

به نام خدا

گزارش پروژه سیگنال ها و سیستم ها

محمد صفری

۹۸۳۱۱۳۸

## بخش اول

### تجزیه و تحلیل طیفی

مقدمه

تجزیه و تحلیل فرکانسی یک راه دیگر برای نگاه کردن به داده ها است. تحلیل فرکانس به جای مشاهده داده ها در حوزه زمانی، داده های زمانی را در سری امواج سینوسی تجزیه می کند. تبدیل فوریه سریع (FFT) یک روش ریاضی برای تبدیل تابع زمان به تابعی از فرکانس است.

تجزیه و تحلیل فرکانسی به جای مشاهده داده ها در حوزه زمان، با برخی از تحلیل فرکانس ریاضیات نه چندان دشوار، اما مبتکرانه، داده های زمانی را در سری امواج سینوسی تجزیه می کند.

همچنین می توان گفت که تحلیل فرکانس وجود فرکانس های ثابت خاصی را بررسی می کند.

فوریه ثابت کرد که هر تابع پیوسته می تواند به صورت مجموع نامتناهی از امواج سینوسی و کسینوس تولید شود. یک موج سینوسی خالص می تواند توسط یک بلندگو به صدا تبدیل شود و به عنوان یک صدای ثابت و خالص از یک گام واحد درک شود. صداهای سازهای ارکستر معمولاً از یک هارمونیک بنیادی و مکمل تشکیل می شوند که می توان آن را برهم نهی امواج سینوسی با فرکانس اصلی  $f$  و مضرب های صحیح آن فرکانس در نظر گرفت.

تحلیل فوریه یک تابع تناوبی به استخراج مجموعه ای از سینوس ها و کسینوس ها اشاره دارد که وقتی روی هم قرار می گیرند، تابع را بازتولید می کنند. این تحلیل را می توان به صورت سری فوریه بیان کرد.

برای داده های گسسته، مبنای محاسباتی تحلیل طیفی تبدیل فوریه گسسته (DFT) است. DFT داده های مبتنی بر زمان یا فضا را به داده های مبتنی بر فرکانس تبدیل می کند.

تبدیل فوریه سریع

تبدیل فوریه سریع یک روش ریاضی برای تبدیل تابع زمان به تابعی از فرکانس است. به عنوان تبدیل از حوزه زمان به حوزه فرکانس توصیف می شود.

تبدیل فوریه سریع (FFT) پیاده سازی از تبدیل فوریه گسسته (DFT) است که عبارت های تکراری را در الگوریتم ریاضی حذف می کند تا تعداد عملیات ریاضی انجام شده را کاهش دهد. به این ترتیب می توان از تعداد زیادی نمونه بدون کاهش سرعت تبدیل استفاده کرد. FFT محاسبات را با ضریب  $\frac{N}{\log_2(N)}$  کاهش می دهد.

## تجزیه و تحلیل طیف توان

طیف توان برای یک سری زمانی  $S_{xx}(f)$  توصیف کننده توزیع توان به مولفه‌های فرکانسی تشکیل دهنده سیگنال است. بر اساس رابطه آنالیز فوری، هر سیگنال فیزیکی را می‌توان به تعدادی از فرکانس‌های گسسته، یا طیفی از فرکانس‌ها در یک محدوده پیوسته تجزیه کرد. میانگین آماری یک سیگنال معین یا نوعی از سیگنال (شامل نویز) وقتی که از دیدگاه محتوای فرکانس آن تحلیل می‌شوند، **طیف** آن سیگنال نامیده می‌شوند.

وقتی که انرژی سیگنال در حول و حوش یک بازه زمانی محدود متمرکز شود، مخصوصاً اگر کل انرژی آن محدود باشد، می‌توان چگالی طیفی انرژی آن را محاسبه کرد. چیزی که معمول تر است، چگالی طیفی توان (طیف توان) است که به سیگنال‌های موجود در کل زمان، یا به یک دوره زمانی به اندازه کافی بزرگ (مخصوصاً در رابطه با مدت زمان اندازه‌گیری)، اعمال می‌شود. چگالی طیف توان (PSD) به توزیع انرژی طیفی که «در هر واحد زمان» یافت می‌شود، اشاره دارد، زیرا انرژی کلی در تمام زمان معمولاً بینهایت است و برای همین «چگالی» آن محاسبه می‌شود. مجموع‌یابی یا انتگرال‌گیری مولفه‌های طیفی منجر به توان کلی (برای یک فرایند فیزیکی) یا واریانس (در یک فرایند آماری) می‌شود، این مشابه آن چیزی است که از انتگرال‌گیری  $x^2(t)$  روی دامنه زمانی به دست می‌آید، که این موضوع در قضیه پارسوال نیز دیده می‌شود.

