The development of a Speech Level Adjustment Technique for late Deaf People

Karim Gabriel Sani AlMoudarres
Supervisor: Nedelko Gbrc
Authors:
Karim Gabriel
Address: Folkparksvägen 15:18, 37240 Ronneby, Sweden
E-mail: karim.gabriel@gmail.com

Sani AlMoudarres
Address: Folkparksvägen 15:18, 37240 Ronneby, Sweden
E-mail: saal05@student.bth.se

Department of
Telecommunication and Signal Processing
Blekinge Institute of Technology
Box 520
SE 372 25 Ronneby
Sweden

Internet : www.bth.se/tek
Phone : +46 457 38 50 00
Fax : + 46 457102 45
## Contents

1 Introduction 2

2 Social Study 4

3 Theoretical Background 6
3.1 Characteristics of speech and noise ........................................ 6
3.2 Signal to Noise Ratio .......................................................... 7
3.3 Stochastic Signals and Analysis .............................................. 7
3.4 Probability Density Function(PDF) ......................................... 7
3.5 Expectation, Mean, and Variance ........................................... 8
3.6 Correlation and Power Spectral Density ................................. 9
3.7 Discrete Fourier Transform ................................................. 10

4 Study of Noise Estimation Techniques 11
4.1 An efficient algorithm to trace the noise floor ....................... 11
4.1.1 Abstract ........................................................................... 11
4.1.2 Introduction ...................................................................... 11
4.1.3 Description of algorithm ................................................. 11
4.1.4 Noise power estimation .................................................. 12
4.2 Noise estimation by using speech pause detection .................. 13
4.2.1 Introduction ...................................................................... 13
4.2.2 Algorithm ........................................................................ 13

5 Design and implementation 17
5.1 Offline approach ................................................................. 17
5.1.1 Introduction ...................................................................... 17
5.1.2 Testing methods ............................................................. 18
5.1.3 Speech pause detection implementation and tests ............. 18
5.1.4 Post Processing ............................................................... 21
5.1.5 Output signal to the deaf person ..................................... 23
5.2 Online approach ................................................................. 26
5.2.1 Introduction ...................................................................... 26
5.2.2 Testing Method .............................................................. 27
5.2.3 Speech pause detection implementation and tests .......... 27
5.2.4 Post processing .............................................................. 29
5.2.5 Output signal to deaf person ....................................... 30

6 Conclusion 35
6.1 Suggested Prototype ........................................................... 35
6.2 Future Work ......................................................................... 36

Bibliography 37
1 Introduction

Speech is one of the oldest forms to communicate among humans. In life we can distinguish two types of Deaf, the first type is Deaf from birth and the other type is a deaf person who lost his hearing ability later in life. Those deaf persons are our target group; they know how to speak but cannot hear themselves.

Our aim is to design a prototype to help the late deafened persons to adjust their level of speech to the surrounding taking into account the ambient noise\(^1\). In this way it is possible to avoid the embarrassment that a deaf person could face in everyday life such as, people giving hand gestures signaling to the deaf person to lower his/her voice. Some people facing this problem developed lip/face reading techniques so they understand when to raise or lower their voice but, they are still having difficulty to estimate the right level of speech according to the people around them.

Adjusting the speech level can be done either by measuring the noise in the environment and add to it a certain level on which their speech can be heard properly, these different levels of added speech are listed in a referential table. Another way of doing the adjusting is to compare their voice with other speakers in the surrounding and try to inform the deaf person to speak in a vicinity of 10 dB in relation to other speakers.

To estimate the noise in the surrounding area we focused on two methods, first using VAD (voice activity detector), another way would be using spectral estimation for the speech spectrum and considering that the portions of spectrum which have the minimum spectral characteristics are the background noise.

The proposed prototype solution aims to help the late deafened people by generating pulses "buzzing effect" through the hand or the neck which tell the deaf person whether to raise or lower his/her voice. It depends on the intensity and frequency of the generated pulses, for instance if they are very frequent (more frequent than a standard constant level, i.e. constant level represents the ambient noise) it tells the deaf person to raise his/her voice, and if the intensity of the buzzer is stronger than a threshold it means that he/she is far away from the desired accepted level.

In chapter 2 we conducted a social study to define the objectives and scope of the project. In chapter 3 we examine the basic concepts used throughout the thesis. In chapter 4 we study comprehensively noise estimation techniques. In chapter 5 the work was divided into two main parts. First, a simulation study of the proposed solution using computing tool Matlab which was named the offline approach. We have tested many algorithms of noise cancellation on Audio files mixed with different types of noise and create an audible output. Second, a real-time implementation on DSP with a single microphone to acquire

\(^1\)like a person with a normal hearing ability adapt to the ambient noise spontaneously without paying attention to it
the input signal. Afterward, it will be processed and then, an output signal will be formed and fed to the user. This signal indicates the deaf person whether to increase or decrease his/her voice. Chapter 6 concludes the paper and contains suggested future work.
2 Social Study

A social study was done to understand the needs and desires of the late deafened persons which will help us to visualize how our device would look like and what it need to perform in the expected prototype. In this prospect we will try to maximize the efficiency of our device and provide the utmost comfort for the deaf person.

The social study was necessary to know what are the needs of the target group (The late Deafened People) and listen to their feedback and perspectives. Therefore, we decided to contact the ALDA (association for the Late Deafened Adults), actually our work concerns the late deafened people who learned the language and later on in their life lost the capability of hearing, but as well the not completely deaf who are facing big hearing loss therefore not able to speak at the right loudness comparing to the surrounding environment. Since the late deafened person no longer receive an auditory feedback from the people who he/she is talking to, which create a vexing situation for the deaf and make him lower his/her voice, sometimes till the point where he/she is not heard anymore. Some members of the LDA (the late deafened American association) said that they replace auditory feedback with lip-reading or interpret facial reactions and expression, so they know whether to raise their voice or lower it down, without any help from an external device. However, they still think that our device is needed even though they have developed lip-reading or other techniques to overcome the problem of not having an auditory feedback. As we might as well assume it’s embarrassing to rely on people to know the right loudness on which they should tune in, also it takes time to do so to the desired accuracy, add to it there might be unseen events which create lots of noise such as rock concerts, construction works . . . etc. For which people unconsciously start to raise their voice and adjust to the surrounding conditions and the deaf person cannot.

Also we consulted the mentioned group for insights on how the device should look like and they proposed a device using vibra-tactile elements to enhance the perception of the ambient sound level.

