AALBORG UNIVERSITY
STUDENT REPORT

Electronics and IT
Aalborg University
http://www.aau.dk

Title:
Testing intelligibility of re-added noisy signals after noise suppression processing

Theme:
Foundations of Sound and Music Computing

Project Period:
September 2017-January 2018

Project Group:
SMC17730

Participant(s):
Marcell Balogh
Sergi Escanes
Juan Christian Johansson
Nikos Menoudakis

Supervisor(s):
Mads Grædsbøll Christensen

Copies: 1

Page Numbers: 49

Date of Completion:
December 19, 2017

Abstract:

Due to byproducts of noise suppression in audio, unwanted auditory artifacts are introduced during processing which affect quality and intelligibility of the processed signal. A proposition expressed in a paper [8] on intelligibility suggests that adding a certain proportion of the original unaltered signal to its processed counterpart could increase intelligibility, giving us inspiration to carry out an experiment testing this proposal. An array of recorded segments of speech with added noise were processed with noise suppression, modified and tested with 20 participants for intelligibility using both objective and subjective tests, in order to assess the aforementioned proposal. We found that using a proportion of 15 percent original unprocessed speech signal to 85 percent processed speech signal resulted in highest results for intelligibility.

The content of this report is freely available, but publication (with reference) may only be pursued due to *agreement with the author.*

Preface

These worksheets represents the work and knowledge gathered by group SMC17730 in the Sound and Music Computing 7th semester, from September to December 2017. We want to thank Mads Grædsbøll Christensen as our supervisor for his invaluable input and time.

Aalborg University, December 19, 2017

Contents

Chapter 1

Introduction

One of the most crucial aspects to efficient communication in day-to-day life is the ability to understand one another. Whether it is visual or auditory, coherence in communication is essential to being able to get your message across adequately. Although it seems trivial, it is taken for granted that many of us can do this naturally, the most common example being the physical act of speaking to one another, which will be referred to as vocal communication. Unfortunately, due to either pre-existing conditions or injury, there is an estimated 350 million people world-wide that suffer from disabling hearing loss [10], and many of them therefore struggle with vocal communication due to loss of the capacity to properly capture important frequency ranges that the human ear uses to take in and process auditory information, especially in the case of human speech. The human ear is known to have a hearing range of $20 - 20000$Hz, of which $200 - 8000$Hz is the range where human speech is heard, so when hearing impairment affects the ability to adequately capture vital frequencies within this speech range, the ability to understand what is said, what we consider as intelligibility, could be hindered.

Fortunately, technology has evolved to a point where this can be helped with hearing aids, devices created to improve hearing for those with hearing impairments. These devices use a variety of techniques in order to help improve auditory information relevant to the type of hearing impairment it is used for. Although giant steps have been made to improve hearing with these devices, due to the complexity of the human auditory mechanism along with technological constraints, we are still at a point in time where hearing aids cannot completely compensate for the sound quality of unimpaired, natural hearing, and even in certain conditions fall short in being able to adequately capture and transmit auditory information.

One example is the presence of different types of unwanted auditory noise that exist in everyday situations, such as babble at a crowded restaurant or the sound of inside an aircraft cabin during flight, causing in mild cases minor annoyance and in more serious cases masking of other more important auditory information taking place at the time. Unfortunately, for hearing aids the problem of prevalent unwanted noise is also present, and in certain situations can negatively affect the mechanics of the hearing aid and the processes it undergoes to enhance certain aspects of hearing. One of these problems is the loss of intelligibility due to unwanted artifacts introduced during the process of noise suppression that hearing aids use to reduce unwanted noise in the very situations stated above.

Digital noise suppression is achieved by an algorithm which identifies noisy regions of the incoming audio signal and suppresses their volume level until it comes across audio which it considers not to be noise, at which suppression stops. Under adequate conditions, this successfully suppresses unwanted noise and overall improves the audio signal being processed. It is quite common, however, that there are situations in which the undesired noise is at a high enough volume level where it is hard for the noise suppression algorithm to differentiate between what is noise and what is not noise, possibly causing it to suppress auditory information it was not meant to suppress. The consequence to this is spurious volume modification of the audio signal that can hinder audio quality and intelligibility in the case where speech is the particular type of audio information attempted to be transmitted in these types of noisy environments.

Problem Formulation

Whilst there are different approaches to help correct this problem, one technique proposed by Mads G. Christensen et al.[8] suggests that the addition of some of the original noisy signal to the noise-suppressed signal (which will be referred to as processed signal) could help improve the overall speech intelligibility. The scope of the project presented in this worksheet will be the investigation of this particular technique, information gathered and used to help apply and test this technique by conducting a formal experiment, and analysis of data gathered to conclude whether this technique can improve intelligibility in noisy speech signals that have had noise suppression processing applied to them.

