

به نام خدا

دانشگاه خواجه نصیرالدین طوسی

دانشکده کامپیوتر

نام درس : سیگنال ها و سیستم ها

استاد درس : دکتر مرادیان

عنوان پروژه : طراحی سیستم مخابراتی چند کاناله (FDM)

و تحلیل اثرات دیجیتال سازی

سید محمد سعید مظهری

ترم ۴۰۴۱

## مقدمه

هدف از انجام این پروژه، پیاده‌سازی یک زنجیره مخابراتی دیجیتال کامل شامل ضبط سیگنال‌های صوتی، کوانتیزاسیون، مدولاسیون تقسیم فرکانسی (FDM)، بررسی قضیه نایکوئیست، طراحی فیلتر میان گذر (BPF) و در نهایت دمدولاسیون و بازیابی سیگنال اصلی است.

این پروژه با زبان برنامه نویسی پایتون (python) انجام شده است. در این پروژه سه سیگنال صوتی متفاوت ضبط شده و هر کدام از آنها بر روی یک فرکانس حامل متفاوت مدوله شده‌اند تا امکان ارسال هم زمان آن‌ها روی یک کانال مشترک فراهم شود. سپس در سمت گیرنده با استفاده از فیلترهای دیجیتال، یک کانال مشخص، استخراج و صدای اصلی بازیابی شده است.

## ضبط سیگنال‌های صوت ورودی

در این قسمت از پروژه با استفاده از کتابخانه‌ی `sounddevice` و تابع `rec`، از این کتابخانه، سه سیگنال صوتی با نرخ نمونه برداری `44100 Hz` ضبط شده است. که هر سیگنال به صورت تک‌کاناله است و طول آن به اندازه ۱۰ ثانیه می‌باشد. برای فاصله انداختن بین ضبط صداها نیز کتابخانه `time` و تابع `sleep` مورد استفاده قرار گرفت. سپس بعد از ضبط، سیگنال‌های صوت با استفاده از ماژول `wavfile` در کتابخانه‌ی `scipy.io`، به فرمت `wav` ذخیره شدند.

سه فایل صوتی ضبط شده :

audio\_1.wav (اعداد ۱ تا ۱۰)

audio\_2.wav (اعداد ۱۱ تا ۲۰)

audio\_3.wav (اعداد ۲۱ تا ۳۰)

## گام اول : کوانتیزاسیون (Quantization)

توضیح مفهومی : برای ذخیره سیگنال های آنالوگ در کامپیوتر، دامنه ها که به صورت پیوسته هستند باید به سطوح گسسته (بیت) تبدیل شوند. همچنین کاهش تعداد بیت ها باعث ایجاد نویز کوانتیزاسیون میشود. در این قسمت از پروژه سیگنال ذخیره شده اول (audio\_1.wav) به کمک فرمول زیر با دو دقت ۸ و ۳ بیت کوانتیزه میشود.