We think it’s applicable, because you can tell the deaf person to increase or decrease his/her voice through a tactile vibration either by using a vibrator like the one used in mobile phone installed on the wrist of the hand, or like a necklace on the neck. From the information gathered until now, we decided to realize the prototype as follows: it vibrate steadily if there is only static noise around, but when there are speakers we calculate the desired speech level according to our algorithm and then produce the vibration. If the deaf is speaking very low then the vibration is faster and stronger than the steady vibration, fast pulses indicate that desired speech level is higher than what he/she speaks at the
moment, and the strong indicate how much different from the desired level of speech, i.e. the stronger it is the louder the deaf person should speak. Lastly, and most important we should take into consideration not to overload the deafened person who already have a hearing aid sometimes two with noise reduction algorithms in the newest version of FM assisting listening device. Thus, they don’t want more devices added to the collection, they suggest that our device should be included in the hearing aid. Moreover, it shouldn’t be so complicated and hard to operate otherwise it wouldn’t be efficient and friendly for the user.
3 Theoretical Background

This chapter suppose to give an overview of the theoretical basis that we used in our thesis to achieve the task of comparing the speech of the deaf person to the ambient noise and to other speakers voice level.

In order to do that we had to investigate various topics such as speech representation, noise characteristics, methods of noise reduction, VAD (voice activity detection), Spectrum analysis and spectral subtraction techniques, and finally and most importantly to be able to estimate how much we need to increase the deaf person speech pitch according to the other speakers (in dB) and the noise in the surrounding area.

In the following sections we will give a comprehensive explanation of the topics mentioned above and the theory behind our work in the thesis.

3.1 Characteristics of speech and noise

In telephony the frequencies lies in a band between 300 Hz - 3400 Hz, so if our purpose is separating noise form the signal, we can do that easily if the noise lies outside the mentioned band, we pass the signal through band pass filter, but the process of filtering out the noise become much harder if we have noise with specific tonality, which is usually concentrated in narrow bands. This type of noise can be filtered out using comb filtering. If the bands are very narrow then its effect on speech will be negligible. However, it’s not always the case, many times noise has a broadband spectral characteristics which makes filtering them out more difficult, these components can voiced elements like vowels caused by periodic vibration of the vocal chords, or unvoiced like whispering or turbulent airflow traveling in pipes. Because of the random nature of these components, we have to employ new methods such as statistical analysis for both speech and noise signals. We can use famous statistical functions such as the autocorrelation function, power spectral density. Moreover it’s a great advancement to divide the signal into frames, or chunks of 20 ms where we can consider in these periods of time, the speech can be considered as stationary signal. Many frames form a bloc, which can be taken from an arbitrary signal x, sampled with period T at nT times, where n is an integer. The sample vector can be given by:

\[ x_N(n) = \{x(n), x(n-1), \ldots, x(n-N+1)\}^T \]  

(3.1)

The length of frames is decided by changes in the shape of vocal tract. Frames of 20 ms are used for pitch detection, longer frames are used for multiple pitch periods. Consecutive frames may be overlapped for example if we have a vector of N sample, a new vector can begin after M \div N and so on every new M samples it will create a new vector. The reason for that is the continuity in transition between vocal tract shapes. Models for speech The
most common assumption, it says that speech and noise are uncorrelated and the noise statistics vary slower than the speech. However a good model to consider for speech would be an auto-regressive model (AR) which can be suitable for speech divided into frames. In this model we have a prior knowledge of the speech signal which can be used to smooth the signal for further processing. In the AR model the current value (sample) determined by previous values (samples), the present sample $x(n)$ is given by the equation:

$$x(n) = x_{LT}^T(n - 1)\vec{a}(n) + S(n)$$  \hspace{1cm} (3.2)

$\vec{a}(n)$ is a vector of length L, $x_{LT}^T(n - 1)$ is an L-vector of previous samples and $S(n)$ is an excitation or a source signal. $S(n)$ can also be regarded as the error signal, i.e. the difference between actual signal and the estimated one. The signal is modeled as a recursive filter $x(n)$ with $S(n)$ as input. Two cases for the excitation signal either Gaussian random noise for unvoiced or an impulse train for voiced speech.

### 3.2 Signal to Noise Ratio

A measure of how strong is the signal compared to the noise is an important quantity called signal-to-noise ratio (SNR). It's given in dB

$$SNR(dB) = 10.log_{10}(S/N)$$  \hspace{1cm} (3.3)

Where S,N are Signal, Noise powers respectively.

### 3.3 Stochastic Signals and Analysis

Usually if we want to describe a known (deterministic Signal) we use means such as like Fourier transform or Signal power, it’s adequate to use these quantities for periodic or quasi-periodic signals, but for stochastic signals of a random nature various quantities were suggested to characterize these Signals. These quantities are described briefly in the following sections.

### 3.4 Probability Density Function(PDF)

To describe this fundamental quantity, we take a time series signal $x[n]$, and intuitively before we calculate the PDF we record how often certain intervals are hit by the amplitude of $x[n]$. Then, by taking large sample set we will have the histogram, and by making the bins of the histogram infinitesimally small and taking the area normalized to one we will get the probability density function PDF. From the shape of the PDF provide many characteristics like signal power $\rho_x$, the variance, and the mean expectation of the signal $\mu_x$. There are famous PDF distributions such as the Gaussian distribution, the signal which has the Gaussian PDF will have also a Gaussian bell shaped distribution.
To overcome the randomness of the signal, we use deterministic values to describe the signal such as the mean $\mu_x$ and Variance $\sigma^2_x$ which can be seen in the equation of the Gaussian Distribution:

$$\rho_x = \frac{1}{\sqrt{2\pi\sigma_x}} e^{-\left((x - \mu_x)^2\right)/2\sigma^2_x}$$  \hspace{1cm} (3.4)

the previous equation gives mathematical representation of the PDF, which evaluates many important parameters like the mean and the variance. since, the signal used here is of random nature. Thus, to calculate the average for this signal would require big number of samples to overcome randomness, but we rely on the expectation operator $E\{\cdot\}$ by averaging over an ensemble of values. The Ensemble by definition is a infinite realization

Figure 3.1: Gaussian Distribution

3.5 Expectation, Mean, and Variance

To overcome the randomness of the signal, we use deterministic values to describe the signal such as the mean $\mu_x$ and Variance $\sigma^2_x$ which can be seen in the equation of the Gaussian Distribution:
of many random signals all parallel in time and share the same statistics PDF, mean ... etc. all the signals used are ergodic. Thus, the notion of ergodicity allow us to average over time instead of ensembles.

$$\mu_x = \mathbb{E}\{x\} = \int_{-\infty}^{\infty} x.p(x)dx$$  \hspace{1cm} (3.5)

$$\sigma_x^2 = \mathbb{E}\{(x - \mu_x)^2\} = \int_{-\infty}^{\infty} (x - \mu_x)^2.p(x)dx$$  \hspace{1cm} (3.6)

we will see in our thesis work the importance of such parameters when we need to estimate the signal-to-noise ratio which give us as we will see how robust is our algorithm.