## گزارش

ابتدای کار برای آنکه بتوانیم از فایل‌های صوتی داده شده، اطلاعات را به صورت یک سیگنال بررسی کنیم، از کتابخانه wavfile موجود در scipy.io استفاده می‌کنیم. از انجا صدای فایل‌های داده شده، به صورت استریو و از طریق دو کانال قابل پخش است، به ازای هر فایل ورودی، یک آرایه دو بعدی از داده‌ها به همراه نرخ نمونه برداری صدا دریافت می‌کنیم. از انجا که قرار است صدا به عنوان یک سیگنال واحد پردازش شود، دو سیگنال استریو را تلفیق کرده و در پردازش‌های آتی از آرایه صدای تلفیقی استفاده می‌شود.

سیگنال صدای داده شده حقیقی است و طیف (spectrum) متقارنی دارد. برای محاسبه چگالی طیف توان، به صورت یک طرفه، ابتدا یک نیمه از تبدیل فوری سیگنال را برداشته و توان دوم اندازه مطلق آنها را بر کل زمان نمونه برداری تقسیم می‌کنیم تا توان به چگالی توان یا به بیان دیگر چگالی انرژی به چگالی توان تبدیل شود. کل مدت زمان فایل صوتی یا نمونه برداری در حقیقت همان ضرب تعداد نمونه‌ها در فرکانس [نرخ] نمونه برداری است.

$$T_{total} = f_s \cdot n$$

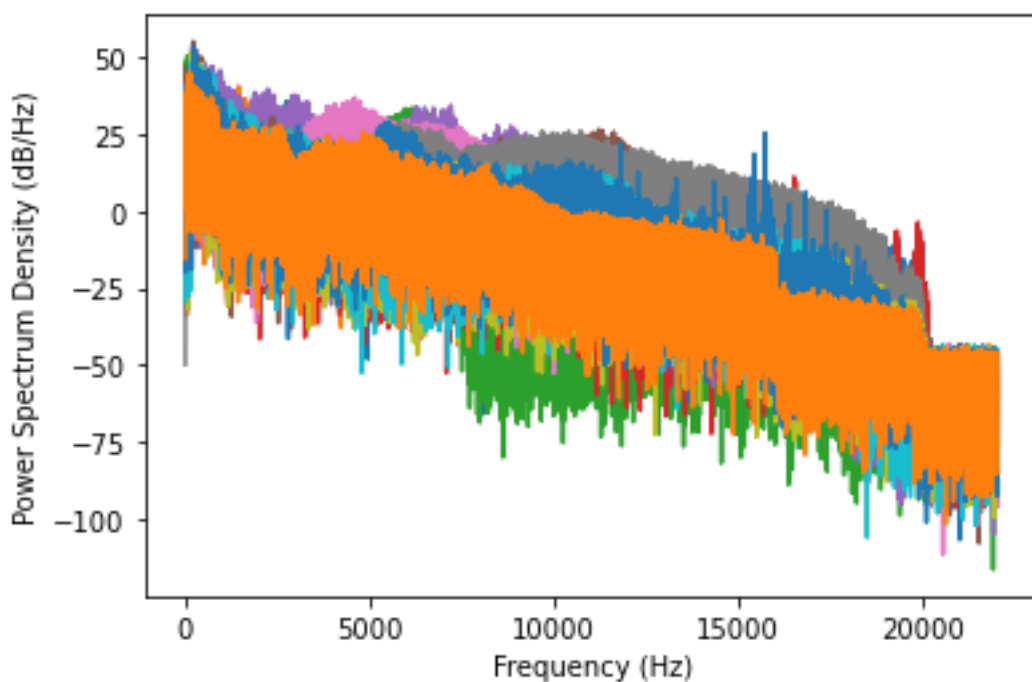
از انجا که طیف متقارن است (سیگنال حقیقی است) و ما مقادیر توان طیفی یه نیمه را در محاسبه اعمال نکردیم، اندازه نیمه دیگر را دوبار می‌کنیم تا در انرژی کل تغییری بوجود نیاید.

