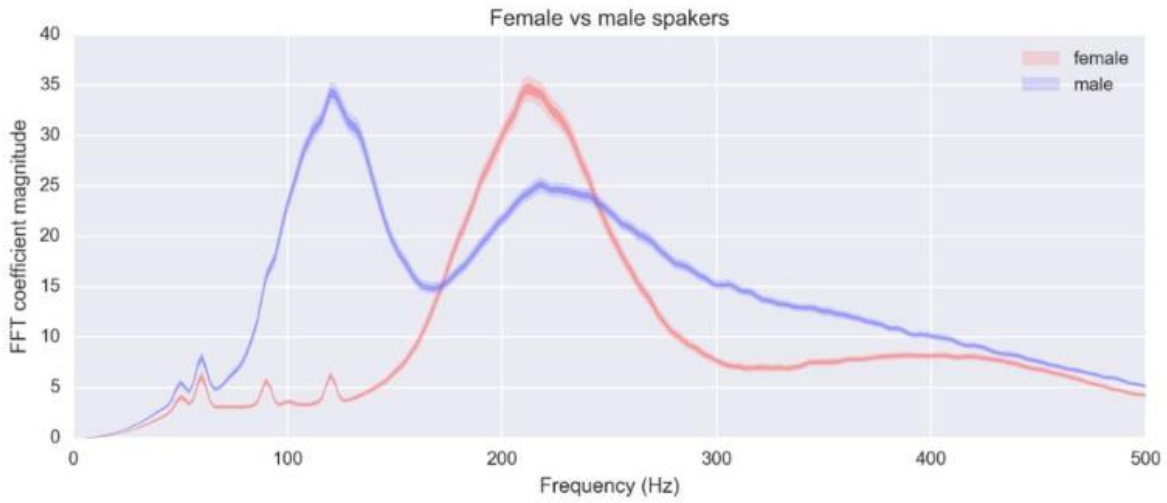
آرش حریرپوش
9731505

بخش اول:

در قسمت اول این پروژه قصد داریم با گرفتن یک صدای ضبط شده تشخیص دهیم که این صدا مربوط به چه جنسیتی است.
گفتار معمولی انسان بین ۵۰و ۳۰۰هرتز متغیر است. بیشتر مردان بین ۸۵ و ۱۸۰هرتز و بیشتر زنان بین ۱۶۵و ۲۵۵هرتز است.
این تفاوت فرکانس در شکل زیر هم قابل مشاهده است.



اوج فرکانسی که در نمودار مشاهده میشود ۱۲۲هرتز برای مردان و ۲۱۲برای زنان است.
مراحل زیر را به ترتیب انجام دهید:

1. در مورد تبدیل فوریه برای تجزیه و تحلیل طیفی(Fourier transform for spectral analysis)مطالعه کنید و دریافت خود را در گزارش بنویسید.

سری فوریه پیوسته یا گسسته زمان، مثل x = x(t) یا $x_n = \{x_1, x_2, \dots\}$، می توان از نظر توضیحات حوزه زمان (time-domain description) و توضیحات حوزه فرکانس (frequency-domain description) مورد تجزیه و تحلیل قرار گیرد. که به آن spectral analysis می گویند و برخی ویژگی های یک سری زمانی (time-series) را نشان می دهد، که به راحتی از تجزیه و توضیح تحلیل حوزه زمان(time-domain description analysis) مشاهده نمی شود. از spectral analysis برای حل کردن طیف گسترده ای از مشکلات کاربردی در مهندسی و علوم استفاده می شود، برای مثال، در مطالعه ارتعاشات، امواج سطحی(interfacial waves) و آنالیز پایداری(stability analysis).

در spectral analysis، سری زمانی به بخش های موج سینوسی (sine wave components) با استفاده از مجموع توابع سینوسی وزن دار به نام spectral components تجزیه می شوند. تابع وزن در تجزیه چگالی spectral components یا چگالی spectral(density) است.