$$xq = np.round(x \cdot 2^{bits-1} - 1) / (2^{bits-1} - 1)$$

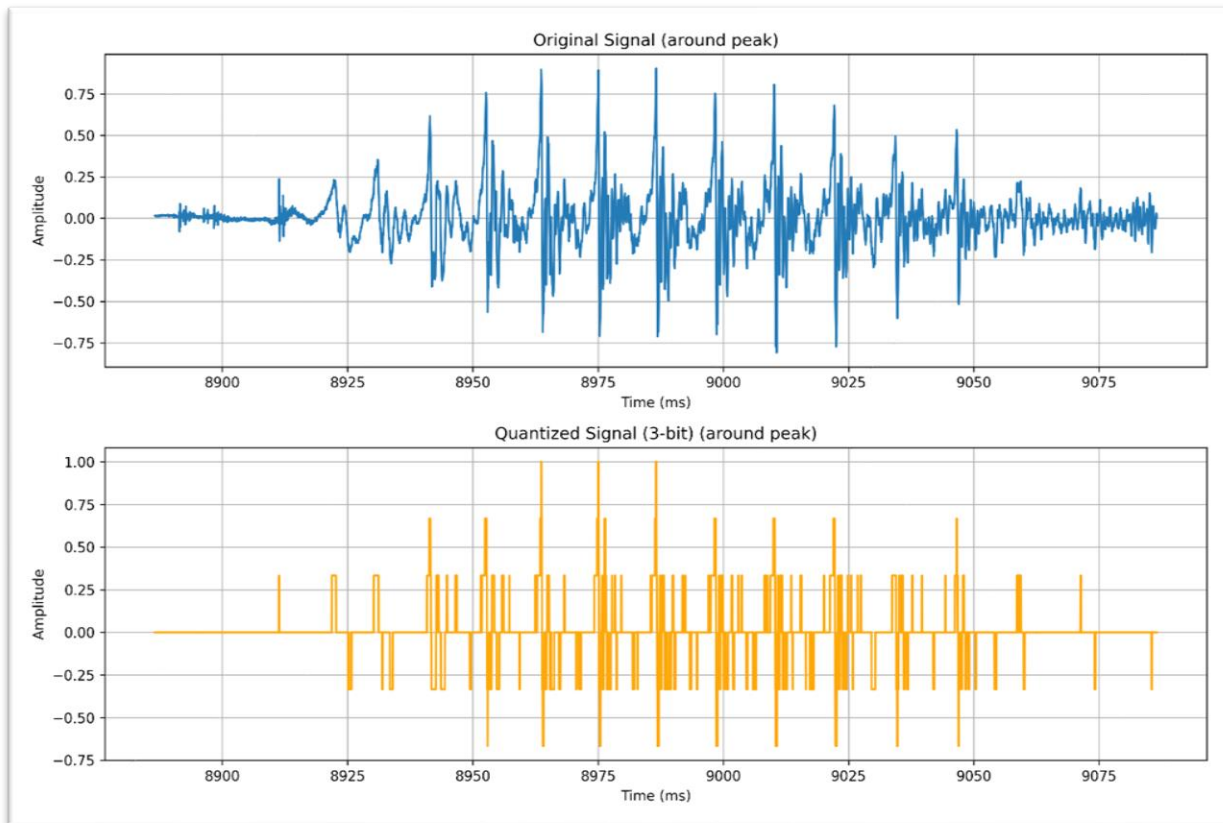
که در آن از تابع round در کتابخانه ی numpy (به اختصار np) برای گرد کردن اعداد به تعداد اعشار مشخص استفاده شده است. و داریم :

X : سیگنال صوت ضبط شده

xq : سیگنال کوانتیزه شده

bits : دقت کوانتیزاسیون

سپس برای نمایش تأثیر کوانتیزاسیون، یک بازه ی زمانی ۲۰۰ میلی ثانیه ای شامل بیشترین دامنه ی سیگنال استخراج شد. بدین منظور، بیشینه ی قدرمطلق سیگنال محاسبه و اندیس متناظر با آن به ترتیب با استفاده از توابع np.abs و np.argmax تعیین گردید. فایل های کوانتیزه شده با نام های audio\_1\_3bit.wav و audio\_1\_8bit.wav ذخیره شدند. در ادامه، نمودار مقایسه ای سیگنال اصلی و سیگنال کوانتیزه شده با ۳ بیت در این بازه ترسیم شد. برای رسم ها نیز از کتابخانه ی matplotlib و برای رسم های مقایسه ای از تابع subplot این کتابخانه استفاده شد.



پرسش : تفاوت شنیداری بین حالت ۸ بیت و ۳ بیت چیست؟ نویز کوانتیزاسیون در کدام حالت

غالب است؟

پاسخ: به این علت که در حالت ۸ بیت تعداد سطوح گسسته ی بیشتری برای دامنه وجود دارد، این صدا تشابه زیادی با صدای اصلی دارد و مقدار نویز اندکی دارد؛ اما در حالت ۳ بیت تعداد سطوح موجود برای دامنه کمتر است و صدای کوانتیزه شده تکه تکه و با نویز شدیدتر است و نویز در این حالت غالب است.

