سروین نامی

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 تابع فرکانس سیگنال را تغییر می‌دهد. با دریافت دو آرگومان signal\_data (سیگنال ورودی) و freq\_shift (مقدار تغییر فرکانس)، ابتدا زمان را بر اساس نرخ نمونه برداری محاسبه می‌کند. سپس از یک موج پیچیده مختلط با استفاده از تابع exp از کتابخانه numpy استفاده می‌کند تا فرکانس سیگنال را تغییر دهد. در نهایت، سیگنال تغییر یافته را به صورت قسمت حقیقی از موج پیچیده مختلط بازمی‌گرداند.

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.