We begin in the second chapter by outlining the type of noise suppression techniques that have been developed and explaining their internal processes. Chapter 3 explains the STOI objective intelligibility measure, an algorithm designed specifically to test objective intelligibility of signals processed with noise reduction, which we used to get preliminary intelligibility results before conducting human trials. Chapter 4 goes through the test materials used for conducting our intelligibility experiment, followed by the methodology of the test procedure in chapter 5. We then present our results and their analysis in chapter 6.

Chapter 2

Noise suppression

Single microphone noise suppression

The use of directional microphones and microphone arrays is an effective way of reducing the amount of undesired interference. However, there are situations where one would like to reduce the deteriorating effects of noise when using a single microphone input to a hearing aid, or even opt for further noise reduction beyond that provided by directional microphones. Single-microphone noise suppression algorithms address this problem.

There are different approaches to noise suppression. They are engineered taking advantage of different characteristics of noise and speech; i.e. how the noise spectrum or envelope modulation of the noisy part of the signal is different from that of the part of the signal that contains speech. The signal properties selected for identifying noise generally determine the nature of the suppression algorithm.

Various algorithms have been proposed and exist for single microphone noise suppression. The general idea is to determine the signal features that make noise different from speech and then design filters that suppress the noise power while affecting as little as possible the speech signal. The general approach is illustrated in figure 2.1.

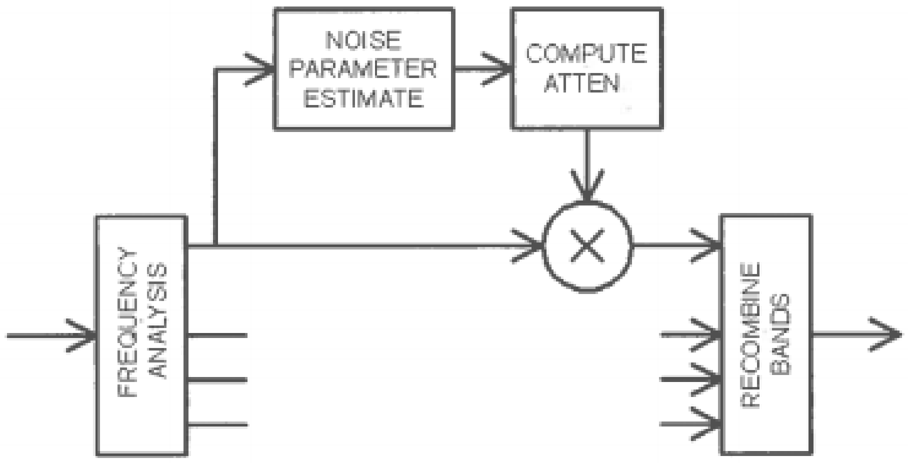


Figure 2.1: Block diagram of a generic multichannel single-microphone noise-suppression system.

The single-microphone noise-suppression algorithms are based on the gain reduction for the signal components estimated to be noise. The suppression can be done in the time domain by:

1 - Reducing the amplitude of low-level sounds (Low level Expansion, Envelope Valley Tracking),

2 - Reducing the signal bandwidth (Adaptive Low-pass Filter, Adaptive High-pass Filter and Zeta Noise Blocker),

or it can be in the envelope modulation domain utilising Envelope Modulation Filters to restrict the envelope modulation rate to the 2Hz - 20 Hz range (which we saw above that is typical of speech). The best any of these approaches could do in improving speech intelligibility was equivalent to about a 1-dB reduction in SNR. Larger benefits for the processed speech were reported by listeners showing that the actual benefit may not be in improving intelligibility, but in reducing the users' listening effort; listeners may still understand the same (thus making the same mistakes, which keeps intelligibility the same), but they understand with less effort.

For instance, the "low level expansion" method (in which low-level signal components are expanded downward) combined with a high-level compression around a chosen knee point, implemented in either broadband or multi band mode, originates from the assumption that speech is more intense than noise: high-level signal components are expected to be speech and low-level components are expected to be noise. However, low-level speech content can be mistaken as noise and be suppressed, this only becoming even worse at poor SNRs situations. It is also notable that low-level expansion has been used as a simulation of hearing loss (Duchnowski and Zurek, 1995), so the level of the speech relative to that of the expansion knee-point is critical.

Furthermore, two additional techniques, Spectral subtraction and Wiener filters, are of high importance and are presented as follows with a separate, more detailed description.