یک روش واقعی برای تجزیه سری زمانی به مجموع توابع سینوسی وزن دار استفاده از تبدیل فوریه است که هر دو نسخه پیوسته و گسسته را به ترتیب متناظر با سری زمانی پیوسته از نوع x = x(t) و سری زمانی گسسته از نوع $x_n = \{x_1, x_2, \dots\}$ دارد. بیشتر دیتا های سری زمانی ثبت شده در تمرین مهندسی از نوع

آرش حریرپوش
9731505

گسسته هستند و محاسبات عددی تبدیل فوریه معمولا با استفاده از کامپیوتر های دیجیتال انجام می شود، که فقط می توانند با دیتا های گسسته سر و کار داشته باشند و بنابراین از تبدیل فوریه گسسته استفاده می کنند.

Power Spectrum

power spectrum برابر است با:

$$powespectrum = |fouriertransform|^2$$

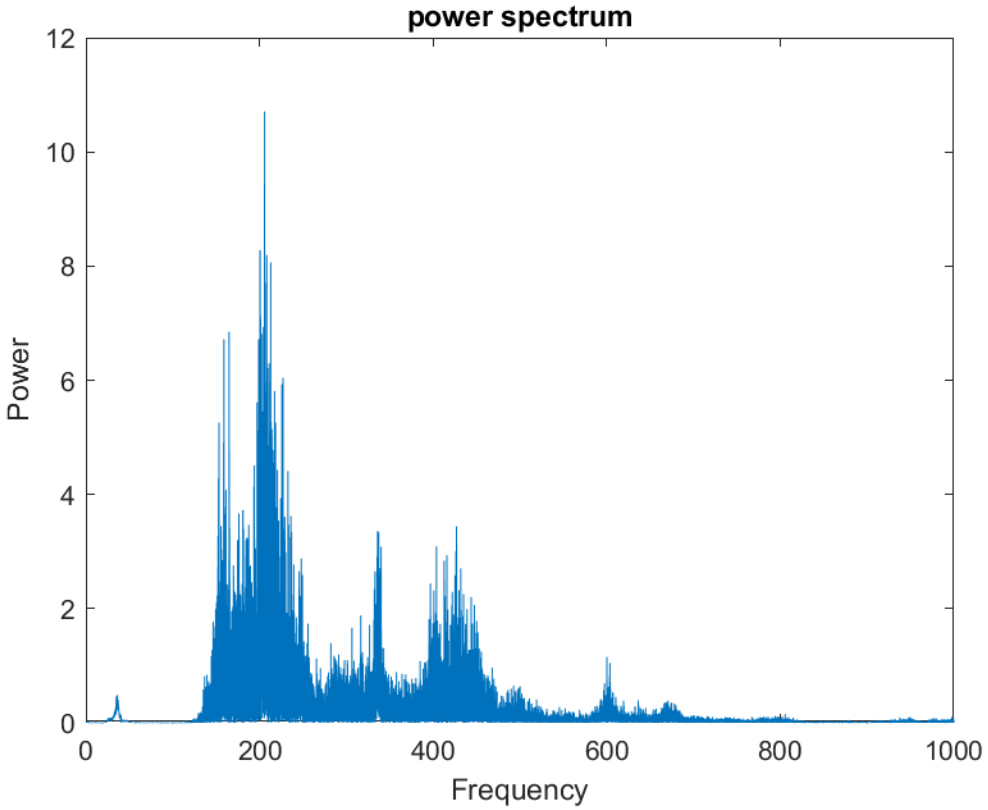
که نشان می دهد powerspectrum یک تابع real-valued با فاز صفر است. power spectrum برابر میانگین ویژگی frequency-domain یک سری زمانی است، که نشان می دهد که آیا یک نوسان متناوب شدید یا نوسان شبه تناوبی (quasi-periodic fluctuation) در سری زمانی وجود دارد یا خیر. (منبع: http://www.thermopedia.com/content/1141)

