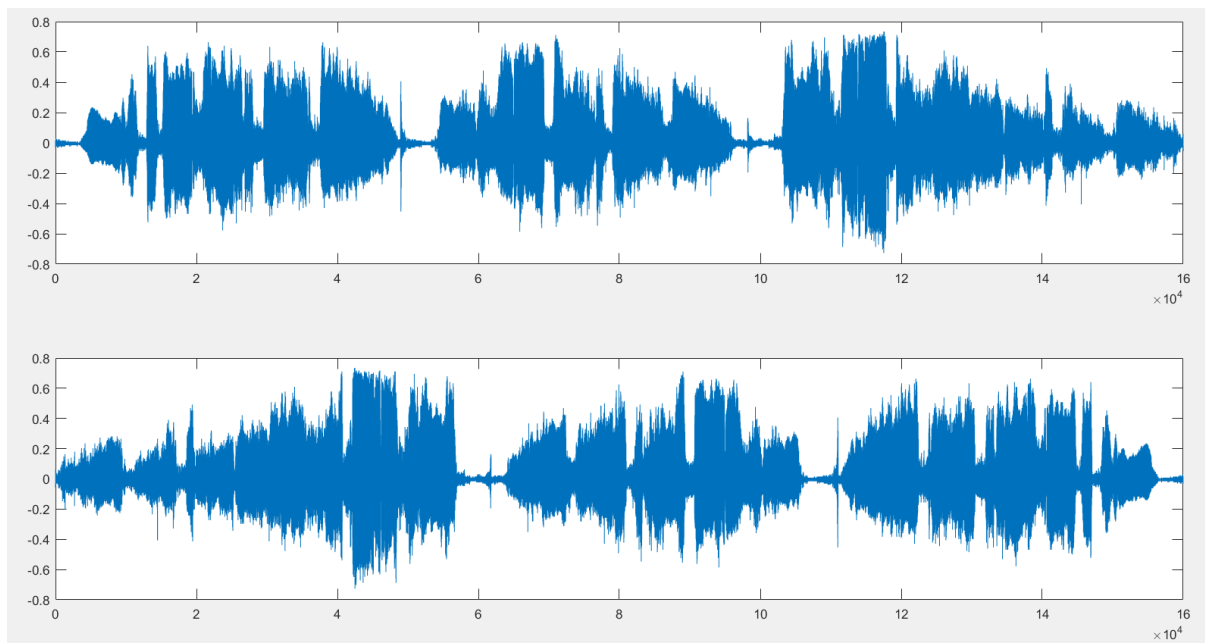


در این آزمایش قصد داریم که فیلترهای مختلفی را برای چند صدا اعمال کنیم و خروجی آن‌ها را بررسی کنیم. کارهای مربوط به اکوی صدا در پروژه‌های قبلی انجام شده بود به همین دلیل از انجام دوباره‌ی اعمال اکو بر روی صدا خودداری شده است و کارهای نو و جدیدی را انجام داده‌ایم.

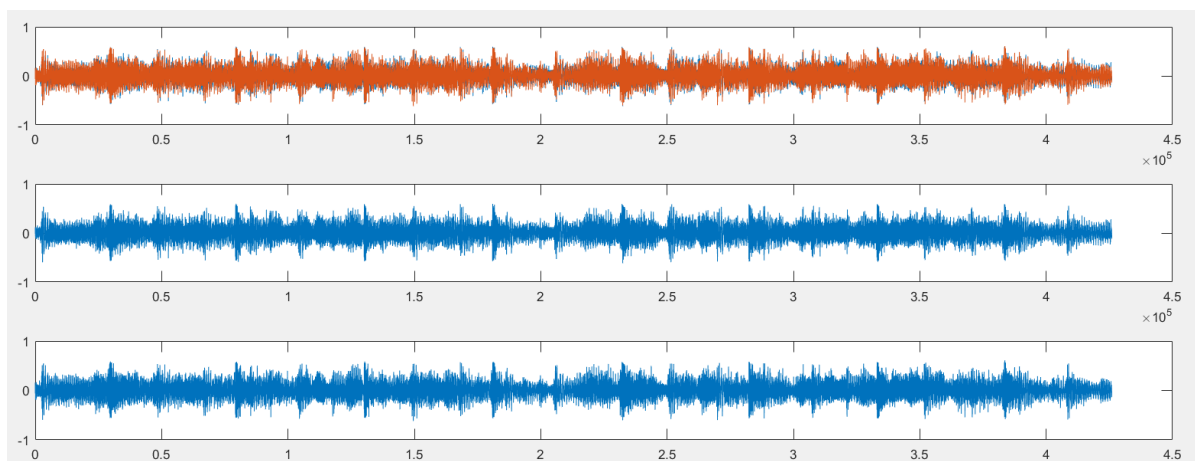
بخش اول

اولین کاری که می‌کنیم، پخش کردن صدا از آخر به اول است که با برعکس کردن آرایه‌ی مربوط به آن انجام می‌شود. برنامه‌ی مربوط به این کار p1.m است که ورودی این برنامه، sound1 و خروجی آن out1 می‌باشد. نمودار سیگنال ورودی و خروجی به شکل زیر است.