### 3.6 Correlation and Power Spectral Density

As its name suggest the correlation means how one random signal is correlated to the another. Suppose we have two signals $X[n]$, $Y[n]$ then we take the similarities between them as a function of relative shift i.e lags in $k$ samples for example then we get the cross-correlation function

$$r_{xy}[K] = \mathbb{E}\{x[n + k].y^*[n]\}$$  \hspace{1cm} (3.7)

one good special case from cross-correlation function is the auto-correlation function, which relate the present signal to itself but with time shift. it’s also very useful to evaluate how predictable successive signal samples are.

$$r_{xx}[K] = \mathbb{E}\{x[n + k].x^*[n]\}$$  \hspace{1cm} (3.8)

this function has important characteristics, it’s symmetrical to lag zero and complex conjugation, and at lag zero it has the maximum value. Obviously because the signal is the same as itself at lag zero.

Now we define the Discrete Fourier Transform, which is the Fourier Transform for discrete finite signal. This transform suit our needs perfectly since our signal is represents by its samples and every time we study the signal we examine one sequence (chunk) at the time.

The DFT is given by the equation

$$X_k = \sum_{k=0}^{k=N-1} x_n.e^{(2\pi si/N)kn} \ \ k = 0, \ldots, N - 1$$  \hspace{1cm} (3.9)

or another notation using $\Omega$

$$X(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} x[n].e^{-jn\Omega n}$$  \hspace{1cm} (3.10)
It is very useful to have the inverse operation especially in practical approaches such as inverse filtering, so the inverse transform is:

\[
x[n] = \left(\frac{1}{2\pi}\right) \int_{-\pi}^{\pi} X(e^{j\Omega}).e^{j\Omega n}
\]

(3.11)

The previous transform exist for finite energy signals, and not random signals in our cases so we have to define spectrum for random signals by relying on PSD of the previously mentioned auto-correlation sequences, by taking the Fourier transform of these sequences.

\[
P_{xx}(\Omega) = \sum_{x=-\infty}^{\infty} r_{xx}[k].e^{-j\Omega k}
\]

(3.12)

### 3.7 Discrete Fourier Transform

We can write any signal as the sum of many sinusoidal signals. We define a linear time invariant system LTI as a system that preserve the shape of input signals at the output. Thus, a sinusoid is the sum of different sinusoid at the inputs, these signals varies in amplitudes and summing those signals linearly gives the output sinusoid. In this fashion system’s function and its behavior at specific frequencies.

\[
X(e^{j\Omega}) = \sum_{n=0}^{N-1} x[n]e^{-j\Omega_0 kn}
\]

(3.13)

As we said before, the transform exist in discrete frequencies \(\Omega = 0, \Omega_0, \Omega_1 \ldots\) and \(\Omega\) it’s written \(\Omega_0 = 2\pi/N\) then we write:

\[
X[K] = \sum_{n=0}^{N-1} x[n]e^{-j\Omega_0 kn}
\]

(3.14)

DFT is used in finding Fourier Transform to study the spectral characteristics of the signal and preform speech enhancement and noise reduction schemes.
4 Study of Noise Estimation Techniques

4.1 An efficient algorithm to trace the noise floor

4.1.1 Abstract

This algorithm was initially developed by Martin [6]. The technique used in this algorithm doesn’t require a speech voice activity detector to calculate spectral density of the noise nor does it need to calculate a histogram. It is based on assuming that noise has minimum spectral values, so we calculate the minimum values of the smoothed power estimate. This method can be used to estimate the noise power and calculate the Signal-to-Noise ratio in a noisy environment.

4.1.2 Introduction

We believe that there are good reasons which led to the development of this algorithm such as noise estimation with less computational complexity in speech processing techniques like our thesis purposes, also this algorithm is very efficient in rapid changes in noise levels which are desirable in real-time application.

To reduce the error in noise estimation we divide the whole spectrum of the signal to small intervals of 0.02 to 0.1 sec which is typical for these applications.

The advantage of this algorithm is: there is no need for decision to distinguish speech/no speech segments, and it can track variable noise continuously while, speech is present in the frame. The algorithm is based on observing the spectral envelope of the signal and it is noticed that, the spectral envelope contains peaks and valleys. Hence, we can assume that peaks correspond to speech activities and valleys to the noise power. As mentioned earlier, it is advisable to have a better performance, to take the smoothed power envelope. So, basically the algorithm tracks the valleys (minimum) in the smoothed power spectrum and creates from it the noise power estimate within a window of finite length.

We perform a Matlab simulation of this method but without calculating the SNR, we estimated the floor noise from the minimum of the smoothed power spectrum.

4.1.3 Description of algorithm

We will assume that the noisy signal is the sum of the speech signal \( s(i) \) and noise signal \( n(i) \), the composed signal \( x(i) = s(i) + n(i) \) where the representation is in the time domain and taking into account that \( s(i) \) and \( n(i) \) are statistically independent. So, we can write \( E\{x^2(i)\} = E\{s^2(i)\} + E\{n^2(i)\} \).

The computation of the noise power estimate \( P_n(i) \) is obtained by taking the minimum of the smoothed short time power estimate \( P_x(i) \) within a window of \( L \) samples.
This algorithm can be divided into two main parts:
1. Calculating the smoothed power spectrum estimate over a short window of time.
2. Find out the minimum in this power estimate and determine the noise power estimate $P_n(i)$

We compute the smoothed power estimate by first, calculating the power spectrum which can be obtain by the FFT algorithm or by using an exponential sliding window of length $N$.

Let $\tilde{P}_x(i)$ be the smoothed power estimate, all the smoothing takes place in short time slot at a time index $i$. The smoothing is preformed by means of a recursive system where the smoothing factor $\alpha$ is typically chosen between 0.95 and 0.98.

The recursion starts at $i > N$

$$P_x(i) = P_x(i-1) + |x(i)|^2 - [x(i-N) \ast x(i-N)]$$  \hspace{1cm} (4.1)

$$\tilde{P}_x(i) = \alpha \ast \tilde{P}_x(i) + (1-\alpha) \ast P_x(i)$$  \hspace{1cm} (4.2)

$N = 128$ or $256$, it is safe to assume that $N=156$ is in fact is a short time interval.

4.1.4 Noise power estimation

Let’s assume that we have a window of $L$ samples and, it is needed to have the noise power estimate, one method is to track the minimum from the signal power within this window. For complexity and data delay reasons, it is recommended to decompose the window with $L$ samples further to smaller windows with size $M$. Therefore, $M \ast W = L$ we chose the sampling frequency to be $f_s = 8KH\text{z}$.

Typical values in these type of experiments as Martin [6] suggested are: $M=1250$, $W=4$ and $L=5000$ these parameters correspond to a time slot of 0.625 sec.

Every $M$ samples we conduct a sample by sample comparison between the actual minimum $P_{M\text{min}}(i)$ and the smoothed signal power $\tilde{P}_x(i)$.