Spectral subtraction

In Spectral Subtraction it is assumed that clean speech is contaminated by additive noise. Its main goal is to estimate the noise spectrum, and then subtract it from the noisy speech spectrum to get an improved estimate of the original clean speech spectrum. The estimated speech spectrum is then used to reconstruct an enhanced signal waveform.

Ideally we would just estimate the noise signal and subtract it from the noisy speech signal to reconstruct the clean speech. But separating the noise from the noisy speech in this way requires access to the original noisy signal, and the noisy signal is generally not available. What is available is the noisy speech. So one has to first estimate the statistics of the noise from observations of the noisy speech signal, then try to produce an enhanced signal which should have statistics matching ones of the original clean speech signal.

Spectral subtraction is implemented on the short-time spectrum of the noisy speech. The incoming noisy signal is split into blocks. Each block is windowed and the short-time FFT is computed from the windowed data sequence. The magnitude spectrum of the noise is estimated, and the noise-suppression gain as a function of frequency is computed from the magnitude spectrum of the noisy signal and the estimated noise magnitude spectrum. The gain function is applied to the noisy spectrum to give the estimated clean speech spectrum, and the modified spectrum is inverse transformed to give the time waveform of the enhanced signal. The signal blocks have a 50 percent overlap; the processed waveform is recombined using the overlap-add procedure. The enhanced signal has a modified envelope, but retains its initial phase.

The estimated noise parameters are used to adjust the gain in each of the analysis frequency bands. In spectral subtraction we start from a distribution of the noisy speech magnitude coming from samples taken over a time interval, and then adaptively adjusting the gain in each of the frequency bands so that the distribution of the magnitude of the processed samples is as close as possible to that of the clean speech.

Spectral subtraction requires an accurate estimate of the noise power in each frequency band. There are two general approaches to noise estimation; one is called a Voice Activity Detector (VAD) and operates by seeking for signal segments containing voiced speech. The noise power estimate is held constant during the voiced speech segments and updated during the non-speech segments. The approach to this, used by most authors reporting on noise reduction - as explained by Marzinzik and Kollmeier (2002) - is to detect the absence of speech in the input signal. In the same paper it is shown that in connection with the Ephraim–Malah noise reduction scheme (being implemented in this very project), the speech pause detection performance can be further increased by using the noise-reduced signal instead of the noisy signal as input for the speech pause decision unit.

The other approach to noise estimation has concentrated on continuously updating the noise power estimate without using a VAD. These newer approaches are also attractive for hearing aids as they require less computation power compared to VAD. Valley Detection, Minima Statistics and Histogram are three known and implemented algorithms for estimating power without VAD.

What is being ignored by the mathematical approach of the algorithms is that the enhanced signals are going to be listened by humans. And gain reductions lowering the noise behind the audible threshold are often with by-products: the fluctuation of the gain from block to block produces artifacts that are audible, known as the "musical noise". So, the maximum amount of signal attenuation must be limited and set so that it just renders the noise inaudible and less obstructive minimizing the generation of processing artifacts.

Models of noise audibility have been implemented for a controlled limiting of the signal attenuation (Azirani et al., 1995; Tsoukalas et al., 1997; Virag, 1999). The auditory masking for the noisy speech signal is then computed using the masking model of Johnston (1988). If the user is hearing impaired, the impaired auditory threshold is taken into account (Natarajan et al., 2005).

Wiener filter

The Wiener filter (Wiener, 1949) is one of the oldest techniques for suppressing noise in a noisy signal. The use of the Wiener filter in suppressing the noise inherent in a speech signal requires separate estimates of the speech power and the noise power and it is also assumed that the speech and noise signals are stationary. A signal is said to be stationary if its statistics do not vary over time. Of course this is not true in the real world. However, if we take a sufficiently small time segment of the signal (from 10 to 30ms) we can assume that, within this segment, the speech content remains almost stationary or at least sufficiently stationary to be able to apply the Wiener filter on it.

In practice this means that the Wiener filter can be designed for the average speech and noise, but cannot take into account the speech or noise fluctuations.

The block diagram for the Wiener filter is illustrated in figure 2.2.