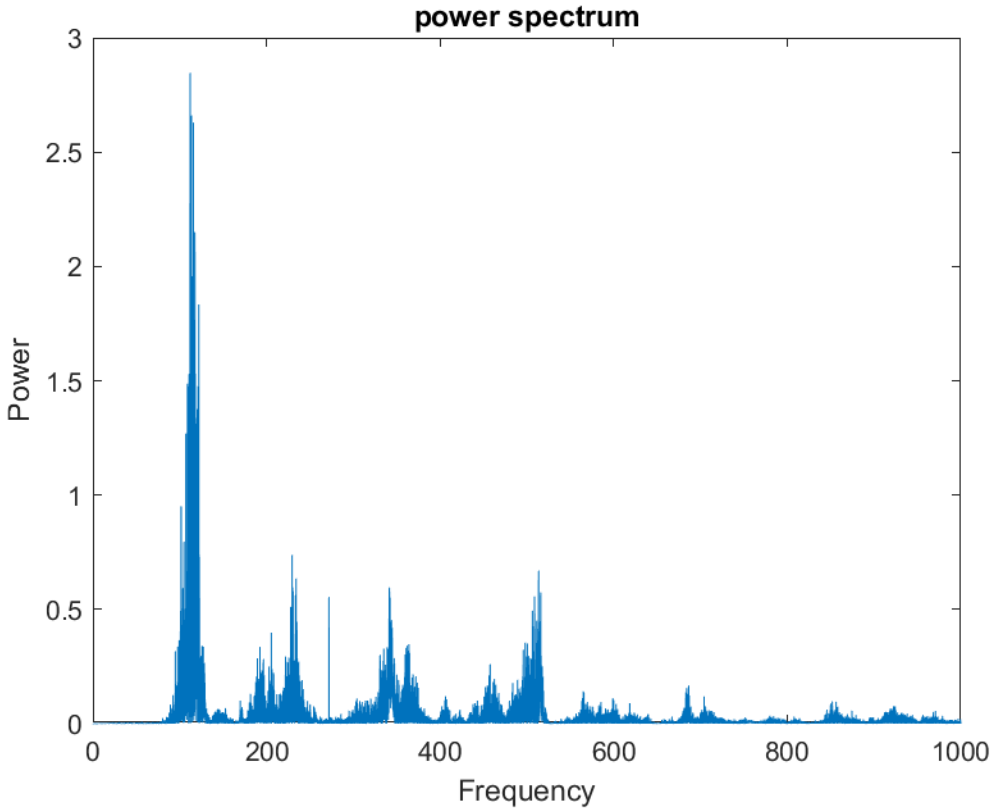
2. کدی بنویسید که بتوانید یک فایل mp3 را باز کنید و نمودار power spectrum آن را رسم کنید. یک صدای مرد و یک صدای زن را از فولدر voices انتخاب کرده و نمودار آنها را رسم کنید و در گزارش خود نمودارها و آنچه دریافتید را بیاورید. در این قسمت از تابع fft متلب کمک بگیرید.

```
function plot_power_spectrum(voice_number)
     a1 =
'D:\semester5\SignalsAndSystems\Project\Bonus_Project\voice
s\v';
     a2 = strcat(a1,int2str(voice_number));
     address = strcat(a2,'.mp3');
     [x,fs] = audioread(address); %read the audio file
     n = length(x);       % original sample length
     y = fft(x, n); % calculate the Fourier transform
     f = linspace(0,1000,fs/2+1);
     power = abs(y).^2/10.^6; %calculate powerspectrum
     plot(f,power(1:length(f)));
     title('power spectrum');
     xlabel('Frequency')
     ylabel('Power')
     h = figure(1);
     file_name = strcat('v',int2str(voice_number),'.png');

saveas(h,fullfile('D:\semester5\SignalsAndSystems\Project\B
onus_Project',file_name));% save the plot result
end
```

نمودار اول صدای یک زن (v0) است و نمودار دوم صدای یک مرد(v1) است.