Whenever $M$ samples have been read, i.e. $i = r \ast M$, we save the minimum power $P_{M\text{min}}$ of the last $m$ samples and reassign $P_{M\text{min}}$ to its maximum value: $P_{M\text{min}}(i = r \ast M) = P_{\text{max}}$

In real life applications we observe two kinds of noise power. First, slow varying noise power, in this case we have a power that doesn’t increase constantly (non monotonic power) then the noise power is set to minimum of $L$ samples, i.e. $P_n(i) = P_{L\text{min}}(i)$ which is obtained by taking the minimum for last $W$ minimum power estimates:

$$P_{L\text{min}}(i) = \min(P_{M\text{min}}(i = r \ast M), P_{M\text{min}}(i = (r-1)\ast M), P_{M\text{min}}(i = (r-2)\ast M), \ldots, P_{M\text{min}}(i = (r-W+1) \ast M))$$

Secondly, monotonically increasing noise power which means that minimum power of the last $W$ windows in always increasing therefore, the noise power is equal to the minimum of the last $M$ samples $P_{n}(i) = P_{M\text{min}}(i = r \ast M)$.

If the estimated noise power is bigger than smoothed power then $P_n(i)$ is updated to the minimum of the two, i.e. $P_n(i) = \min(P_x(i), P_n(i))$. 


4.2 Noise estimation by using speech pause detection

4.2.1 Introduction

The new advances in speech recognition and mobile communication were the driving force for enhancing noise reduction techniques. Speech pause detection algorithms play an important role in many of the single microphone noise reduction applications.

It is common in speech applications that the noise is not stationary. Thus, to have an accurate estimation of the noise we need to update the noise spectrum estimate gradually. Any sudden changes in the level of noise then will be considered and fed to the application with minimum delay.

Those noise level estimates can be calculated whenever speech is missing from the frame, which explains why we need an algorithm for pause detection and, when a pause occur we calculate the level of the noise at these instants.

As we saw previously in Martin [6] we don’t need an explicit pause detection mechanism, instead we continually update the noise estimate based on the minimum signal power envelope within a window of 1 sec this is considered as noise power.

Algorithms which avoid pause detection are faster to track non-stationary noise fluctuations because they adapt to changing noise levels even during the presence of speech. Nevertheless updating continually the noise estimate in the sub-bands independently is susceptible to erroneously capture speech energy and consider it as noise.

Other scientists like Fischer and Stahl [4] tried a spectral subtraction noise reduction algorithm with a continuous noise spectrum updating scheme, but the speech corrupts the noise scheme and this cannot be taken lightly. Thus we come to a conclusion that voice activity detection algorithms are necessary for noise estimation and reduction.

4.2.2 Algorithm

Our algorithm was based on Marzinzik and Kollmeier [7] which tracks the envelope dynamics to estimate whether speech is present or pause. This algorithm suggests that whenever there is a pause at this instant, the envelope spectral characteristics represent the noise characteristics. It is shown in the next equation that in order to calculate the signal temporal power envelope we apply DFT transform to the input signal. Next step is summation of the squared frequency components over the whole band.

\[ E(p) = \sum_{k} |X(p, w_k)|^2 \]  \hspace{1cm} (4.3)

\( X(p, w_k) \) is a spectral component of this signal at time frame p. For pause detection, one technique is to divide the whole spectrum into high pass and low pass characteristics. This approach is useful for us to determine if the pause is of low-pass or high pass nature then, according to its nature we would process it adequately.

\[ E_{LP}(p) = \sum_{l} |X(p, w_l)|^2 \]  \hspace{1cm} (4.4)

\[ E_{HP}(p) = \sum_{m} |X(p, w_m)|^2 \]  \hspace{1cm} (4.5)
In above l goes from zero to cut-off frequency, while m runs in the rest of the spectrum (high-pass), it is recommended to apply smoothing to the spectral components by averaging. For low pass averaging we use a low pass recursive filter over short time intervals with release time $\tau_E$ but, if the signal is increasing then, it is best to stop the smoothing to avoid smearing over the onsets. Later in this short summary of the algorithm we are going to show how that algorithm will track the minimum and maximum peaks of each spectral power envelops, and how the pause detection decision is made according to the scheme:

1. First step is, to initialize the process. We leave 200 ms for initial phase of noise only, after that we assign the maximum and minimum values as follows:

   \[
   E_{\text{min}}(p) = E(p) \quad E_{\text{max}}(p) = E(p) \\
   E_{LP,\text{min}}(p) = E_{LP}(p) \quad E_{LP,\text{max}}(p) = E_{LP}(p) \\
   E_{HP,\text{min}}(p) = E_{HP}(p) \quad E_{HP,\text{max}}(p) = E_{HP}(p)
   \]  

   This assignment means that we match the minimum values of $E_{LP}(p), E_{HP}(p)$ to the noise energy at the beginning.

2. Then, we update values of the envelops in its minimum and maximum states as follows:

   a) If the current value of envelope exceed the maximum, then a new maximum is set to the current value, if it doesn’t exceed it then, the maximum value is decreased with a time constant($\tau_{\text{decay}}$) where the input to the recursive first order low pass filter are current envelope values.

   b) In a similar manner if current value of the spectral power is below the minimum assumed in the beginning then a new minimum is set to the current value, if not then the envelope is slowly raised by recursive filtering with a time attack ($\tau_{\text{raise}}$) where the input is the current value of the Envelope.

3. The differences between maximum and minimum values are calculated for each envelope:

   \[
   \Delta(p) = E_{\text{max}}(p) - E_{\text{min}}(p) \\
   \Delta_{LP}(p) = E_{LP,\text{max}}(p) - E_{LP,\text{min}}(p) \\
   \Delta_{HP}(p) = E_{HP,\text{max}}(p) - E_{HP,\text{min}}(p)
   \]  

4. To make a decision about pause or speech present, we can distinguish three cases we study the actual frame:

   a) If both the minimum and maximum values of envelope are below certain threshold $\eta$ then we determine that, no speech is present but only noise is found in this frame (this case we call low dynamic speech pause).
b) We decide whether pause is detected according to information from low pass band.

c) The same decision but made upon information taking from high pass band information.

i. If $\Delta_{LP} < \eta$ and $\Delta_{HP} < \eta$
   
   First case explained earlier represented of low range dynamics and only noise is assumed to be present in this frame.

ii. If the previous condition is not met and if $\Delta_{LP}(p)$ then, there are very small dynamic changes and no LP pause is found in this frame but is $\Delta_{LP}(p)$ is bigger than $\eta$, at the moment we examine if the difference between $E_{LP}(p)$ and $E_{LP,\text{min}}(p)$ is smaller than $\rho_c$ of the $\Delta_{LP}$ we conclude that the envelope values are close to its minimum, and we should focus on the high pass-band characteristics to make the right decision about whether there is a pause detected or not.