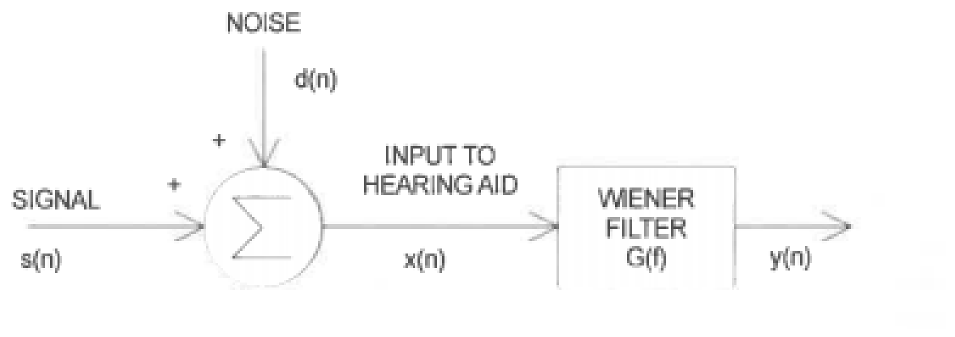


Figure 2.2: Block diagram showing a Wiener filter used to process the input of a hearing aid.

The speech signal $s(n)$ is "corrupted" by additive noise $d(n)$ to generate the input signal $x(n) = s(n) + d(n)$. Then we apply the Wiener filter $G(f)$ to this noisy signal and gather its output $y(n)$. The criterion for the design of the Wiener filter is to minimize the mean-squared error between the clean input and the filtered output:

$$\epsilon = \sum_n [s(n) - y(n)]^2 \tag{2.1}$$

The Wiener filter thus filters the noisy speech to create the closest match possible to the clean speech. The solution to the filter design problem in the continuous frequency domain is:

$$G(f) = \frac{\mid S(f) \mid^2}{\mid S(f) \mid^2 + \mid D(f) \mid^2} \tag{2.2}$$

where $S(f)$ is the long-term average speech spectrum and $D(f)$ is the long-term average noise spectrum. But in practice the speech and noise spectra are not available, so we use the noisy signal spectrum $X(f)$ and the estimated noise spectrum $N(f)$ to get an approximation of the Wiener filter equation:

$$G(f) \approx \frac{\mid X(f) \mid^2 - \mid N(f) \mid^2}{\mid X(f) \mid^2} \tag{2.3}$$

The Wiener filter maximizes the signal-to-noise ratio (SNR) of the noisy speech averaged over the entire signal. The Wiener filter can be effective if the noise is stationary and concentrated in narrow frequency regions or in regions that do not contain much speech energy (Lim, 1986). In most problems of interest, however, speech and noise both fluctuate in amplitude and the noise spectrum substantially overlaps the speech spectrum. Signal processing for these conditions requires a Wiener filter that varies over time to reflect the effect of the noise on different speech sounds (Levitt et al., 1993).

If we replace $X(f)$ and $N(f)$ by $X(k,m)$ and $N(k,m)$ respectively so that changes in the noise level can be tracked, we can create an adaptive Wiener filter whoes gain formula is

$$G_W(k,m) = \frac{\mid X(k,m) \mid^2 - \mid N(k,m) \mid^2}{\mid X(k,m) \mid^2} = 1 - \frac{\mid N(k,m) \mid^2}{\mid X(k,m) \mid^2}, \tag{2.4}$$

where m and k are the corresponding mth block of the kth frequency band once the signal is segmented into blocks.

There is also the power spectral subtraction approach (Lim and Oppenheim, 1979), which assumes that the noise and the clean speech signal are uncorrelated, which expresses the gain as the square root of the Wiener filter gain:

$$G_P(k,m) = \sqrt{1 - \frac{\mid N(k,m) \mid^2}{\mid X(k,m) \mid^2}} \tag{2.5}$$

while in the magnitude spectral subtraction (Boll, 1979) the squared expressions are missing from the gain rule:

$$G_M(k,m) = 1 - \frac{\mid N(k,m) \mid^2}{\mid X(k,m) \mid^2} \tag{2.6}$$

All of the above can be effectively expressed (for $\alpha = 1$ and $\beta = 0$) with the general equation (Berouti et al., 1979; Virag, 1999):

$$G(k,m) = \begin{cases} [1 - \alpha \frac{|N(k,m)|^\gamma}{|X(k,m)|^\gamma}]^{\frac{1}{\delta}} \text{ if } \frac{|N(k,m)|^\gamma}{|X(k,m)|^\gamma} < \frac{1}{\alpha+\beta} \\ [\beta \frac{|N(k,m)|^\gamma}{|X(k,m)|^\gamma}]^{\frac{1}{\delta}} \text{ otherwise} \end{cases} \tag{2.7}$$

The "over subtraction factor" α increases the signal attenuation at poor SNRs, further reducing the amplitude of the background noise when no speech is present. The "attenuation floor" β limits the maximum attenuation that the algorithm produces, reducing the gain fluctuations which can contribute to "musical noise" at low signal levels.

This last equation leads us to the Ephraim-Malah algorithm.