$\{negative\ frequencies\ energy\}, S_0, \{positive\ frequencies\ energy\}$

محور فرکانس (فرکانس های مثبت) نیز از ۰ تا نصف فرکانس نمونه برداری به تعداد نمونه ها خواهد بود.

$$0, \frac{f_s}{n}, \frac{2f_s}{n}, \dots, \frac{\frac{n}{2}f_s}{n} (= \frac{f_s}{2})$$

تابع `calcSpectrum` با دریافت سیگنال و فرکانس (نرخ) نمونه برداری، محور فرکانس و مقادیر چگالی طیف را برای ما محاسبه می کند و بر می گرداند.



تصویر ۱: نمودار چگالی طیف توان نمونه فایل ها

تابع `getMaxFreq`، طیف با بیشترین توان را برمیگرداند. از آنجا که فرکانس تولید شده توسط تار های صوتی انسان به طور معمول بین ۵۰ تا ۳۰۰ است با محاسبه اندیس مناسب در ارایه طیف توان، ناحیه فرکانسی مذکور را بررسی میکند تا فرکانس بیشترین توان را پیدا کند.

```
rangeIndex[0] = int((freqRange[0] * len(freqAxis)) // totalRange)
rangeIndex[1] = int((freqRange[1] * len(freqAxis)) // totalRange)
rangeIndex[1] = len(freqAxis) if len(freqAxis) < rangeIndex[1] else rangeIndex[1]
```

تکه کد ۱: محاسبه اندیس محدوده فرکانس های مورد بررسی در چگالی طیف توان

در نهایت تابع `maleOrFemale` بر اساس اینکه این فرکانس در چه رنجی قرار دارد، برای هر فایل صوتی مشخص میکند که صدای غالب، صدای مرد یا زن بوده است:

```
voice 0: [211.02 HZ] female
voice 1: [110.43 HZ] male
voice 2: [050.05 HZ] male
voice 3: [131.32 HZ] male
voice 4: [283.72 HZ] female
voice 5: [205.49 HZ] female
voice 6: [086.56 HZ] male
voice 7: [181.34 HZ] female
voice 8: [149.90 HZ] male
voice 9: [180.00 HZ] female
voice 10: [218.05 HZ] female
voice 11: [128.95 HZ] male
```

تصویر 2: خروجی تشخیص دهنده

برای بهینه تر کردن این تابع صرفاً از ناحیه مربوط به گفتار انسان فرکانس بیشترین توان اعلام می شود.

## بخش دوم

### تولید نویز در کانال awgn

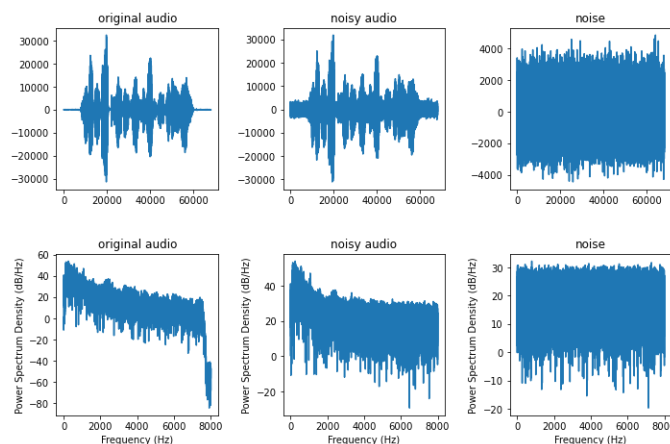
نویز سفید به نویزی اطلاق می‌شود که تمام فرکانس‌ها در سراسر طیف صدای قابل شنیدن را به یک اندازه در بر می‌گیرد. زیرا نویز سفید در بر می‌گیرد. از آنجایی که نویز سفید چندین باند صدا را در بر می‌گیرد، گاهی اوقات به آن نویز پهن باند می‌گویند. نویز سفید در مقابل نویز های رنگی (colored) یکی از بدترین انواع نویز هستند زیرا توزیع آنها به توزیع داده های سیگنال وابستگی ندارد و به سختی میتوان آنها را صدای اصلی جدا کرد. درحالی که بررسی این نویز در مقابل انواع دیگر، از ساده ترین ترین هاست زیرا از توزیع متقارن نرمال یا گاوسی تبعیت میکند.