   - If the difference $\Delta_{HP}$ is smaller than $\eta$ then no further investigating is required and we are certain that speech pause is detected because of the low dynamics in high pass band range, if this condition is not met then we cannot determine that pause speech is detected, thus we cannot make a decision.

   - If the difference in the high pass band $\Delta_{HP}(p)$ is bigger than twice $\eta$, then there are dynamic changes enough to pay attention to the high pass band. Thus, we examine a new criteria if the difference between the current envelope $E_{HP}(p)$ and $E_{HP,\text{min}}(p)$ is less than twice the fraction $\rho_c$ of $\Delta_{HP}$ then we assure that with the small envelope in low pass band, it is indeed a speech pause. Otherwise, if this condition is not fulfilled then, speech might be found in this frame.

   - If the difference is smaller than $2\eta$ but bigger than $\eta$. This case is a bit ambiguous because it is required that $E_{HP}(p)$ lies in its lower half of its dynamic range then, we can be sure that a speech pause is present, otherwise speech might be present in that frame.

5. All the b) section assumes that the mitigated noise is of high pass nature. Therefore the decision whether there is a speech pause is made based on information from the low-pass band. Now to consider the other case, in which the disturbing noise is of high pass nature, then we apply the same study with same conditions examined but we exchange every LP with HP and vice versa.

Now the whole algorithm is fully applied, we need to test its robustness, so we introduce the false alarm-rate (when a pause is detected when speech is present, or the other way around). And in this algorithm we can actually adapt the threshold $\eta$ and it fragment $\rho_C$ to have an optimal false-alarm rate. When we have a low false-alarm rate (which is the optimum case), it will reduce the speech distortion in the subsequent noise reduction process. Nevertheless, there is a trade off here where hit rate is significantly reduced. To test its
performance, we generated different kinds of noise, such as factory noise, car noise... etc. We used different levels of noise in dB’s and different SNR’s. In the experimentation phase, we find out the best sampling frequency for the DFT of the signal is 22050 Hz, then passed by partitions of 8 ms windows and padded by zeros to avoid delays which can cause lip reading or stuttering when somebody is speaking (in the synthesis phase). The cut-off frequency that separate between low-pass and high pass frequency is chosen between 1.9 KHz and 2 KHz, if chosen below that it affect the intelligibility of the speech. The time constant for envelope smoothing was set to 32 ms, $\tau_{\text{raise}}$ and $\tau_{\text{decay}}$ were set to 3 s these value simulate the actual signal in real life under normal condition. Threshold $\eta$ is 5 dB and fragment of it is $\rho_c$. We decided to implement the second method of finding speech pauses since we need to estimate speech power as much as we need background noise power to form a proper feedback signal for the deaf to adjust his level of talking.
5 Design and implementation

Experiment terminology

- Audio sample: On our system for both online and offline we used the same audio file, it is a female audio sentence of length 2.866 sec and it says “don’t ask me to carry an oily rag like that” and it is sampled on 22050Hz.

- Hit rate: main method to test the robustness of our system, the hit rate indicates the percentage of the right decision made by the algorithm to detect the presence of speech in a given sentence.

- False alarm: is the second method of testing the robustness of the algorithm. And it indicates the percentage of samples where decision of the algorithm is the opposite from what we are expecting.

5.1 Offline approach

5.1.1 Introduction

In our offline implementation of algorithm we divided our program into four parts, first part contains the power spectrum calculation and the smoothing of it, in the second part we implemented the algorithm and the speech pause detection decision is then sent to post processing part which contains power calculation and other techniques to improve the result of algorithm, and at the end we have the fourth and last part where output sent to deaf person is implemented.

Offline loop processing uses pre-recorded audio samples which help to deduct clear results about the system and how to implement it online. Moreover, it is easier to implement offline which means more processing techniques can be implemented. On the other hand in online approach various limitations exist which makes these techniques harder to implement without proper adjustments. We used speech pause detection algorithm suggested by Marzinzik and Kollmeier [7], it was the key point in our system to calculate noise and from this estimation we form and send out an output message to our deaf person. Thus, testing speech pause detection and its efficiency is the most important part in testing the robustness of whole system in general.
5.1.2 Testing methods

During our implementation of that algorithm we depend on parameters and values suggested by Marzinzik and Kollmeier [7]. Other parameters were introduced by us which, we found more suitable by observing the results. After the noise rises to high levels, it almost masks the voice of speakers around it. Thus, the speech level in the signal became unimportant since it’s masked by the noise anyway and the idea of our system is to send a signal to the deaf person regarding the noise level. Therefore, differentiating between pause and speech is no longer crucial and that gives us flexibility.

5.1.3 Speech pause detection implementation and tests

**Smoothing factor**

The smoothing factor is one of the most important parameters that affect the performance of the algorithm. In Marzinzik and Kollmeier [7] were the speech pause detection algorithm was introduced, the author suggested a smoothing factor between the values 0.93 and 0.96 for best performance of the algorithm. For testing what effect the smoothing factor has on the **hit rate** and the **false alarm** of the speech pause detection algorithm, we used a standard audio file, adding to it a fixed amount of noise (Factory noise and Car noise) and, changing the amount of the noise until we got 30 results ranging from -10 dB to +20 dB we ran the modified algorithm with three different values of \( \alpha \) (smoothing factor) 0.95, 0.90 and 0.85 then, we compare the results. In figure 5.1(a) we plotted the **hit rate**

![Hit rate to SNR](image)

(a) **hit rate** to SNR

![False alarm to SNR](image)

(b) **false alarm** to SNR

Figure 5.1: **Hit rate** and **False alarm** in factory and car noise with 0.95 smoothing factor with a smoothing factor 0.95 and for two different types of noise, car and factory noise. We see that the algorithm works better for the factory noise. Moreover, the algorithm’s **hit rate** decreases when we get near the value 0 dB till it reaches almost 18% of its value at -10 dB. While we can see in figure 5.1(b) the **false alarm** rate for the output of our
algorithm when smoothing factor is 0.95 and for both car and factory noise, we can see from the graph that the false alarm rate will rise to 60% for -10 dB. At this point the algorithm is not able to detect the speech in the audio file due to the high level of noise and, we see that, on a high dB’s (over 7 dB), false alarm will fall down to almost 20%. The cause for false alarm drop is due to short peaks which result in false detection of speech and that was solved by means of using a filter which, will be introduced later in the paper and, it was used only in the offline implementation.