## گام دوم : مدولاسیون تقسیم فرکانسی (FDM)

توضیح مفهومی : برای ارسال همزمان ۳ صدا روی یک کانال مشترک، هر پیام را به یک فرکانس حامل متفاوت منتقل میکنیم تا طیف آن ها با هم تداخل نکنند.

در این قسمت از پروژه هر سیگنال اصلی ضبط شده را با کمک فرمول زیر در یک موج حامل ضرب، و نتایج با هم جمع شدند.

$$y = signal * np.\cos(2\pi f_c t)$$

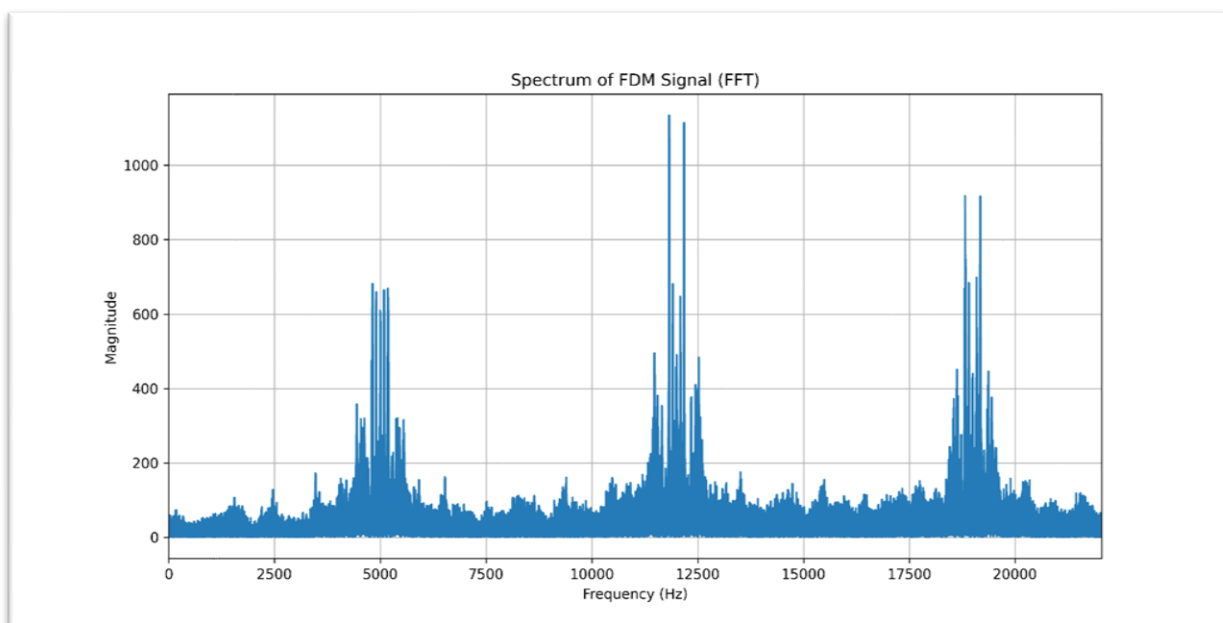
$$Y_T = y_1 + y_2 + y_3$$

که در آن تابع  $np.\cos$  تابع ریاضیاتی کسینوس است و  $f_c$  فرکانس حامل می باشد.

فرکانس های حامل:

$$f_{c1} = 5 \text{ kHz}, f_{c2} = 12 \text{ kHz}, f_{c3} = 19 \text{ kHz}$$

سپس با استفاده از توابع  $np.fft.fft$  و  $np.fft.fftfreq$  طیف فرکانسی (FFT) به دست آمده و مشابه با قبل برای سیگنال مجموع رسم شد.