یکی از پارامترهای اصلی در این کانال، نسبت سیگنال به نویز یا SNR (signal to noise ratio) است. با فرض کانالی با پهنای باند B، توان سیگنال دریافتی  $P_r$  و چگالی طیفی توان (PSD) نویز  $\frac{N_0}{2}$ ، نسبت سیگنال به نویز (SNR) توسط رابطه زیر محاسبه میشود:

$$\gamma = \frac{P_r}{N_0 B}$$

در تابع awgn با گرفتن این نسبت در مقیاس لگاریتمی یا dB سیگنال ها، توان نویز یا همان  $N_0$  محاسبه شده و مقادیر سیگنال نویز توسط تابع standard\_normal (توزیع نرمال استاندارد) ساخته شده و با اضافه کردن آن سیگنال نویز دار تولید می شود.

به طور مثال برای توان نویز معادل با توان سیگنال صدای نمونه یا نسبت SNR(dB) ۱۰ خواهیم داشت:



تصویر ۳: تولید سیگنال نویز دار و نمودار سیگنال و چگالی طیف آن

## الگوریتم spectral subtraction

تفریق طیفی از نظر تاریخی یکی از اولین الگوریتم‌هایی است که برای تقویت گفتار تک کانالی پیشنهاد شده است. در این روش، طیف نویز در حین مکث گفتار تخمین زده می شود و برای تخمین گفتار شفاف، از طیف گفتار پر سر و صدا کم می شود.

ای الگوریتم همچنین با ضرب طیف گفتار پر سر و صدا با تابع افزایش و ترکیب آن با فاز گفتار پر سر و صدا به دست می آید. اشکال این روش وجود اعوجاج های پردازشی است که به آن نویز باقیمانده می گویند. تعدادی از تغییرات این روش در طول سال های گذشته برای رفع اشکال توسعه یافته است. این گونه ها خانواده ای از الگوریتم های طیفی از نوع تفریق را تشکیل می دهند. تفریق طیفی پایه، تفریق بیش از حد طیفی، تفریق طیفی چند باندی، فیلتر وینر، تفریق طیفی تکراری، و تفریق طیفی بر اساس ویژگی های ادراکی. برای آزمایش عملکرد الگوریتم های تفریقی یا کاهشی، معیارهای هدف (SNR و PESQ)، طیف نگارها و تست های شنیداری غیررسمی برای انواع نویزهای ثابت و غیر ثابت در سطوح مختلف SNR انجام می شوند.

## پیاده سازی الگوریتم spectral subtraction یا کاهش طیفی

برای گرفتن نویز در حوزه فرکانسی (طیفی) بدون داشتن اطلاعاتی از توزیع و نمونه های سیگنال نویز در این گزارش، از سه روش اقدام شده است:

- حذف طیف های بالاتر از فرکانس معین
- حذف طیف های بالاتر به کمک فیلتر پایین
- تعدیل توان طیف ها به کمک تخمین توان طیف و روش windowing