In the figure 5.2(a) we did the same as previously for hit rate, but this time by using a smoothing factor of 0.90. By comparing with the results from the 0.85 smoothing factor we found out that the hit rate has improved by 11.5% on the average level for car noise and by 16.13% on the average level for factory noise and also we had noticed that, it worsen the efficiency of the algorithm (when the hit rate falls down to 65%) the SNR is around -5 dB while it was almost 3 dB for smoothing factor by 0.95 Figure 5.2(b) indicates false alarm of our algorithm when smoothing factor is 0.9, and we see from the results we know now that, reducing smoothing factor to 0.9 has improved the false alarm rate to 5.69% lesser than that in 0.95 smoothing for car noise and 8.14% for factory noise.

![Figure 5.2: Hit rate and False alarm in factory and car noise with 0.9 smoothing factor](image)

Next results are observed from finding hit rate and false alarm for the algorithm when using smoothing factor of 0.85. Figure 5.3(a) shows the hit rate of our algorithm when the smoothing factor is 0.85, and we can see an obvious improvement in hit rate compared to the result found with higher smoothing factor, we got 7.83% improvement of average level with smoothing of 0.9 and 19.23% with smoothing factor of 0.95 for the car noise, and 7.38%, 23.51% respectively for factory noise.

Figure 5.3(b) shows the false alarm result of our algorithm when the smoothing factor is 0.85, and we can observe that there is also a small improvement for false alarm by reducing the smoothing factor, false alarm rate has been reduced by 3.48% and 9.17% for smoothing factor of 0.9 and 0.95 respectively for car noise, also 3.08% and 11.3% for
Figure 5.3: Hit rate and False alarm in factory and car noise with 0.85 smoothing factor

smoothing factor of 0.9 and 0.95 respectively for factory noise.

From the results we can conclude that, reducing smoothing factor improve performance of the algorithm and the suggested system as whole. A very important draw back for reducing the smoothing factor under 0.93 is that there will be wrong decisions when we have a noisy environment without speech, we see that effect by applying only noise audio sample and the algorithm still considers some segments as speech. In our case we used car noise and a smoothing factor of 0.85 and the result is shown in figure 5.4.

Figure 5.4: Car noise with 0.85 smoothing factor and the output of the algorithm

Due to lowering the smoothing factor, the power spectrum of the noise will have sharp edges and rapid fluctuations. Therefore, the algorithm will make wrong decisions taking noise for a speech, and by observing the result of false alarm we found that, it is raised up
to 25%, and that’s not really satisfactory. Because it was 0% when 0.95 smoothing factor was used. In our suggested system we depend on detecting the parts were the speech is not presented, to be able to estimate the power of the noise, and send a feed-back to deaf person. The feed-back signal adequacy is not affected if some parts of the noise are considered as speech, since there will be enough pause intervals to update our noise power. By observing the previously mentioned results, we found that the using of a mode switch, which enables the system to use a high smoothing factor (0.95), in environments where speech is not presented, and also in a situations where the noise level is high that it masks the speech. In this case, detecting the speech is not a top priority. The second mode of switch is used when conversation occur, where we need to perform a very robust pause detection technique with high hit rate of pauses, in this case it is preferable to use low smoothing factor (0.85) to determine speech and pause parts accurately, and construct feed-back signal accordingly.

5.1.4 Post Processing

Improvement on the Algorithm

Another problem which we encountered while observing the results, is that we have some times wrong decisions on short number of samples the algorithm considers them as speech, while in fact they are small spikes of noise. An improvement was introduced to solve this problem, we used a filter that removes all peaks that were interfered wrongfully, which are shorter than 15 samples. \(^1\) To see how the filter improved the performance of the algorithm, we applied same criteria that were used before. i.e. see how hit rate and false alarm of algorithm’s output from algorithm was affected. Of course the same study was repeated for all smoothing factors. In figure 5.5(a) and figure 5.5(b) we can see the hit rate and false alarm for a smoothing factor of 0.95 and after using this filter.

There was a degradation of 0.66% and 1.18% on the hit rate for car noise and factory noise respectively after introducing the filter, and no change in false alarm. In figure 5.6(a), and figure 5.6(b) we see the result of hit rate and false alarm when using 0.9 as a smoothing factor.

And we observe a degradation also occur in hit rate up to 4% and 5.8% for car noise and factory noise respectively, but false alarm remains the same. Figure 5.7(a) and figure 5.7(b) shows hit rate, and false alarm after using the filter for a smoothing factor of 0.85.

And we observe also a higher degradation in performance up to 7% in Hit rate for both car noise and factory noise and a 1% improvement in False alarm. The best environment in which, the filter will perform optimally is when only noise exists, and to have even better results we combine the usage of the filter with low smoothing factor in only noise environment. We enter only noise to our algorithm (smoothing factor of 0.85), check the result with and without filter, and compare them to each other. The results are shown in figure 5.8

We see from the latest results, that false alarm had been reduced from 25% to 0% after

\(^1\)corresponds to 60 ms which is less than even a letter.
using the filter. So, in this case we added filter with a higher (or similar) smoothing factor when there is only noise. Thus, it will reduce faulty decisions about noise, and not consider it as speech.

**The power calculation**

In this stage of the system, it calculate the power of the audio sample where the algorithm decided that it is a noise, to be processed later and changed into feedback signal to be sent to the deaf person. Also in this stage it calculates the power of the speech to be used to improve our system and also suggest different solutions for sending the best information to the deaf person regarding the level of other Speaker’s speech. Since the testing is offline, that’s gives us more flexibility in calculating and processing the power. The suggested way to calculate the noise power when the algorithm determine a no speech segment, the system will save the power of that sample and check the one after and save it if it’s noise also, when the system finds a speech sample, then it calculate the average of previous noise power samples, in case of having long periods of just noise in the frame without speech, the system will calculate the power each 1.5 sec, and in this case it will adapt to the changes of noise level in the ambient environment. When speech presents in the mixed signal, it is also averaged in the same way and saved to be used in the output part where feedback signal is shaped. After finishing power calculation for both noise and speech, we get values of averaged powers along the audio sample, where the power of speech and noise are known and can be used.
5.1.5 Output signal to the deaf person

This is the last stage of processing, and forming final post-processing output signal, which will be sent to the deaf person containing the information needed to have a feeling about surrounding noise and speech levels, and then be able to adjust his voice depending on it. The signal which will be sent to the deaf person is created from the power calculated in previous stages and scaled, and then form a vibration which changes in frequency or intensity depending on output from the algorithm. There was different ways we used to shape the vibration output signal in our offline implementation of the system. The first is based on noise power only, and the second depending on both noise power and speech power.

Depending on noise only

To create an output for the system which would only depends on the noise, we followed two methods. Firstly, it depends on changing the frequency or the amplitude of a sinusoidal, and the second depends on Matlab function beep, this method is based on forming a loop of function beep that depends on noise power. In the first method, we used two overlapped sinusoidal waves to create one wave where its frequency depends on the noise power level. The first sinusoid has a very high frequency and the other overlapping sinusoid frequency depends on the power of noise, the higher the power the higher is the frequency. Those two waves are overlapped and the output is created only if, the second signal is over zero, as shown in figure 5.9, and we got a beeping message that depends on the level of noise, the higher the noise the faster is the beeping and the lower the noise, the lower is the beeping. In this method we used a sinusoid of a length of 1.5 sec which will give us information.