آرش حریرپوش
9731505





آرش حریرپوش
9731505

همانطور که از نمودار های بالا نیز مشخص است، بیشترین مقدار power برای بیشتر زنان بعد از فرکانس 165 هرتز و برای بیشتر آقایان قبل از فرکانس 165 هرتز است.

3. تابعی بنویسید که با گرفتن آدرس یک فایل mp3 بتواند مقدار اوج فرکانس آن را بدست آورد.

```
function peak = peak_finder(address)
    [x,fs] = audioread(address); %read the audio file
    y = fft(x); % calculate the Fourier transform
    power_specturm_density = abs(y).^2; %calculate the
power spectrum
    f = linspace(0, 1000, fs/2 + 1);
    [~, peak_x] = max(power_specturm_density);
    peak = f(peak_x); %find the peak
    fprintf('%f \n',peak);
end
```

4. تابعی بنویسید که با گرفتن آدرس یک فولدر برای هر یک از فایلهای mp3داخل آن بتواند تشخیص دهد که صدا مربوط به
یک زن است یا یک مرد و سپس یک وکتور خروجی از برچسب زن یا مرد تولید کند. این تابع را بر روی فولدر voices اجرا کنید.
همچنین پیشنهادات خود را برای بهینه کردن این تابع در گزارش بیاورید.

می توان برای بهینه تر شدن این تابع، هنگام محاسبات تبدیل فوریه به کمک تابع `pow2(nextpow2(m))` اندازه سیگنال را به اندازه اولین توان 2 آن افزایش داد که این باعث می شود تابع fft سریع تر بتواند تبدیل فوریه را حساب کند.(هر چند این کار مقدار power را کم می کند و مقدار peak را اندکی زیاد می کند اما تاثیر زیادی در تشخیص صدا نمی گذارد).

آرش حریرپوش
9731505

```
function genderDetection(folder_address)
    voice_folder = dir(fullfile(folder_address,'*.mp3')); % 
The folder in which the voice files has been saved
    fileID = 
fopen('D:\semester5\SignalsAndSystems\Project\Bonus_Project
\gender_label.txt','w');
     for i = 1:length(voice_folder)
        peak = 
peak_finder(fullfile(folder_address,voice_folder(i).name));
%find the peak
        fileAddress = 
strsplit(fullfile(folder_address,voice_folder(i).name),
'\');
        full_file_name = strsplit(fileAddress{7},'.');
        file_name = full_file_name{1};% find the name of
the file
        fprintf('%s \n', file_name);
        if (peak <= 165)
            label = strcat(file_name, ' male'); % detect
the voice as male voice
        elseif(peak >= 165)
            label = strcat(file_name, ' female'); % detect
the voice as female voice
        else
            label = strcat(file_name, ' not detected');
        end
        fprintf(fileID, ' %s \n',label); % write the result
in the file
    end
end
```

آرش حریرپوش
9731505

بخش دوم:

برای قسمت دوم این پروژه قصد داریم تا به پیادهسازی الگوریتم Spectral Subtraction بپردازیم. این الگوریتم یک روش ساده برای speech enhancement است که به کمک آن میتوان نویز صدا را حذف کرد.

1. در مورد الگوریتم Spectral Subtraction تحقیق کنید و بطور خلاصه در گزارش خود بیاورید.