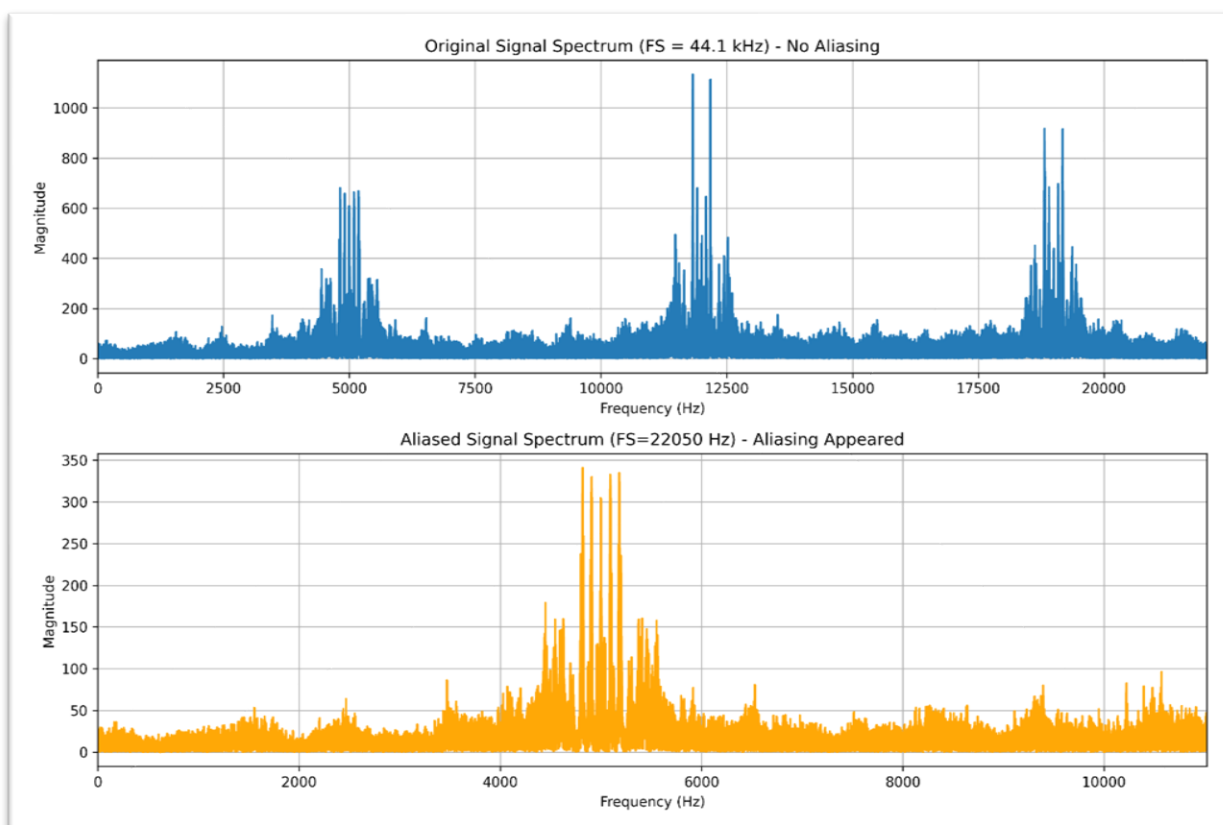
### پرسش : آیا در نمودار طیف، سه بخش مجزا مربوط به سه پیام را مشاهده میکنید؟

پاسخ : بله در طیف فرکانسی سیگنال مجموع، سه ناحیه‌ی مجزا در اطراف فرکانس‌های 5 kHz و 12 kHz و 19 kHz مشاهده می‌شود که متناظر با سه سیگنال پیام مدوله‌شده هستند. این جدایی طیفی نشان‌دهنده‌ی عملکرد صحیح مدولاسیون تقسیم فرکانسی (FDM) و عدم تداخل بین کانال‌ها است. در صورت رسم نمودار برای فرکانس‌های منفی، به علت حقیقی بودن سیگنال همین شکل از سیگنال به صورت تقارن یافته مشاهده می‌شد. (برای نمودارهای طیف فرکانسی بعدی نیز به همین ترتیب است.)

