Ecole Polytechnique de Tunisie

TSS Project report

# Denoising audio signals

*Supervised by*
Mrs.Ines BOUSNINA

*Elaborated by :*
Ghanem BAHRINI
Hamdi BARKOUS

2021/2022

# Table des matières

# 1 Introduction :

Based on the fact that noise and distortion are the main factors that limit the capacity of data transmission in telecommunications and that they also affect the accuracy of the results in the signal measurement systems, whereas, modeling and removing noise and distortions are at the core of theoretical and practical considerations in communications and signal processing. Another important issue here is that, noise reduction and distortion removal are major problems in applications such as ; speech recognition, image processing, medical signal processing, sonar and so on ..
There are many types and sources of noise or distortions and they include :Electronic noise, Acoustic noise and Electromagnetic noise which can be classified into several categories :White noise, Narrow band noise, Colored noise, Impulsive noise and Transient noise pulses.
In this project we will only focus on additive white noise and the different methods to remove it from audio signals.

# 2 Objective of the work

During the transmission, storage process mostly occurred noise in audio. De-noising techniques are applied to reduce unwanted signal form audio. In this project, MATLAB is used to perform the algorithms of filters. In this project filters are used to minimize the unwanted noise. To reduce the noise from audio is difficult task, so, Butterworth filter and wiener filters are used with low-pass, high-pass, Band-reject.
We present an overview to reduce the noise

# 3 White noise

In signal processing, white noise is a random signal having equal intensity at different frequencies, giving it a constant power spectral density.
The term is used, with this or similar meanings, in many scientific and technical disciplines, including physics, acoustical engineering, telecommunications, and statistical forecasting.
White noise refers to a statistical model for signals and signal sources, rather than to any specific signal. White noise draws its name from white light, although light that appears white generally does not have a flat power spectral density

over the visible band.
In discrete time, white noise is a discrete signal whose samples are regarded as a sequence of serially uncorrelated random variables with zero mean and finite variance, a single realization of white noise is a random shock. Depending on the context, one may also require that the samples be independent and have identical probability distribution (in other words independent and identically distributed random variables are the simplest representation of white noise).
In particular, if each sample has a normal distribution with zero mean, the signal is said to be additive white Gaussian noise.

Let $X = (X_k)_{k\in\mathbb{N}}$ be a discrete random process. $X$ X is Gaussian white noise iff

- $X_k$ are independent

$-X_k \sim \mathcal{N}\left(0, \sigma_X^2\right)$

**Proprieties**

- The auto-correlation function is a dirac pulse :

$$R_X(t) = \delta(t)$$

- The PSD of $X$ is constant :

$$S_x(f) = \sigma_X^2$$

- Let $h$ be a filter, such that :

$$Y = h \star X$$

  then

$$S_Y(f) = |h(f)|^2 S_X(f)$$

For the rest, we will consider the following problem :

$$Y = X + n$$

where
- n is a white noise with unknown PSD
- X is the original signal that we are looking for

# 4 Denoising methods :

## 4.1 Classical methods

### 4.1.1 spectral subtraction

The principle of denoising by spectral subtraction, consists in eliminating the noise in the spectral domain. Indeed, when a signal is degraded by an additive noise $n(t)$, the noisy signal is obtained according to the expression :

$$y(k) = x(k) + n(k)$$

In the frequency domain, the previous equation implies that :

$$Y(f) = X(f) + N(f)$$

where $X(f)$ and $N(f)$ are respectively the Fourier transforms of the signal and the noise. It is necessary to estimate the frequency amplitude of the noise.
Once this estimate is known, all that remains is to subtract the value obtained from the signal :

$$\widehat{X}(f) = Y(f) - \hat{N}(f)$$

Some use the PSD instead of the amplitude, and in this case we will express the energy of the noisy signal according to the equation :

$$|Y(f)|^2 = |X(f)|^2 + |N(f)|^2 + X(f)N^*(f) + X^*(f)N(f)$$

If we suppose that the noise and the signal are independent then the previous equation can be written as :

$$|Y(f)|^2 = |X(f)|^2 + |N(f)|^2$$

hence :

$$|\hat{X}(f)|^2 = |Y(f)|^2 - |\hat{N}(f)|^2$$

For the reconstruction of the denoised signal in the time domain, it is necessary to have the phase of the original signal. The latter is obtained by extracting the phase of the noisy signal itself, which is used with the inverse Fourier transform on the estimate of the frequency amplitude to recover the noisy signal.