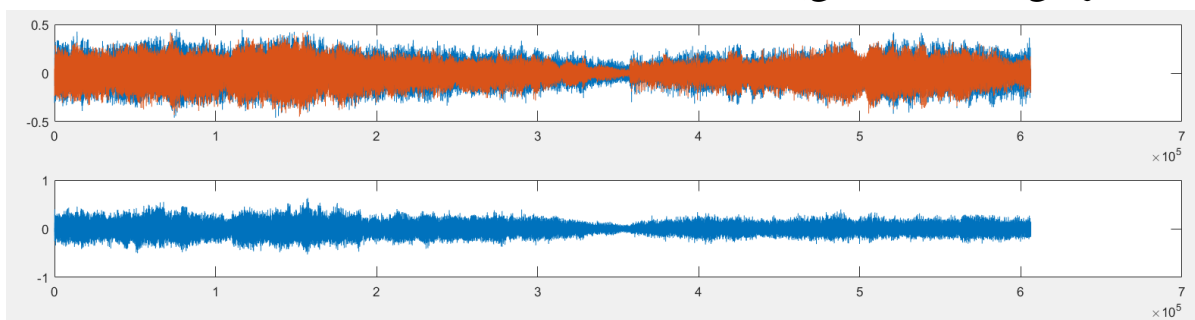
بخش دوم

در ابتدا قصد داریم که با گرفتن یک صدای استریو از ورودی، کانال راست و چپ آن را جدا کنیم و آن‌ها را پخش کنیم. صدای دو کاناله، آرایه‌ای $n \times 2$ است که ستون اول آن مربوط به یک کانال و ستون دیگر مربوط به کانال دیگر است که ما از آن‌ها به عنوان left و right یاد می‌کنیم. نام این برنامه p2.m است که ورودی آن فایل hootie.wav است و خروجی آن فایل‌های right2 و left2 هستند که کانال‌های راست و چپ آن می‌باشند. نمودار آن‌ها:



بخش سوم

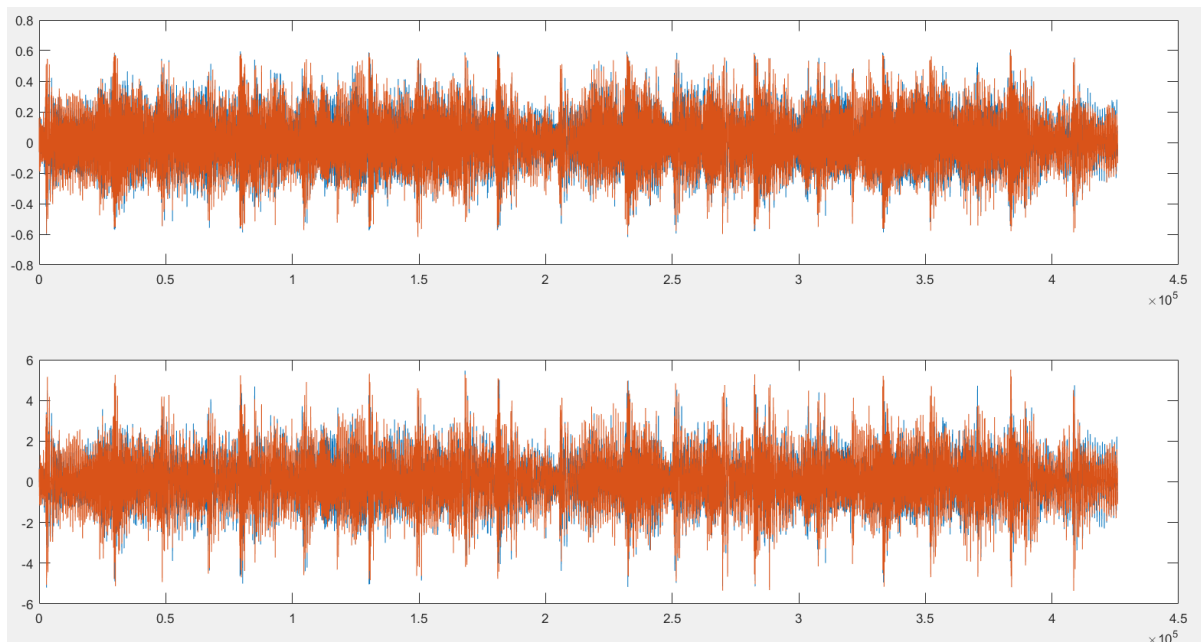
در این بخش می‌خواهیم صدای خواننده را حذف کنیم و تنها صدای موزیک بیاید. در اکثر موسیقی‌های فعلی، صدای خواننده در کانال راست و چپ به یکی است و یا بسیار شبیه است اما صدای موسیقی این‌گونه نیست و در کانال راست و چپ تفاوت‌های محسوسی دارد. به همین دلیل سعی می‌کنیم که با کم کردن یک کانال از دیگری، صدای خواننده را حذف کنیم. این برنامه p3.m نام دارد که ورودی آن road است و خروجی آن karaoke3 می‌باشد.