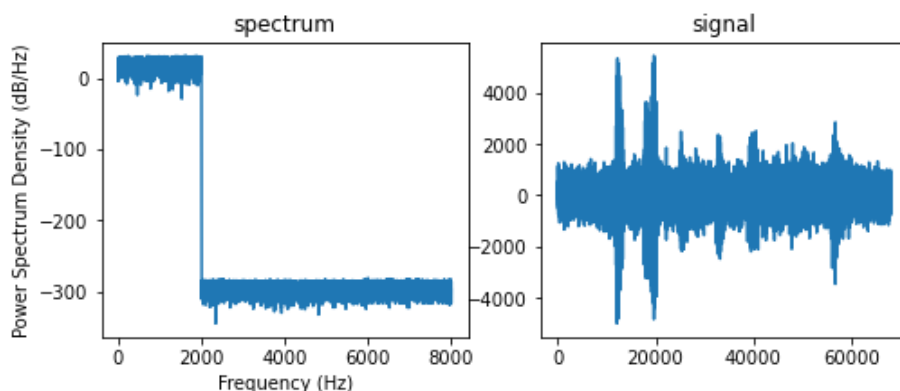
## حذف طیف های بالاتر از فرکانس معین

تابع restrictSpectrum با محاسبه سریع طیف به صورت متقارن، با فرض فرکانس قطع  $f_c$ ، ناحیه بزرگ تر از  $f_c$  و کوچکتر از  $-f_c$  را صفر میکند و در نهایت با بازسازی سری فوریه سیگنال جدید با توجه به فاز سیگنال ورودی و نهایتاً تابع معکوس فوریه (سنتز سیگنال) تلاش میکند نویز را بگیرد:

```
signal = ifft(amplitude * np.exp(np.angle(fftshift(ff)) * 1j))
```

تکه کد ۲: بازسازی سیگنال جدید از روی توان با توجه به فاز سیگنال ورودی

همان طور که در شکل زیر مشاهده میکنیم توان نویز تا حدود زیادی کاهش یافته، اما به علت سیاست سخت حذف طیف های بالا شدت صدا نیز کم شده است. با این حال وضوح صدای گوینده تحت تاثیر قرار نگرفته است زیرا به طیف ناحیه گفتاری انسان کار نداشته ایم.



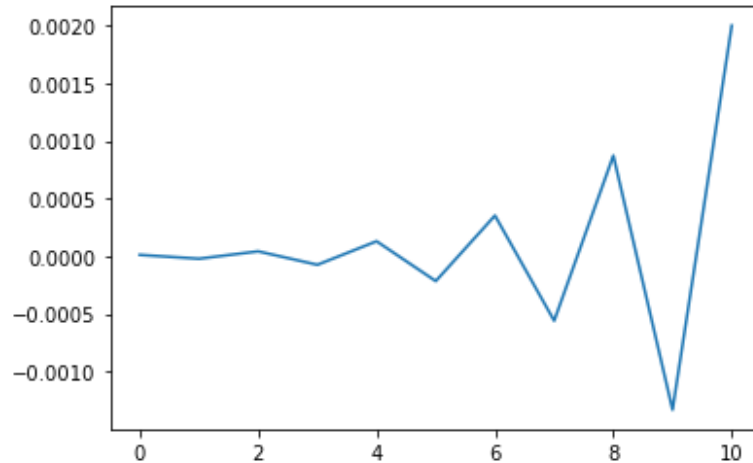
تصویر ۴: خروجی نویزگیری سیگنال با حذف طیف های بالا و بررسی چگالی طیف آن (فرکانس قطع ۲۰۰۰)

در روش بعد با کمک فیلتر پایین گذر ملاحظه بیشتری در رابطه با سیاست حذف کامل طیف ها خواهیم داشت. یکی از عللی که این روش برای نمونه تاثیر خیلی خوبی ندارد، نرخ نمونه برداری پایین 16 k صدا است.