**MATLAB implementation :**

we will consider a noisy signal defined as following :

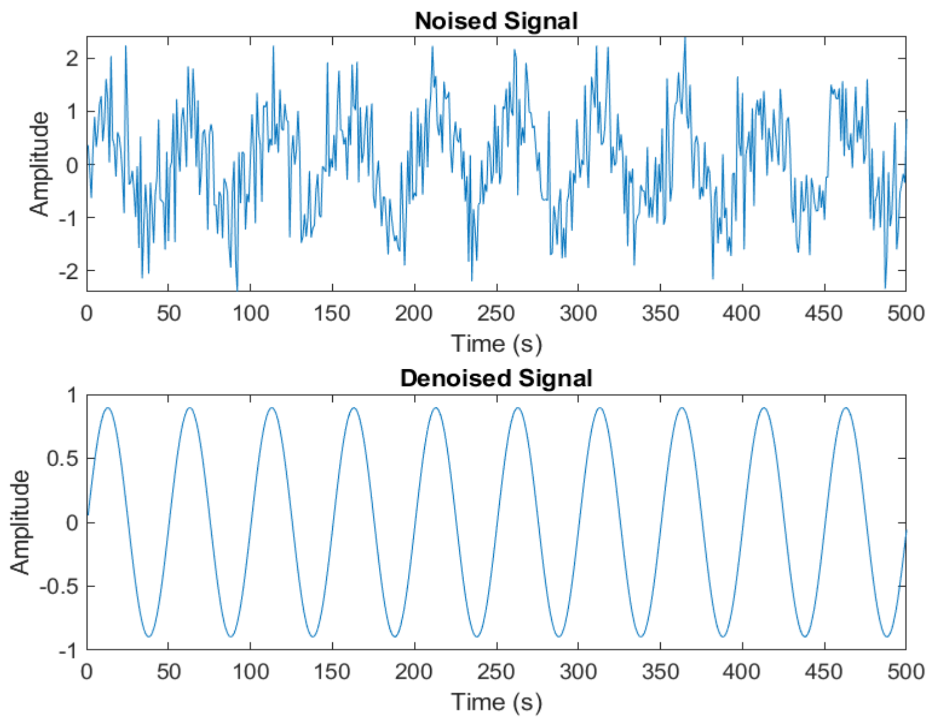
$$X(t) = \sin(2\pi f t) + N(t)$$

where

$$f = 1000Hz$$

and $N$ is a white noise signal with $\sigma_X^2 = 0.7$

**Results**



### 4.1.2 Wiener filter

Some assumption should be made for the efficient design of the Wiener filter. First, the noise-affected input audio signal is single-channel (from one source), and the noise and audio signals are uncorrelated. If the noise affecting the speech signal is additive, then

$$\boldsymbol{y}(\boldsymbol{K}) = \boldsymbol{x}(\boldsymbol{k}) + \boldsymbol{n}(\boldsymbol{k})$$

where

By merely observing the time domain samples, detecting noise or interference in the input signal is very difficult. The analysis and detection of such signals become easy by mapping the signals in frequency domain. Thus, we need to convert the time domain noise-affected signal to the frequency domain. By taking RFFT, we get

$$\boldsymbol{Y}(\boldsymbol{f}) = \boldsymbol{X}(\boldsymbol{f}) + \boldsymbol{N}(\boldsymbol{f})$$