Ephraim-Malah algorithm

For the Ephraim and Malah approach to spectral subtraction it is assumed that the noise samples and the speech samples come from separate and independent Gaussian distributions. The gain is set close to 0dB if a sample appears to belong to the speech's Gaussian; but if a sample appears to come from the noise Gaussian distribution the gain will be reduced (Ephraim and Malah, 1984; McAulay and Malpass, 1980; Wolfe and Godsill, 2001). The gain reduction rule that results in this way depends on the SNR, and is a weighted linear combination the power spectral subtraction and Wiener filter gain rules. For a block m of a frequency band k,

$$\varepsilon(k,m) = \alpha \frac{\mid \hat{S}(k,m-1) \mid^2}{\mid N(k,m-1) \mid^2} + (1-\alpha) max[\frac{\mid X(k,m) \mid^2}{\mid N(k,m) \mid^2} - 1] \tag{2.8}$$

This is a combination of two terms, the a priori SNR and the a posteriori SNR. The a posteriori SNR uses the noisy signal measured values and it is expressed as

$$\varepsilon_{post}(k,m) = \frac{\mid X(k,m) \mid^2}{\mid N(k,m) \mid^2} - 1 \tag{2.9}$$

The a priori SNR is given by the formula

$$\varepsilon_{prior}(k,m) = \frac{E[\mid S(k,m) \mid^2]}{E[\mid D(k,m) \mid^2]} \tag{2.10}$$

Here, $E[\dots]$ is the expectation operator, (typically implemented as a long-term average), $S(k,m)$ the clean speech spectrum and $D(k,m)$ the noise spectrum. In practice, the a priori SNR is not known, so in the algorithm it is approximated by using the speech signal estimate and the noise power estimate from the previous block.

The factor α in the formula controls the trade-off between the noise reduction and the signal distortion regulating how much each part is involved in the estimation. A small α makes the algorithm more sensitive to the signal level changes but produces more artifacts, whilst a high α smooths it and makes the algorithm more relaxed but having fewer by-products.

At a good SNR, the SNR is similar to that used in spectral subtraction; it depends primarily on the instantaneous signal level and the estimated average noise level. At poor SNRs, however, the SNR calculation averages the instantaneous SNR value over many signal blocks, thus the SNR estimate and the resultant suppression gain are being less aggressive. The smoothed gain has reduced fluctuations compared to classical spectral subtraction. One can see the effect of the averaging on the estimated SNR in figure 2.3.

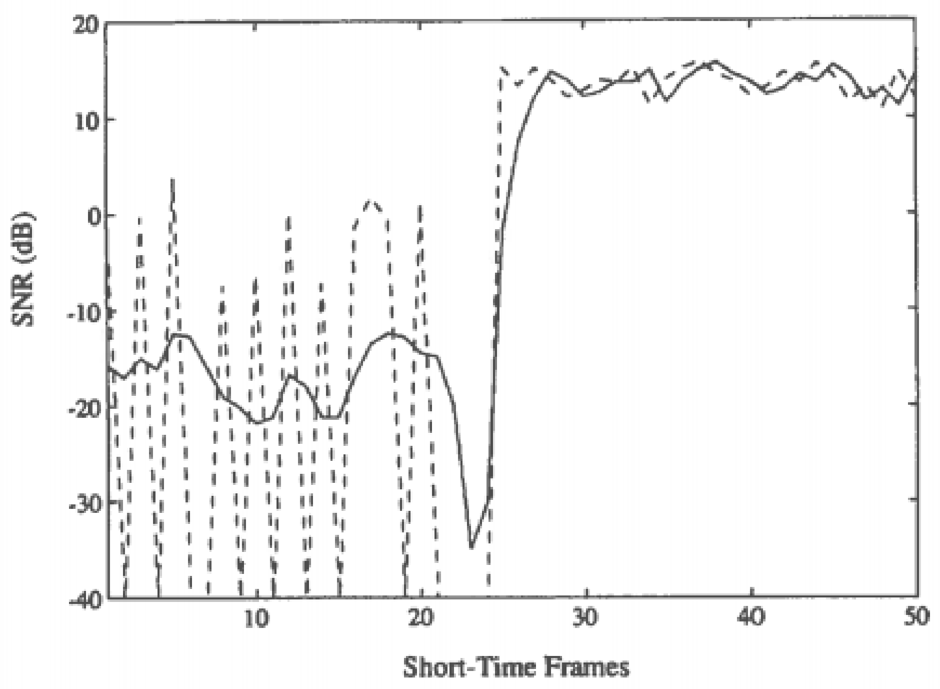


Figure 2.3: Estimated SNRs in successive signal blocks (denoted as short-time frames) for the Ephraim and Malah (1 984) noise suppression algorithm. The dashed line gives the SNR computed from the instantaneous signal envelope samples, and the solid line gives the averaged SNR computed using the algorithm. For the first 25 blocks the signal contains only noise. For the next 25 blocks speech is present at a SNR of 15 dB (from Cappé, 1994).