بخش چهارم

در این بخش می‌خواهیم اجزای فرکانس بالا را نرم‌تر کنیم و بخش با فرکانس پایین را نگه داریم. مانند این است که treble روی بلندگو را تا انتها کم کنیم. در این جا برای هر کانال از فرمول زیر استفاده می‌کنیم:

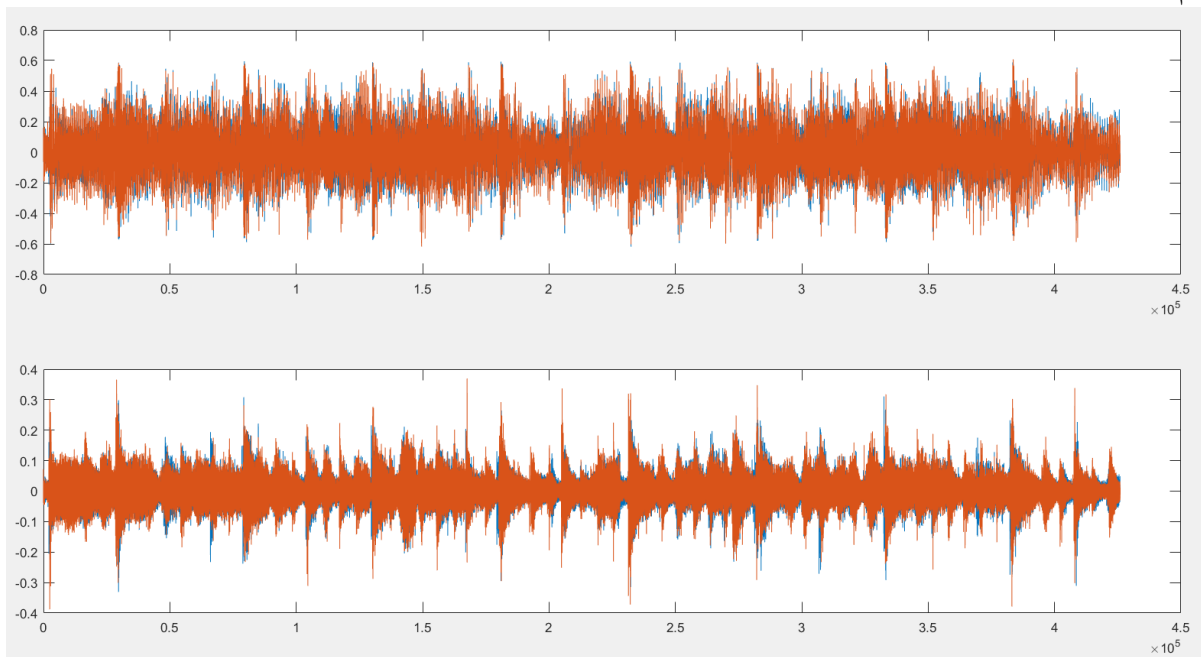
$$\text{out}(n,1)=.9*\text{out}(n-1,1)+\text{mySound}(n,1);$$



یک بار هم برعکس عمل می‌کنیم و فرکانس بالاها را عبور می‌دهیم و فرکانس پایین‌ها را عبور نمی‌دهیم.

`out(n,1)=mySound(n,1)-mySound(n-1,1);`

نام این برنامه p4 است. ورودی آن hootie.wav است و خروجی‌های آن treble4 و high4 می‌باشند.



بخش پنجم

در این بخش می‌خواهیم با تغییر فرکانس نمونه‌برداری سرعت پخش صدا را زیاد و کم می‌کنیم. نام این برنامه، p5 است و ورودی sound1 است. صدا را در برنامه‌ی متلب می‌توانید پخش کنید.

بخش ششم

در این بخش می‌خواهیم بخش ابتدایی صدا را fade کنیم که در واقع وقتی صدای اصلی در حال شروع شدن است، شدت آن به مرور زیاد می‌شود. با مشخص کردن طول fade کردن صدا، باید عناصر آرایه‌ی صدای اصلی را از آغاز تا این طول، همگی را در یک scale ضرب کنیم. این کار در عبارت زیر انجام شده است:

```
sig_faded(1:fade_samples) = mySound(1:fade_samples).*fade_scale;
```

فایل این برنامه p6 نام دارد، ورودی آن hootie است و خروجی آن fade6 می‌باشد. نمودارهای مربوط به آن در شکل زیر قابل مشاهده هستند.