## گام سوم : بررسی قضیه نمونه برداری نایکوئیست (Nyquist)

توضیح مفهومی : طبق قضیه نایکوئیست، نرخ نمونه برداری ( $f_s$ ) باید حداقل دو برابر بزرگترین فرکانس موجود در سیگنال باشد تا بتوان سیگنال اصلی را با استفاده از آن بازیابی کرد.

در این قسمت از پروژه ، سیگنال اصلی یک بار با نرخ اصلی و یک بار با نرخی کمتر از دو برابر بزرگترین فرکانس حامل (در اینجا 20 kHz) نمونه برداری شد. در این مرحله، به منظور تغییر نرخ نمونه برداری سیگنال، از تابع `signal.resample(Yt,N)` استفاده شد. این تابع با تعیین تعداد نمونه‌های جدید ( $N$ )، سیگنال را در حوزه فرکانس بازنمونه برداری کرده و آن را به طول جدید تبدیل می‌کند. بدین ترتیب، نرخ نمونه برداری سیگنال از مقدار اولیه به مقدار موردنظر تغییر یافت.





پرسش : پدیده aliasing در کدام حالت رخ می دهد؟ این پدیده چه تغییری در محتوای

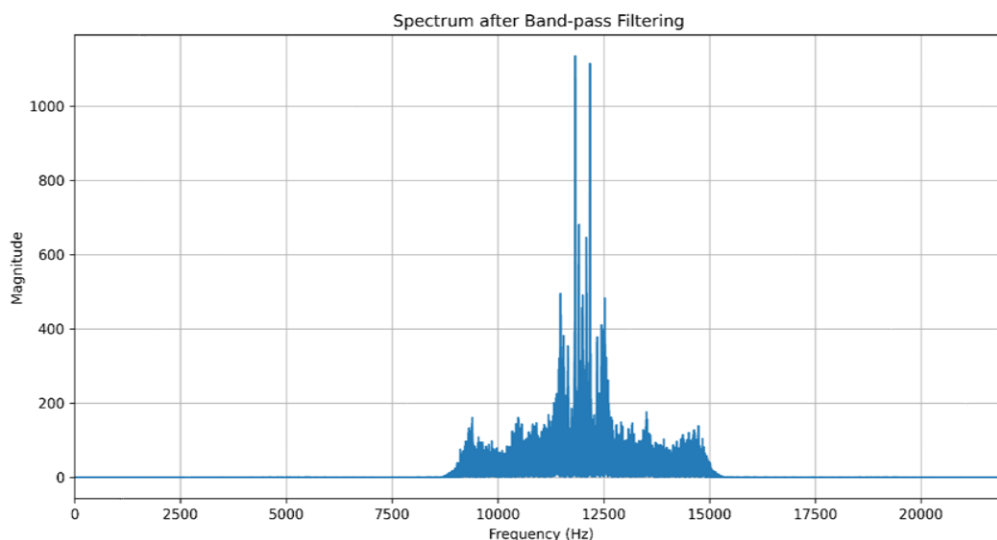
فرکانسی سیگنال ایجاد کرده است؟

پاسخ : در حالت با نرخ 20 kHz پدیده aliasing رخ میدهد. این پدیده باعث ایجاد تداخل در محتوای فرکانسی می شود و دیگر نمی توان از روی آن سیگنال (صدای) اصلی را بازیابی کرد. و میتوان دید در واقع طیف فرکانسی نمونه برداری شده با نرخ پایینتر، بخشی از طیف فرکانسی نمونه با نرخ اصلی است.

## گام چهارم: گیرنده و طراحی فیلتر میان گذر (BPF)

توضیح مفهومی: در سمت گیرنده ما با یک سیگنال ترکیبی (FDM) روبرو هستیم که شامل تمامی پیام هست. فیلتر میان گذر مانند یک غربال عمل می کند که فقط اجازه عبور به فرکانس های بازه مشخصی را می دهد. در این مرحله ما قصد داریم با استفاده از یک فیلتر دیجیتال، فقط محتوای مربوط به فایل دوم (اعداد ۱۱ تا ۲۰) را که در فرکانس 12 kHz مدوله شده است، استخراج کنیم.