The original speech signal $\boldsymbol{X}(\boldsymbol{f})$ can be extracted from the noisy signal by multiplying the noisy speech signal $\boldsymbol{Y}(\boldsymbol{f})$ with the Wiener filter function $W(f)$.

$$\boldsymbol{X}(\boldsymbol{f}) = \boldsymbol{W}(\boldsymbol{f})\boldsymbol{Y}(\boldsymbol{f})$$

In Eq. (7), W(f) represent the Wiener filter in frequency domain and can be estimated as

$$\boldsymbol{W}(\boldsymbol{f}) = \frac{|\boldsymbol{X}(\boldsymbol{f})|^2}{|\boldsymbol{X}(\boldsymbol{f})|^2 + |\boldsymbol{N}(\boldsymbol{f})|^2}$$

where $|\boldsymbol{X}(\boldsymbol{f})|^2$ is the PSD of the original speech signal; $|\boldsymbol{N}(\boldsymbol{f})|^2$ is the PSD of the noise speech signal. In reality, we have no idea about the original speech and noise spectra, as the input to the device is noisy speech signal and noisy signal. The Wiener filter is approximated from the noisy speech signal and the estimated noisy signal as

$$\boldsymbol{W}(\boldsymbol{f}) \approx \frac{|\boldsymbol{Y}(\boldsymbol{f})|^2 - |\boldsymbol{N}(\boldsymbol{f})|^2}{|\boldsymbol{Y}(\boldsymbol{f})|^2}$$

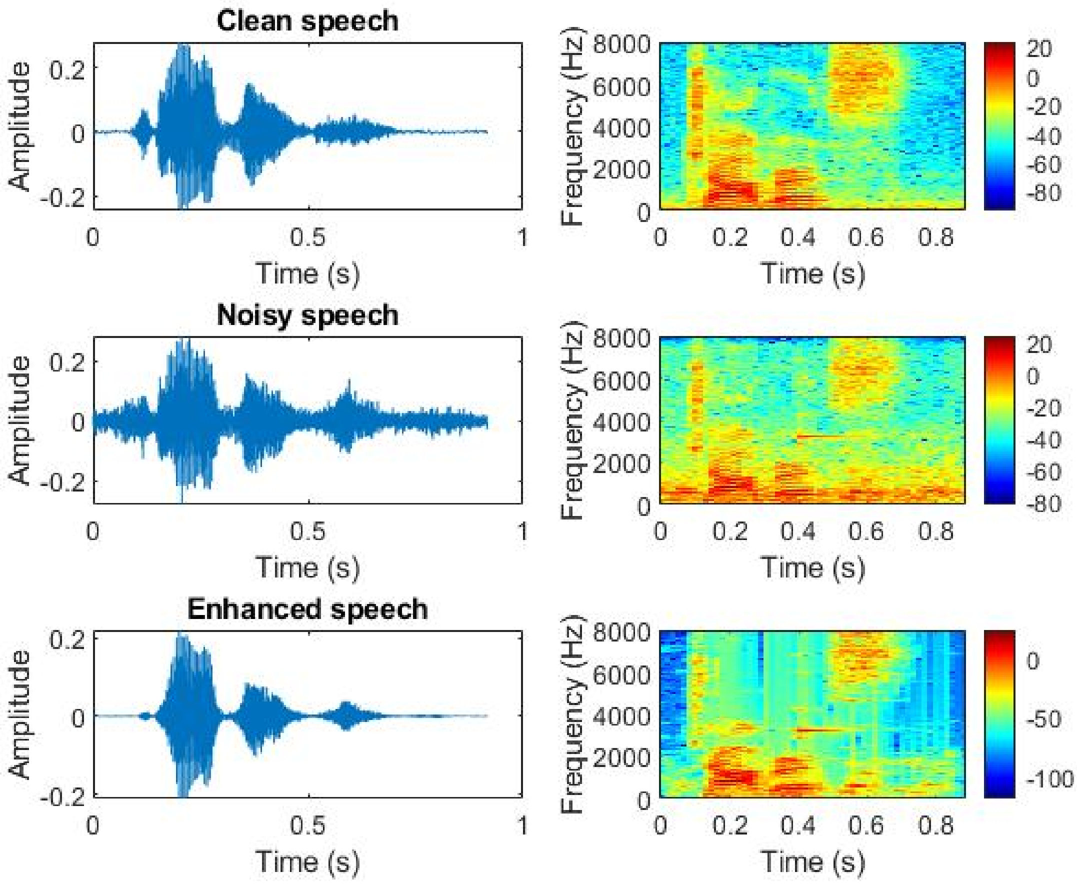
where $|\boldsymbol{Y}(\boldsymbol{f})|^2$ is the PSD of input noisy speech signal, $|\boldsymbol{N}(\boldsymbol{f})|^2$ is the PSD of input noisy signal.
Taking IFFT, we get the original signal $\boldsymbol{s}(\boldsymbol{t})$