In this graph one can see that during the noise-only portion of the signal the averaged SNR is much "smoother" (less fluctuating) than the instantaneous SNR. Once the speech starts, however, the amount of averaging is reduced and the estimated SNR resembles the instantaneous SNR used in the classical spectral subtraction implementation.

This is a "decision-directed approach" facilitated by the Ephraim and Malah's SNR estimator and it is creatively used in the Scalart's et al. (1996) noise suppression algorithm.

Noise and speech signal differences

There are substantial differences in the properties of speech and noise signals. Noise in the context of hearing aid audio is considered to be any unwanted sound that comes from the environment and can interfere with the perception of speech. This noise consists of "noises" that exist in everyday listening situations.

The temporal behaviour of speech and noise is a main point of difference. The speech signal has relatively high dynamic level fluctuations compared to those of a Gaussian noise signal. In terms of power, Gaussian noise, unlike speech signal, is stationary: its statistics do not change over time so measurements at various time intervals will be similar. For common, everyday life noises (like multi-talker babble or cars passing by) their behaviour lies between the Gaussian and speech: they present more fluctuations than the Gaussian but not as clearly pronounced as those of speech. So these are non-stationary signals but do not present the "agility" of speech.

From the modulation spectrum (which is the Fourier transform of the signal envelope samples) of the signals, one can see that the modulation frequency range of the speech is from 2 to 20Hz (for -10 to $+10$dB fluctuations) while the Gaussian noise stays basically invariant around a level (eg -20dB in figure 2.4)

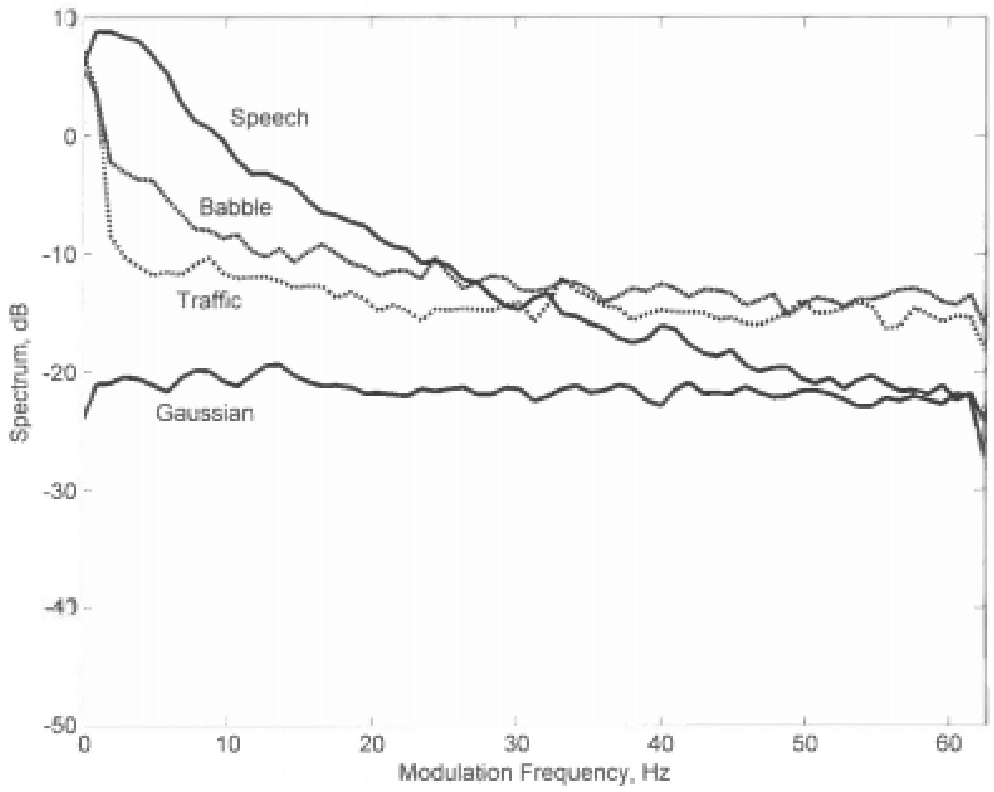


Figure 2.4: Average modulation spectra for the four signals of Figures 9-2 through 9-5. Each plot represents the average of 30 sec of data sampled at 16 kHz.