در این الگوریتم، یک فایل speech دارای نویز را دریافت می کند و سعی می کند تا نویز آن را حذف کند تا صدای سخنرانی بهتر شنیده شود، در حالی که هیچ بخشی از سخنرانی حذف نشود. اصل اساسی آن به این صورت است که: اگر ما یک نویز اضافی در نظر بگیریم، ما می توانیم spectrum آن نویز را از spectrum صدای نویز دار کم کنیم، بنابراین چیزی که باقی می ماند باید شبیه spectrum یک صدای سخنرانی واضح(تمیز) باشد. برای این کار نیاز است تا بدانیم که spectrum نویز چگونه است، بنابراین آن را از ناحیه هایی که هیچ سخنرانی نیست (ناحیه ای که هیچ کس صحبت نمی کند) تخمین می زنیم(بخشی از سیگنال که فقط شامل نویز است) و سپس در نظر می گیریم که در بخش های بعدی خیلی تغییر نمی کند.

2. یک نوع از این الگوریتم را انتخاب کنید، پیاده سازی کنید و در گزارش خود توضیح دهید.

ابتدا بخشی از سیگنال که در آن فقط نویز وجود دارد و کسی صحبت نمی کند را انتخاب می کنیم، سپس مقدار میانگین amplitude، spectrum آن بخش را محاسبه می کنیم، سپس آن را از spectrum سیگنال دارای نویز کم می کنیم و برای بخش هایی که مقدار آن تفریق کمتر از صفر می شود یک مقدار کم مثلا 0 یا 0.5 می گذاریم سپس با تبدیل فوریه معکوس آن را به حوزه زمان می بریم و ذخیره می کنیم.

حال برای بهتر شدن عملکرد دو متغیر omega و gamma در نظر می گیریم که متغیر اولی در نویز ضرب می شود و spectrum سیگنال حاوی نویز و نویز به توان متغیر دومی می رسد؛ با تغییر این مقادیر برای هر نویز می توان تصمیم که صدای سخنران بیشتر باشد که نویز هم بیشتر می شود یا صدای سخنران کمتر باشد و نویز هم کمتر بشود. با اندازه gamma برابر 1 ما اندازه spectral subtraction را داریم، ولی با gamma برابر 2 ما power spectral subtraction را داریم. همچنین می توانیم مقادیر 0.8 یا 3 را هم برای gamma داشته باشیم. هر کدام از این مقادیر ویژگی های متفاوت اندکی دارند، بدین صورت که حذف نویز در مقابل حذف اطلاعات سخنرانی قرار می گیرد. اگر احساس کنیم که به اندازه کافی نویز حذف نشده است می توانیم با افزایش مقدار omega از 1 به 1.5 آن را کمتر کنیم. این اتفاق باعث می شود که نویز بیشتری حذف شود، اما بخشی از اطلاعات سخنرانی نیز ممکن است حذف شود؛ در نتیجه با افزایش مقدار omega نویز بیشتری حذف می شود، و با کم کردن مقدار gamma نویز بیشتری حذف می شود اما صدا نیز ضعیف تر می شود و با افزایش مقدار gamma صدای بیشتری شنیده می شود اما نویز هم کمتر حذف می شود.