$$\boldsymbol{s}(\boldsymbol{t}) = \boldsymbol{RIFFT}\{\boldsymbol{S}(\boldsymbol{f})\}$$

**MATLAB Implementation :**

for this filter we will consider a clean audio file "jarvus.wav" and its noisy audio file "jarvus-pub.wav". we will you apply our Wiener filter on the noisy audio and then we will compare the output with the original audio.

**Results**

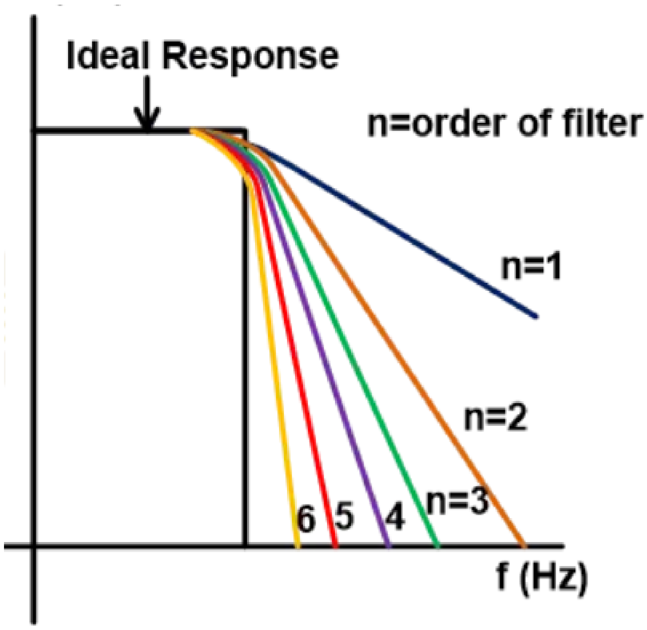



**Interpretation :**

As we can see, our filter eliminated the noise from the noisy audio by eliminating low frequencies at $[0, 0.1s]$ and low frequencies at $[0.6s, 0.8s]$

## 4.2 Butter-worth filter

A Butterworth filter is a type of signal processing filter designed to have a frequency response as flat as possible in the passband.
Hence the Butterworth filter is also known as maximally flat magnitude filter. The frequency response of the Butterworth filter is flat in the passband (i.e. a bandpass filter) and roll-offs towards zero in the stopband. The rate of roll-off response depends on the order of the filter.
The below figure shows the frequency response of the Butterworth filter for various orders of the filter.




The generalized form of frequency response for nth-order Butterworth low-pass filter is;

$$H(j\omega) = \frac{1}{\sqrt{1 + \varepsilon^2 \left(\frac{\omega}{\omega_C}\right)^{2n}}}$$

Where,

n = order of the filter, $\omega$ = operating frequency (passband frequency) of circuit $\omega_C$ = Cut-off frequency $\varepsilon$ = maximum passband gain = $A_{max}$

The below equation is used to find the value of $\varepsilon$.