The procedure

We will be using the algorithm proposed by P. Scalart et al. (1996) for enhancing the noisy signals; a MATLAB implementation of Scalart's et al. algorithm (by Esfandiar Zavarehi [14]) has been studied, analysed (for learning from it) and implemented for the practical implementation of the noise suppression in noisy signals.

Feeding the unprocessed signal back to its enhanced version has been performed using simple linear summation of the two signals: Predetermined quantities of the original noisy signal were introduced (mixed in) to the enhanced signal, while equal quantities of the enhanced signal were subtracted to keep the balance. Then the summed results were being fed to the STOI algorithm for an evaluation of intelligibility. A simple MATLAB script has been created to automate this procedure.

Apart from Esfandiar Zavarehei's MATLAB script we also found and evaluated Pascal Scalart's MATLAB script on the Plapous et al. 2006 proposal for a "two step noise reduction" (TSNR) technique. This is also a Wiener filter gain rule based on a-priori SNR tracking, using the decision-directed method. We screened both scripts using various noisy speech parts and found that both scripts produced results; but Scalart's implementation was introducing audibly more processing artifacts with no significant (if any at all) trade-off in noise removal. This is why we decided to use Zavarehei's script and reserve Scalart's script for further investigation. It is worth saying though that Scalart's script on the Plapous et al. (2006) takes into consideration the harmonic distortion introduced by all the classic short time noise reduction methods and implements a method called "harmonic regeneration noise reduction" (HRNR) to overcome the problem. A refined a-priori SNR is used to compute a spectral gain that will preserve the speech harmonics. The algorithm actually produces two results, a TSNR and an HNNR version.

About the MMSE Noise Suppression Algorithm

For the scope of this project, a MATLAB® realization (by E. Zavarehei, 2005) of P. Scalart's et al. (1996) MMSE noise estimation algorithm is used to suppress the background noise inherent in our speech signals.

The assumption for many single microphone noise reduction techniques is that the spectral magnitude is mainly important for speech intelligibility and quality - rather than the phase of a noisy speech signal. In these designs, the noisy speech is first windowed and then transformed in the frequency domain. The enhanced spectral magnitude is evaluated on each frequency according to a short-time suppression rule. Then the enhanced speech signal is recovered by inverse transforming this spectral magnitude estimation combined with the phase of the noisy speech signal.

Various methods have been proposed for the evaluation of this short-time suppression factor. Its value is adjusted individually on each frequency as a function of the local SNR estimation. Such methods include power spectral subtraction (Boll, 1979), Wiener filtering (Lim, Oppenheim, 1979), soft-decision estimation (McAuley, Malpass, 1980) and Minimum Mean Square Error (or MMSE) estimation (Ephraim, Malah, 1984).

Pascal Scalart's algorithm is an implementation of the MMSE filter; his team's work addresses the issue of an optimal choice for this short-time suppression rule so that one can obtain the best results (in terms of speech quality and intelligibility, nature of residual noise and amount of noise reduction).

In Scalart's experiments the speech was corrupted by "...noise recorded in car on a highway at 120 km/h speed and added to a clean speech signal recorded in a stopped car to obtain a noisy signal..." [11]. The signals were "Hanning" windowed (16ms window length) and their FFTs (Fast Fourier Transforms) taken on 256 points (Fe = 8kHz). Additionally, the noise power spectral density must be evaluated during non-speech activity periods with a first-order recursive (IIR) filter (time constant 140ms) - a constraint added to the estimated time-variant impulse response of the noise reduction filter for conformity with the linear convolution operation.

By incorporating the Ephraim and Malah "decision-directed approach" for "a priori" SNR estimation, Scalart's team has been able to have the practical implementation of the Wiener and MMSE suppression rules that is being used in our work.

Chapter 3

STOI

The Short-Time Objective Intelligibility (STOI) algorithm is an objective, machine-driven intelligibility measure created out of interest for providing a test to check intelligibility for speech processing applications as a low-cost and time saving alternative to doing human listening tests. As it is the case that many times when working with speech processing systems, a certain amount of degradations and unwanted auditory artefacts are introduced and tests are done to determine the perceptual consequences of these modifications, one of them being intelligibility.

As opposed to other intelligibility measures such as the Speech Intelligibility Index (SII) which are meant for simple linear degradations in speech processing, STOI was designed to work better with time-frequency varying gain functions which are used in noise-reduction algorithms.