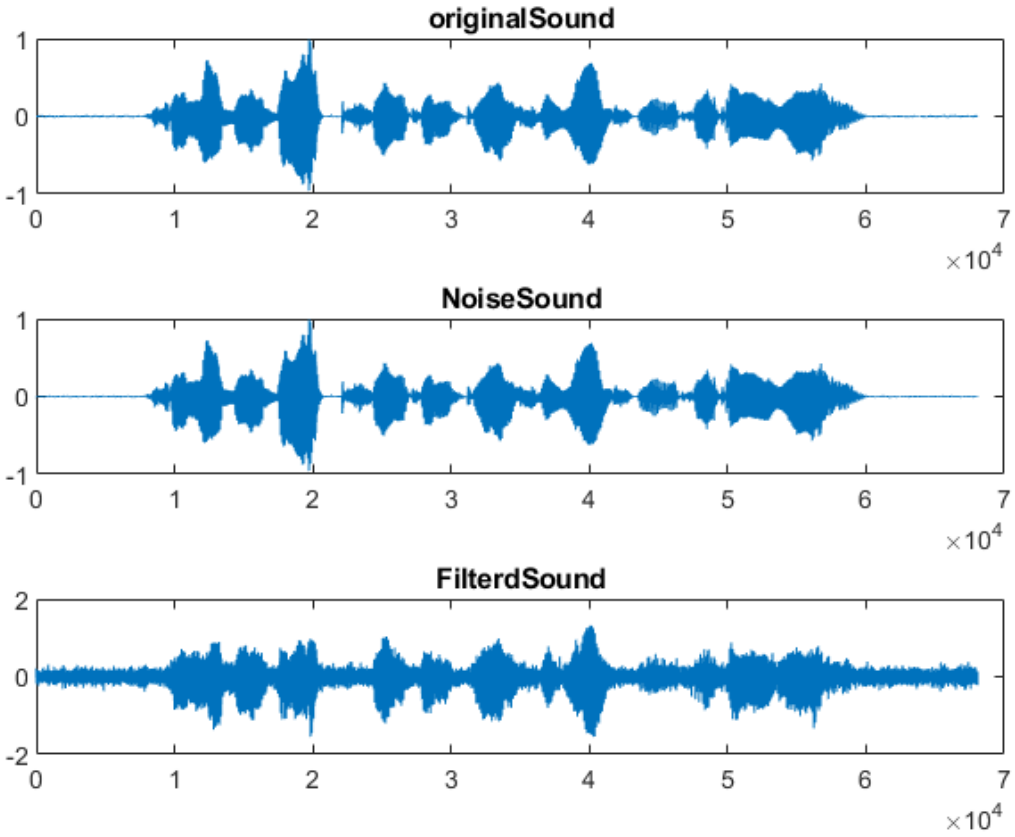
3. فایل با نام Test.wavرا در متلب باز کنید، با تابع awgn متلب به آن نویز اضافه کرده و کد خود را در مقابل قدرتهای متفاوت نویز بسنجید. نتایج خود را گزارش کنید.

آرش حریرپوش
9731505

```
function SpectralSubtraction(awgn_rate)
[y,fs] =
audioread('D:\semester5\SignalsAndSystems\Project\Bonus_Pro
ject\Test.wav');
%% Add noise
S = RandStream('mt19937ar','Seed',5489);
Noisesignal = awgn(y,awgn_rate,0,S); %noise2
%Noisesignal = awgn(y,awgn_rate,'measured'); %noise1

%% spectral substraction
gamma = 0.25;
omega = 0.75;
% signal spectrum
spectrum_noisy_signal = fft(Noisesignal);
spectrum_noisy_signal_mag = abs(spectrum_noisy_signal);
phase_signal = angle(spectrum_noisy_signal);
%findNoise
noise_r = spectrum_noisy_signal_mag(1:8218);
Noise = mean(omega.*noise_r);
fprintf("%f\n", abs(Noise));
% subtract spectrum
subtracted_spectrum_mag = (spectrum_noisy_signal_mag.^gamma
- omega.*Noise.^gamma).^(1/gamma);
subtracted_spectrum_mag(subtracted_spectrum_mag < 0) =
max(subtracted_spectrum_mag(subtracted_spectrum_mag < 0),
0.5);

 %recover original spectrum and signal
enhanced_spectrum = awgn_rate.*subtracted_spectrum_mag .*
exp(1i*phase_signal);
original_signal = ifft(enhanced_spectrum);

figure;
 subplot(3,1,3);
 plot(real(original_signal));
 title("FilterdSound");
 subplot(3,1,2);
 plot(Noisesignal);
 title("NoiseSound");
 subplot(3,1,1);
 plot(y);
  title("originalSound");

sound(real(Noisesignal), fs);
```

```
pause(4);
sound(real(original_signal), fs);
filename =
'D:\semester5\SignalsAndSystems\Project\Bonus_Project\WT.wa
v';
audiowrite(filename,real(original_signal), fs);
end
```