Figure 5.6: Hit rate and False alarm in factory and car noise with 0.9 smoothing factor after filtering.
about the level of noise each 1.5 second.

And the other method depends on Matlab function beep and one creating a loop that delays the continuing of the execution of the big loop by a certain time that depends on the level of noise, the higher is the noise the shorter is the delay time, so the beeping will be faster, if the noise is high. The lower is level of noise means longer delay. So, it signifies slower beeping. This method was not efficient since, the beep function and the delay both depends on the speed of processor of the platform that we are working with.

**Depending on both noise power and speech power**

This method rely on both noise and speech powers to give better perception to the deaf person. Depending on power of the speech we got from previous stages in the system and by assuming that speech power we got from the algorithm is the power of the speech of the deaf person himself. since the separation of the speech between the speaker and the other speakers is difficult with one microphone, we tried other ways to overcome this problem by assuming that other speakers will speak in a reasonable right level (+ or - 3 dB from the optimum conversation level 60 dB), and the person will speak in the higher or lower than this level, and in this case we discard the power which is, near the optimum and we use the one far away from it. We used the power of speech of deaf person to send him information about whether his voice is high and need to be lowered or vice versa, this is done by multiplying our output signal, created from the noise power and depends on its frequency, which is multiplying with decreasing signal such as a ”Line-space command” in Matlab which, depends on speech level. So, if the speech level is higher than the optimum (which also depends on the noise) the Linespace will change from 1 to 0 to tell the speaker...
Figure 5.8: Car noise with smoothing factor 0.85 and the output of the algorithm before and after filtering

to lower his voice as shown in the figure 5.10

And if the speech power is lower than the optimum it will be multiplied by Linespace going from 0 to 1 along the 1.5 sec output, so the deaf person will know he is advised to raise his voice, as shown in figure 5.11. And once the deaf person became close to the optimum area Linespace will not change anymore (a fixed line on value 1), the only component left in the output is the frequency that, depends on the noise.

An example about a speech mixed with car noise increased intentionally to a high level and then going down is seen in the figure 5.12, where it shows that increasing the noise increases frequency of output and in at the same time giving a message to deaf person to increase his/her voice depending on the speech power, we can infer from the output that the deaf person increased his/her voice in the beginning and then it reached a level where he/she didn’t need to increase any more, but with noise level suddenly increased, he/she missed that stable stage and asked to raise his/her voice again. (In our recording the speaker didn’t adapt his/her voice to the signal was receiving from the system, so he/she was constantly asked to raise his/her voice).

for the graph to be more clear, it was also done with only the input power and the output signal result figure 5.13
5.2 Online approach

5.2.1 Introduction

In order to test our online algorithm we used the Matlab platform put together by Dr. Nedelko Grbic in the signal processing lab at BTH in Ronneby. That platform provides us with a multi microphones live feeding to a PC where the processing take place and fed back the output directly from the device, so it is consistent with our concept for online real-time algorithm (frame in frame out concept). Changing the concept of recording audio files and use speech live, drove us to change the logic of implementing the algorithm, and implementation where we can’t use previous samples for making better decision, and we have based our decision whether there is a speech present or noise upon present sample only. Thus create an output for each input signal, which on the other hand gave us an extremely fast updating for output when the input changes. The offline implementation was the corner stone to implement the online approach because it is based on the same principles, and one of ground tools is the smoothing factor which plays a very important roll in securing robustness of algorithm, in online implementation we used same values and parameters we were using in the offline implementation, and also we applied the same concept of having two smoothing factors depending on the expected noise environment and weather we are in a conversation situation or an only-noise environment. The architecture for online system is the same as for offline except that, the four stages in the offline used to process the whole file from beginning till end on each stage before moving to the next one. While, the online gets sample by sample from input signal, and process it through the system, and produce an adequate output.
Figure 5.10: Combining the changes in the frequency and the magnitude of the output signal, decreasing

5.2.2 Testing Method

By calculating hit rate, and false alarm of output of the algorithm for different noise levels starting from -9 dB and up to 15 dB with 3 dB increase. The audio sample file used for testing the online algorithm is the same used in the offline algorithm, a female audio sentence of the length 2.866 sec and it says "don’t ask me to carry an oily rag like that" and it is sampled on 22050Hz. And the noise used is a factory noise. To find the SNR in each test we have made four different levels of noise, we put the microphone 20 cm away from a speaker, and then played the audio sample of clean speech, calculate the average power of it, and then used another speaker 20 cm away from the microphone to find the average power of noise, and find the SNR for them using the equation.

$$SNR = 10 \times \log_{10}(P_{speech}/P_{noise}) \quad (5.1)$$

Repeating the same procedure for calculating all our SNR’s, by fixing the clean speech power and changing the noise level and calculating the power of it, and then mixing them to have the composite signal. Afterward, we run our trials to test the system.

5.2.3 Speech pause detection implementation and tests

We used in our tests two smoothing factors 0.9 and 0.95, we know from our offline study that, the best choice for only noise environment is higher smoothing factor, and a lower smoothing factor for conversation environment. We used the same testing method as in the offline implementation depending on false alarm and hit rate described in the previous
Figure 5.11: Combining the changes in frequency and magnitude of output signal, increasing

section, by comparing the speech pause detection output with the real speech pauses we have in our audio file. After making 9 trials, for SNR from 15 dB to -9 dB we had the following results. For a smoothing factor of 0.9, Figure 5.14 shows output of the algorithm for a 15 dB and also the actual speech pauses we got from the audio sample. The work of algorithm with low noise environment gave us up to 81.46% of hit rate and only 10% false alarm. Increasing the noise up to 9 dB and 6 dB gave us a decreased in the hit rate down to 76.97% and 76.97% respectively, and increase in the false alarm up to 14.4% and 15.6% respectively.

And by increasing more noise level up to 3 dB we reach to a 68.54% Hit Rate which means degradation of 16% from what we had in 15 dB noise level, and a false alarm up to 18% which means 8% more that what we had in 15 dB. Figure (21) shows the output from our algorithm compared to real pauses in the speech we get from our audio sample. By increasing further more the noise to reach a levels of -6 dB and -9 dB we found that the performance of our speech pause detection algorithm is degraded severely, at -6 dB the hit rate fall down to 52.25% and at -9 dB it reached 42.13%, on the false alarm side, it was up to 35.2% and 38.4% for -6 dB and -9 dB respectively. Figure 5.15 shows the output from the algorithm at SNR equal to -9 dB compared to real pause in speech we had from our audio sample.