در این قسمت از پروژه با ماژول signal از کتابخانهی scipy و با استفاده از تابع signal.firwin یک فیلتر میان گذر دیجیتال ایجاد می شود که فرکانس های قطع آن 9 kHz و 15 kHz می باشد. سپس آن را با کمک تابع signal.lfilter روی سیگنال مجموع اعمال نموده تا بخشی از سیگنال که خواسته شده بود به دست آید. سپس نمودار سیگنالی که فیلتر بر روی آن اعمال شده رسم می شود. در آخر نیز صدای فیلتر شده را خروجی گرفته و با نام extracted\_y2.wav ذخیره می گردد.



### پرسش ۱: ماهیت صوتی سیگنال خروجی فیلتر را تحلیل کنید.

پاسخ: از روی صدای خروجی ذخیره شده نمی توان تشخیص داد که محتوای آن چه چیزی است و صدایی که به گوش میرسد شبیه به سوت و یا جیغ است. به این دلیل که سیگنال هنوز روی حامل 12 kHz مدوله شده است و عمل دمدولاسیون روی آن انجام نگرفته است این اتفاق رخ می دهد.

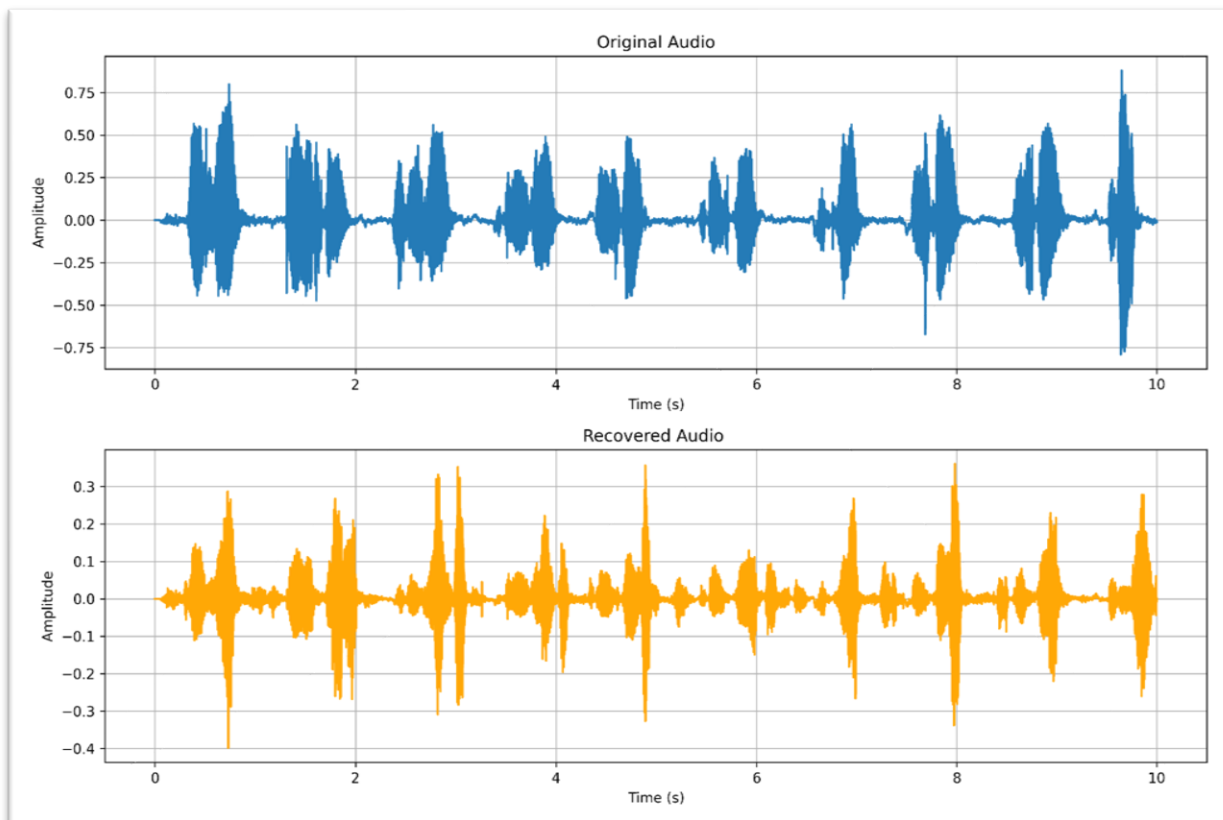
### پرسش ۲: فیلتر چقدر در حذف پیام های اول و سوم موفق بوده است؟