حذف طیف های بالاتر به کمک فیلتر پایین گذر

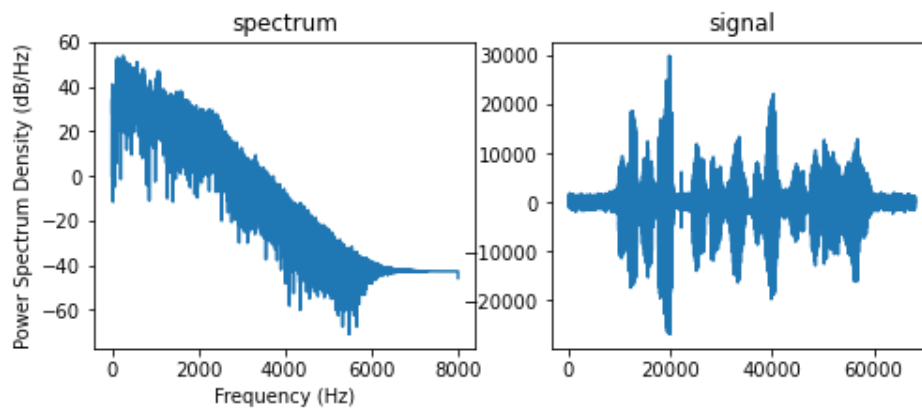
برای اعمال فیلتر باید ضرایب فیلتر را به ازای فرکانس قطع معین محاسبه کرده و در نهایت با سیگنال اولیه **convolve** کنیم. از آنجا که سعی میکنیم ضرایب به شکلی باشد که سیگنال قابل بازسازی باشند، فرکانس قطع را به کمک نرخ نایکوئست، نرمال میکنیم. برای تولید این ضرایب از تابع **butter** در ماژول سیگنال کتابخانه **scipy** استفاده میکنیم.





تصویر ۵: ضرایب فیلتر با مرتبه ۱۰، و فرکانس قطع ۲۰۰۰ (در حوزه زمان)

با اعمال فیلتر شاهد هستیم که این تقطیع کمی ملایم تر است، چرا که به جای صفر کردن ناگهانی فرکانس های مجاور، با یک ضریب کوچک توان آنها کمتر می شود. بنابراین شدت صدا میزان افت کمتری دارد در حالی که نویز تقریباً با خروجی اول تفاوتی نمیکند. این روش شاید برای این مورد کاربرد سودمندی بیشتری نداشته باشد اما در مورد های کاربری که طیف های بالا نیز اهمیت دارند، از روش قبلی کارا تر است.



تصویر ۶: خروجی نویزگیری سیگنال به کمک فیلتر و بررسی چگالی طیف آن (فرکانس قطع ۲۰۰۰)

تعدیل توان طیف ها به کمک تخمین توان طیف و روش windowing

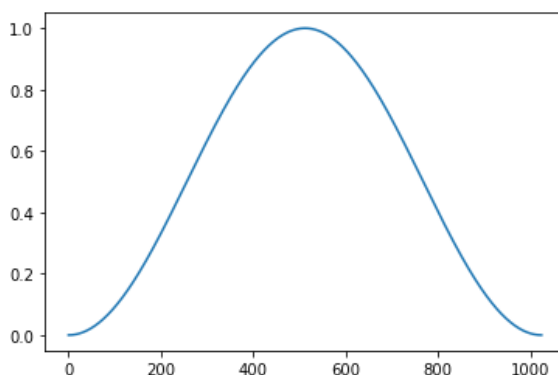
در همه شرایط، ما همه داده ها را از پیش نداریم و نیاز داریم تا در لحظه فرایند نویزگیری را انجام دهیم؛ پس:

1 - باید حجم محاسبات را معقول کنیم

2 - باید الگوریتم را قابل اجرا برای فریم های با اندازه معین از داده های سیگنال بکنیم.

نیازمندی اول ما را ملزم به استفاده از یک تخمین برای محاسبه spectrum میکند که حجم کمتری از محاسبات را برای سیستم میزبان به همراه داشته باشد. در اینجا از تخمین welch برای این امر استفاده شده است که میانگین توان هر فریم یا پنجره را خروجی میدهد.

برای اینکه بتوانیم ساده تر فرایند فریم کردن یا windowing را بر روی سیگنال داشته باشیم از شیوه hanning استفاده میکنیم که از موج های کسینوسی برای windowing استفاده میکند.