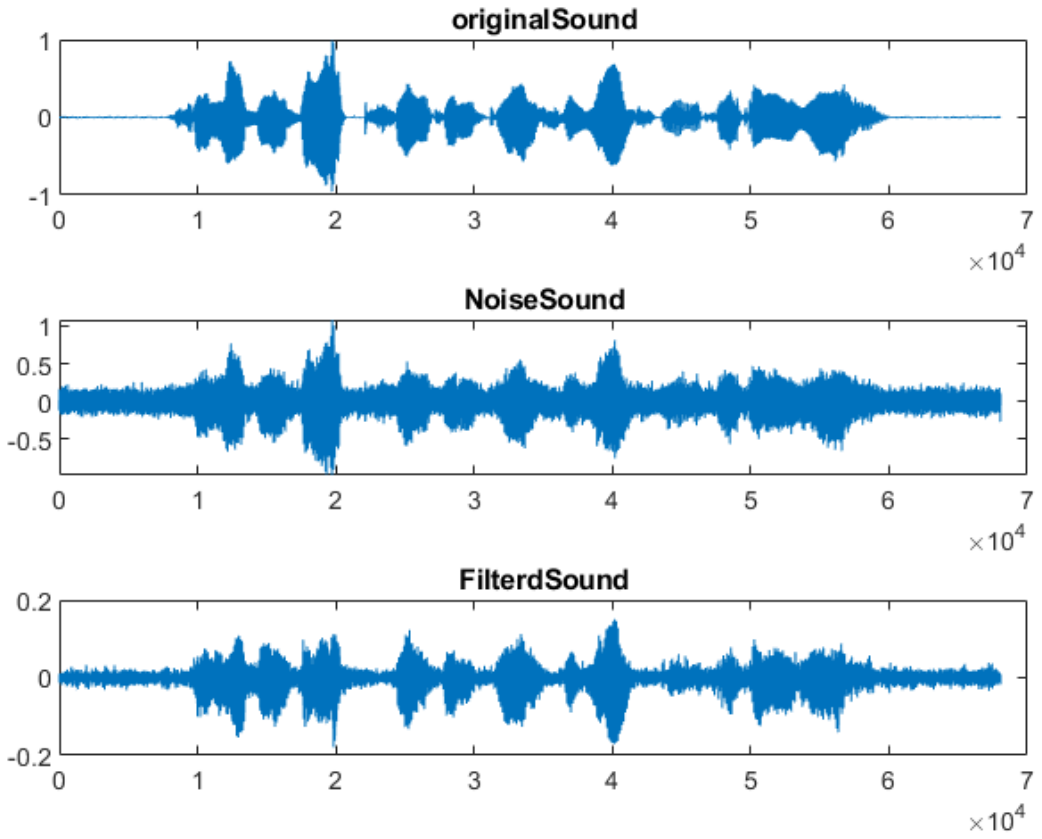
ابتدا کد خود را با noise1 و شدت های نویز متفاوت بررسی می کنیم؛ این نویز یک نویز سفید Gaussian به سیگنال اضافه می کند؛ ابتدا از شدت کم مقدار نویز شروع می کنیم، مقدار نویز را برابر dB 40 می گذاریم:



همانطور که از نمودار های بالا نیز مشخص است، برای مقادیر کم نویز، این الگوریتم نویز را کاهش می دهد اما صدا نیز نسبت به قبل اندکی ضعیف تر می شود. برای مقدار نویز 30dB هم این نتیجه مشاهده شد، با این تفاوت که اندکی نویز باقی می ماند و صدا هم بهتر از حالت قبل می شود، همین صورت با افزایش میزان نویز صدای خروجی نویزش کمتر می شود، اما صدای سخنران نیز ضعیف تر می شود. در