پاسخ: با توجه به طیف خروجی فیلتر میان گذر، مشاهده می شود که محتوای فرکانسی سیگنال باقی مانده در بازه فرکانسی مربوط به کانال دوم متمرکز شده است و مؤلفه های مربوط به پیام های اول و سوم به میزان زیادی تضعیف شده اند. بنابراین فیلتر اصل محتوای فرکانسی مربوط به صدای دوم را نگه میدارد و عملکرد مناسبی در جداسازی کانال دوم دارد.

## گام پنجم : دمدولاسیون و بازیابی نهایی (LPF & Demodulation)

توضیح مفهومی : پس از جداسازی سیگنال در مرحله قبل، برای بازگشت به صدای اصلی، باید عمل دمدولاسیون انجام شود. این کار با ضرب مجدد سیگنال فیلتر شده در همان موج حامل (Carrier) انجام می گیرد. این عمل باعث می شود سیگنال به باند پایه (فرکانس صفر) برگردد. در نهایت ، با استفاده از یک فیلتر پایین گذر (LPF)، مؤلفه های فرکانس بالای ناخواسته حذف شده و صدای اصلی بازیابی می شود.

در این قسمت از پروژه سیگنال خروجی فیلتر به دست آمده در گام قبل را مجدداً در حامل خود (12 kHz) ضرب کرده و با عبور از فیلتر پایین گذر (LPF) با فرکانس قطع 4 kHz، صدای اصلی را بازیابی کرده. برای ضرب حامل مانند گام اول، و برای ایجاد فیلتر و اعمال آن مانند گام چهارم عمل انجام می شود. سپس نمودار حوزه زمان سیگنال باز سازی شده با نمودار سیگنال اصلی رسم و مقایسه می شود و فایل بازیابی شده با نام recovered\_audio2.wav ذخیره می گردد.



### پرسش : در صدای بازسازی شده نهایی آیا اعداد ۱۱ تا ۲۰ به وضوح شنیده میشوند؟

پاسخ : صدای بازپایی شده به علت وجود قسمتی از سیگنال که مربوط به ضرب شدن حامل با فرکانس دوبرابر است، دارای نویز می باشد اما اعداد به وضوح شنیده میشوند.

### پرسش : چرا در مرحله دمدولاسیون ، سیگنال در عدد ۲ ضرب شده؟

پاسخ : وقتی سیگنال را دوباره در حامل خود ضرب می شود به عبارت  $x \cdot \cos^2(2\pi f_{c2}t)$  رسیده و از

$$\text{روابط مثلثاتی داریم : } \cos^2(\theta) = \frac{1+\cos(2\theta)}{2}$$

حال اگر سیگنال ضرب در  $\frac{(1+\cos(4\pi f_{c2}t)) \times 2}{2}$  شود می توان سیگنال را به دو بخش تقسیم کرد که

یک بخش سیگنال اصلی است و بخش دیگر ضرب شده ی سیگنال در حامل با دوبرابر فرکانس مربوطه است. و چون یک فیلتر پایین گذر روی سیگنال اعمال می شود ؛ اصل محتوایی که توسط حامل ایجاد شده، حذف شده و مقدار کمی از آن در بخش فیلتر شده باقی می ماند که همین بخش باقی مانده باعث نویز می شود. در نتیجه این ضرب در ۲ به این خاطر بود که بتوان به همان دامنه ی سیگنال اصلی رسید.