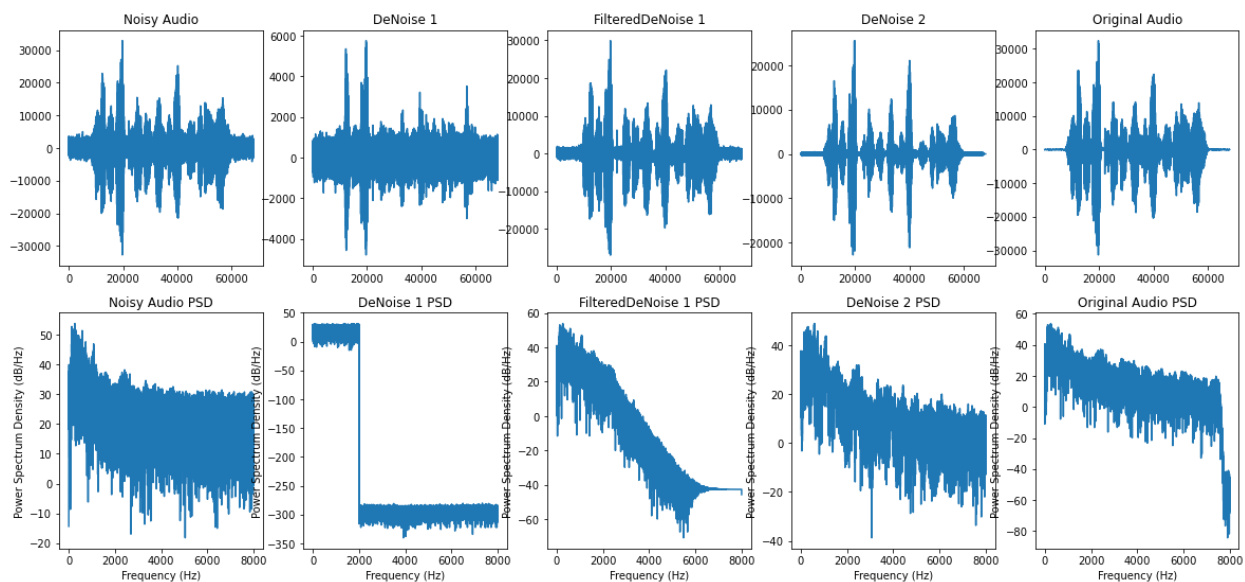


تصویر ۷: مقادیر hanning

در برداشت هر فریم از سیگنال، با صدا زدن `denoiseWindowByNoisePow` فرایند نویزگیری از هر بخش انجام می شود و بخش به بخش سیگنال بازسازی می شود.

در این متد، در هر فریم یک بار تلاش می شود به کمک مقادیر hanning سیگنال بازسازی شود و به اندازه یک ضریب معین (در اینجا 0.5) از توان تخمین زده شده (میانگین فریم) از توان طیف بازسازی شده سیگنال کاسته شده و در نهایت با یک نسبت مناسب از توان طیف بازسازی شده را در طیف اصلی جایگذاری کنیم (در اینجا کاملاً جایگذاری می شود). توجه میکنیم در صورتی که حین تفریق توان مقدار منفی شد، از ماکسیمم تضعیف شده نزدیک به صفر، توان اصلی سیگنال استفاده میکنیم.

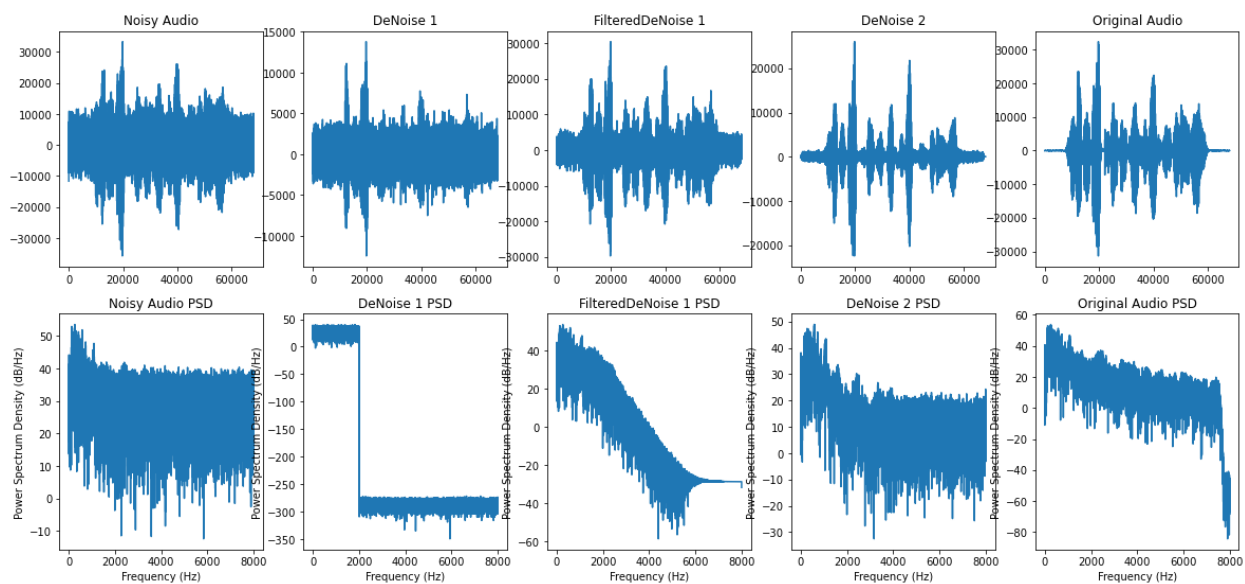
در نهایت با توان طیف تولید شده، با توجه به فاز سیگنال نویز دار، سیگنال بدون نویز شده را تولید میکنیم.