آرش حریرپوش
9731505

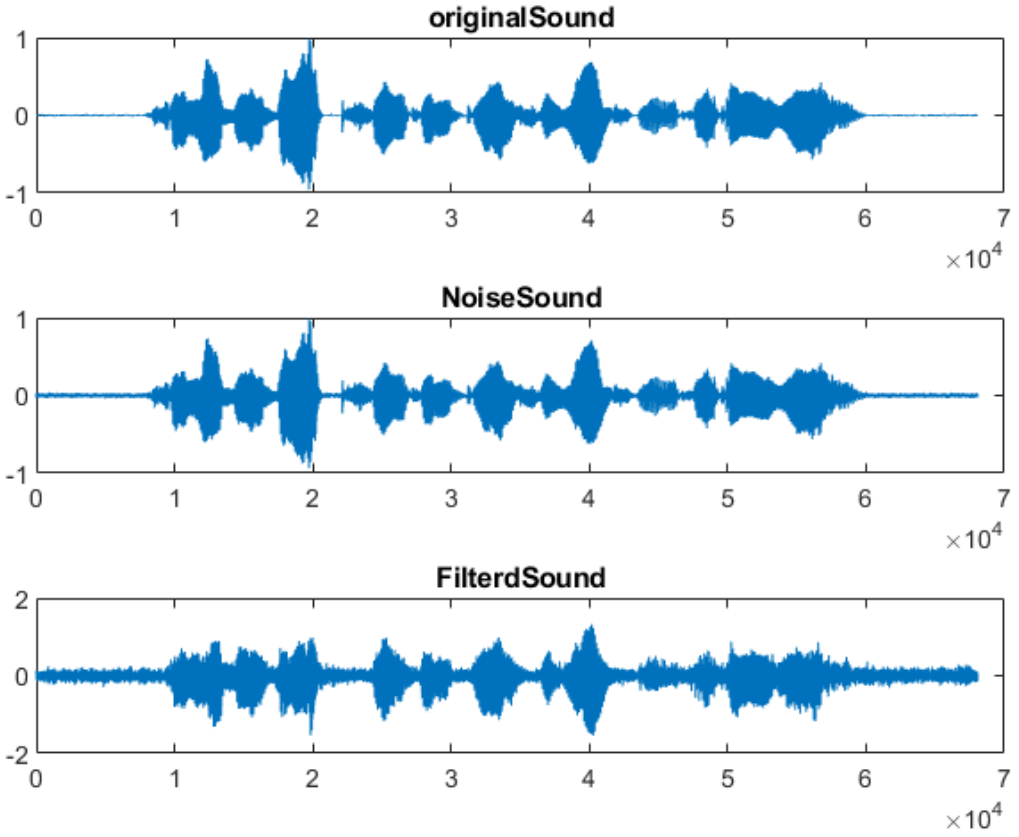
ادامه نمودار های مرتبط با نویز 5dB قرار داده شده است.



می دانیم که با تغییر مقادیر omega و gamma میزان نویز و صدای باقی مانده را تغییر داد، و این نتایج برای مقادیر ثابت omega = 0.75 و gamma = 0.26 بدست آمده است.

حال همین آزمایش را بر روی noise2 نیز انجام می دهیم، که یک نویز سفید Gaussian را با استفاده از randstream تولید می کند.(این نویز از نویز قبلی شدید تر است) همانند آزمایش قبل از مقادیر کم نویز

مثل 40dB شروع می کنیم و مقادیر آن را زیاد می کنیم(با کم کردن 40dB مقدار نویز بیشتر می شود).



همانند آزمایش قبل، در نویز های کم اندکی صدای پس زمینه به صدا اضافه می شود که باعص می شود کیفیت صدای سخنران کاهش یابد هر چند نویز کم شود، با افزایش مقدار نویز این تابع بهتر عمل می کند و نویز بیشتری را کاهش می دهد، هر چند صدای سخنران ضعیف تر می شود، در مقادیر نویز زیاد صدای نویز کمتر می شود و صدای سخنران نیز ضعیف تر می شود. در ادامه نمودار های مربوط به نویز 20dB قرار داده شده است.

آرش حریرپوش
9731505