To find the differences of using lower and higher smoothing factor, we did the same test, but we changed the smoothing factor from 0.90 to 0.95 and we compared the results. Raising the smoothing factor to 0.95 didn’t change significantly in low level noise environment (6 dB and higher) where we see that the degradation on hit rate is in the range from 1% and up to max 4%, and on the false alarm we observed that the degradation was only up
to 4%. By raising the noise to a higher levels (-3 dB and lower) we find that the effect of lowering the smoothing factor is obvious were hit rate was degraded from 52.25% to 19.1% at -6 dB and false alarm is higher by 15% for same noise level. Figure 5.17 shows the results for hit rate for both smoothing factors (0.9 and 0.95), and for noise levels ranging from 15 dB to -9 dB.

And we can infer from the results that, lower smoothing factor gave us a lower false alarm and more robust results. From the previous tests we can conclude that, the lower smoothing factor gives us better hit rate and a lower false alarm which, match with the conclusion we found in the offline implementation. Nevertheless, the main draw back of using a lower smoothing factor is the degradation of false alarm for noisy environment when no speech is present. If we compare using smoothing factor of 0.95 or 0.9 for a noise only environment, where the input is a pure factory noise without speech, the false alarm is improved from 8% to 31.87% when 0.9 is used.

In pure noise environments, the false alarm occurs when the algorithm takes a certain parts of the noise and consider it as a speech, and in this case gives us a faulty estimation about the speech power. But if the speech power is not used in creating the output signal sent to the deaf person, this will not have a significant impact on the robustness of algorithm.

5.2.4 Post processing

Improvements on the algorithm The output of the algorithm was not improved by any post processing method. Since, live concept of our online implementation has put constraints
on the way we can improve the output. Each input sample fed to the algorithm will have an effect on the output regardless from the sample before it and after it. Power calculation

Depending on the output from our speech pause detection algorithm we calculate the noise and the speech power. This calculation is done sample by sample, where no averaging is used in order to maintain our online concept. The noise power and the speech power can be used for creating the output for the deaf person.

5.2.5 Output signal to deaf person

To create the output signal for deaf person, we used in our online approach only the noise power to make the decision about how the shape of the output will look like. We didn’t use in our online approach the power of speech because, we where not able to make any post processing for the output, and not able to implement a filter to take away the sudden faulty decisions by the algorithm for some samples, where the algorithm consider them as a speech while they were noise. The first method we used in the creation of the output for the deaf person is depending fully on the online approach, where we tried to test how our output will be if we feed (after scaling) the power of each single input. In that method, we added a fixed DC value for each single noise power coming from the algorithm, and then we made a decision by observing to what range of noise levels this it belongs to. Moreover, if we have speech samples in the overall signal, we assign latest value of noise power to the speech power, in this case we keep giving the same output till the next noise power comes in. Depending on which range of values the power of the noise plus the DC value belongs to a scaled value will be given to that power, and that value will be representing
the power in creating the output, the higher the noise power, the higher that scaled value is.

As we can see from the previous figure that scaled values are following the increasing of our noise, but a drawback is the fluctuation that takes place when the power changes its value by small amount around the threshold value, which decides to which level of noise it belongs. We tried to overcome the fluctuation problem and then send a better signal to the deaf person, by making an update for the noise each few samples (5 to 15 samples), which corresponds to updating the output each 100 to 150 ms, and depending on the average of those samples power, we update our scale. This method is not fully online, which means not a real time system in the sense that, we enter an input, process it, and have an output according to the processed input. Instead we have implemented a way that combine offline and online to get the type of solution adequate to our purpose. Figure 5.19 shows the effect of this method in improving our scale output. The problem with this method is that, when a false alarm occur in the algorithm, where it consider the noise as a speech, there will be no update for noise power and consequently not for scale, that will not have a strong effect on the output since, the scale will be updated at the moment when it detect noise. Most of our false alarm detections happen in a short time. Therefore, it will not affect the algorithm robustness significantly.
Figure 5.15: Speech pause detector output for a 3 dB SNR

Figure 5.16: Speech pause detector output for a -9 dB SNR
Figure 5.17: Hit rate to SNR with 0.9 and 0.95 smoothing factors

Figure 5.18: False alarm to SNR for 0.9 and 0.95 smoothing factors
Figure 5.19: The scaled values output
6 Conclusion

6.1 Suggested Prototype

We go back to our social study and draw conclusions from it. We have interviewed and listened to many feedback from LDA association and this is what we came with:

Firstly, they need a small device, not so embarrassing to them, and don’t overload them since they have already a hearing aid. Thus, we thought we can incorporate our device in the hearing aid itself having in mind that some of these devices have a noise estimator built-in already. Therefore, all we need to do is to send out a signal based on the background noise, music, and other speakers. However, our experimentation with the online algorithm shows us that creating an output depending only on one input sample is not very convenient. So we can conclude that the best way in implementing the required system is by, combining both online and offline approaches, which means applying very short delay to be able to post process the power we get from the algorithm.

Secondly, we noticed from our experiment and feedback’s from deaf society and friends who conformed our vision that we shall use a independent vibrotactile device that vibrate accordingly to the output signal generated by our algorithm to adaptively follow the background noise. The mentioned device will not be incorporated with any other device because it would be clearer for the Deaf person not to be confused with other output signals.

Figure 6.1: Speaking aid in a hearing piece
6.2 Future Work

We envision the vibra-tactile device to have many shapes such as:

**a neck collar**: the vibrator is incorporated inside the collar around the person’s neck, this solution is pretty good since it give a feeling of the vibration to the Deaf on all the neck/chest area, easy to ware and, practical. On other hand it can be embarrassing and bulky. it can be in a form of shoulder vibra-tactile pad with a Bluetooth connection to the device that calculate the environmental noise.

**vibrotactile bracelet or arm wrist** (as in the picture below)

![Vibrotactile bracelet or arm wrist](image.png)

**Figure 6.2: Vibrotactile bracelet or arm wrist**

vibra-tactile cloths which can vibrate according to the algorithm, in wish these vibrators can be installed in the sleeves or the collar. I told the gentleman who suggested this model that it might be not practical to implement because, cloths need to be washed and so on.

**Thirdly**, we can use regular pagers equipped with single-microphone constellation, our algorithm and it can provide the adequate corresponding vibration or it can communicate with the exciting ear piece (hearing aid) with Bluetooth technology.

**Fourthly**, we can use devices that has a shape of a cellular phone or even better an actual cellular phone enhanced features to collect the noise parameters and give a display of how much the level of noise in dB, or vibrates depending on that value. All the suggested devices or the output methods can be varied and can be implemented depending on the deaf person suggestions. The deaf person should rehearse with these devices until he get used to them, and be able to find the levels on which he should speak.
Bibliography