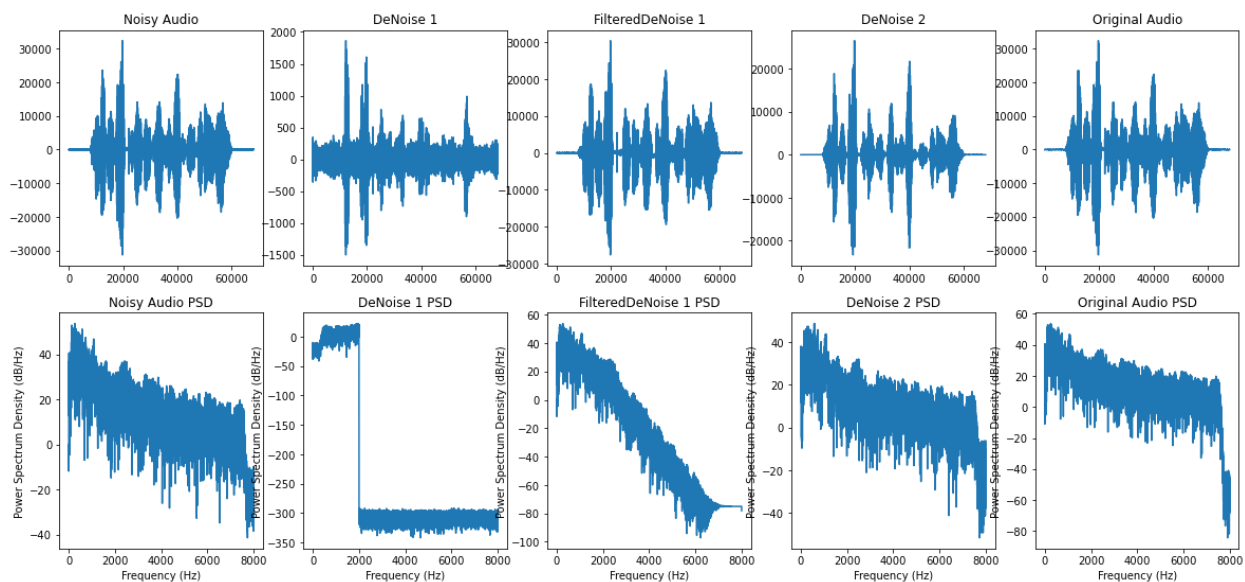
تصویر ۸: مقایسه نهایی روش های نویزگیری در مثال گزارش (TEST.Wav, SNR(dB) 10, fc=2000)

در این مثال این روش بیشترین دقت در حذف نویز با کمترین کاهش توان را دارد.

با نویز های قوی تر و ضعیف تر خروجی زیر را داریم:



تصویر ۹: مقایسه نهایی روش های نویزگیری در مثال گزارش (TEST.Wav, SNR(dB) 1, fc=2000)



تصویر \* ۱: مقایسه نهایی روش های نویزگیری در مثال گزارش (TEST.Wav, SNR(dB) 50, fc=2000)

در انتها نیز یک شبیه سازی نزدیک به عملکرد تخمین welch (تعدیل نشده) و مقایسه ان با تابع کار رفته welch آورده شده است.