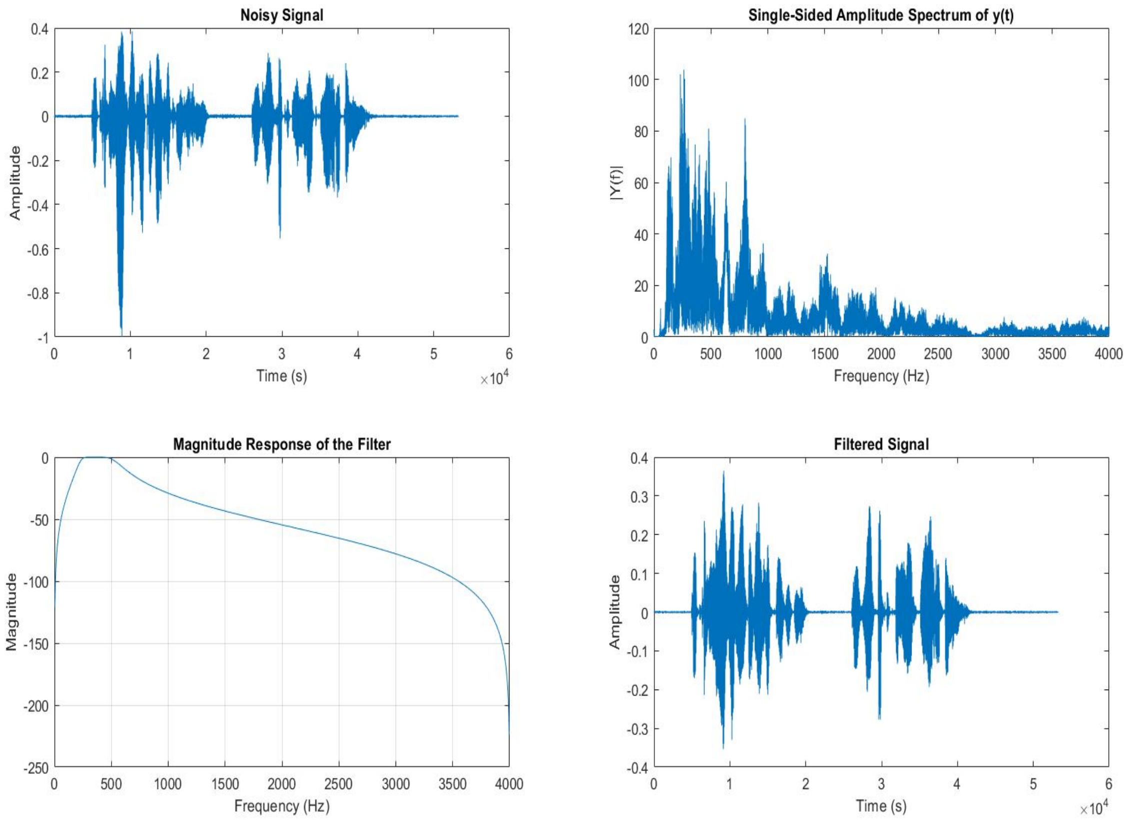
$$H_1 = \frac{H_0}{\sqrt{1+\varepsilon^2}}$$

Where,
$H_1$ = minimum passband gain
$H_0$ = maximum passband gain

**MATLAB implementation**

for this filter we will consider a noisy audio file "parole1.mp3". we will you apply our Butter-worth filter on the noisy audio and then we will compare the output with the original audio

**Plots :**



# 5 Conclusion

In this project, different audio signal denoising techniques based on the methods of spectral subtraction, adaptive Wiener filtering and butterworth filtering are presented.
In conclusion, we must confess that in retrospect we are satisfied with this work since we have reached new objectives.
Indeed, this mini project allowed us to understand and learn to master the theory of signals and systems course using matlab software