خروجی تابع برگردانده می‌شود.

1. تابع play\_audio(audio\_data):

این تابع مسئول پخش سیگنال صوتی است. با دریافت آرگومان audio\_data (داده صوتی)، از تابع play از کتابخانه sounddevice استفاده کرده و داده صوتی را با نرخ نمونه برداری مشخص شده پخش می‌کند.

با استفاده از این توابع، برنامه قادر است سیگنال ورودی را از فایل خوانده و به صورت پشت سر هم فیلتر پهن باند و تغییر فرکانس دهد، سپس سیگنال تغییر یافته را پخش کند.