The algorithm is a function of clean x and processed y speech based on an intermediate measure of short time time-frequency regions, where firstly the TF-representation is obtained by segmenting the signals into overlapping, Hanning windowed frames which are then Fourier transformed (see figure 3.1 on the following page). Then, 15 one-third octave bands are grouped to DFT-bins and analysed. If $x(k, m)$ denotes the kth DFT-bin of the mth frame of the clean signal, then the norm of the jth one-third octave band, or TF-unit, is defined as:

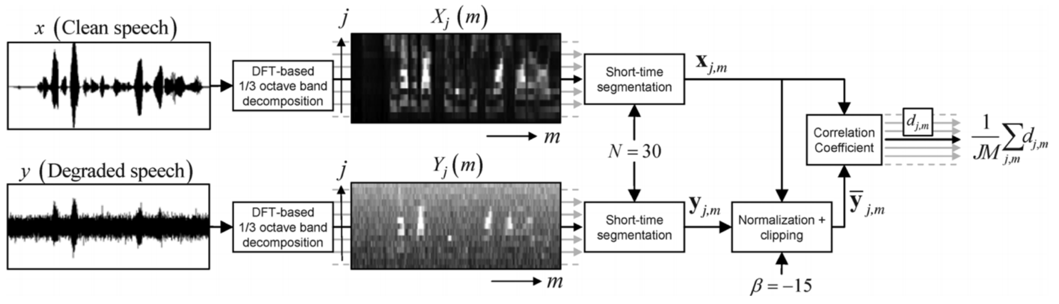


Figure 3.1: Block diagram showing the different steps followed in STOI

$$X_j(m) = \sqrt{\sum_{k=k_1(j)}^{k_2(j)-1} \mid \hat{x}(k,m) \mid^2} \tag{3.1}$$

where k_1 and k_2 denote the one-third octave band edges rounded to the nearest DFT-bin. The TF-representation of the processed speech is done in the same way and is denoted as $Y_j(m)$. At this point, we rescale $Y_j(m)$ by multiplying it by a factor α:

$$\alpha = \frac{\sum_n X_j^2(n)}{\sum_n Y_j^2(n)} \tag{3.2}$$

Where $n \in M$ with $M = [m - N + 1, m - N + 2, \ldots, m - 1, m]$. This normalization procedure is applied so that the energy of $Y_j(n)$ equals that one of the clean speech within that TF-region. Then $\alpha Y(n)$ is clipped to lower bound the signal-to-distortion ratio (SDR), which may be defined as:

$$SDR_j(n) = 10 \log_{10} \frac{X_j^2(n)}{\alpha Y(n) - X_j^2(n)} \tag{3.3}$$

Therefore,

$$Y' = max(min(\alpha Y, X + 10^{-\beta/20} X), X - 10^{-\beta/20} X) \tag{3.4}$$

where Y' represents the TF-unit once normalized and clipped and β denotes the lower SDR bound. With all of this in place , we can compute the intermediate intelligibility measure as follows:

$$d_j(m) = \frac{\sum_n [X_j(n) - \frac{1}{N}\sum_l X_j(l)][Y_j'(n) - \frac{1}{N}\sum_l Y_j'(l)]}{\sqrt{\sum_n [X_j(n) - \frac{1}{N}\sum_l X_j(l)]^2 \sum_n [Y_j'(n) - \frac{1}{N}\sum_l Y_j'(l)]^2}} \tag{3.5}$$

where $l \in M$. The objective intelligibility measure is given then by the average of the measures of the last equations over all bands and frames:

$$d = \frac{1}{JM} \sum_{j,m} d_j(m) \tag{3.6}$$

where M represents the total number of frames and J the number of one-third octave bands. Experiments pointed out that the optimal choices for β and N are -15 and 30 respectively.

This particular algorithm was used in our research to provide preliminary results as to how the noise suppression processing of the noisy speech was affecting intelligibility, giving us insight into how to prepare our audio for testing considering its results. Since this type of intelligibility test is objective, it can give us an idea of the degree to which the processed signal compares to the original signal, but cannot give us insight into how intelligibility is perceived by people.

The MATLAB implementation of both the Wiener filter and the STOI calculations are included in the Appendix. The results will be presented and discussed on Chapter 6.

Chapter 4

Test material

Recording

To be able to double check the answers of the participants, we decided to record the experiment session. Recording was done in both audio and video formats. Sound recording was performed with the omni directional microphone available in the lab. The concrete model is the Behringer ECM 8000 (figure 4.1), which ensures a sufficiently flat frequency response according to our needs.

Figure 4.1: Behringer ECM 8000

Testing

The headphones used by the participants were the Urbanears Plattan, a low impedance set shown in figure 4.2 (32 ohm). The rest of the specifications are: $20 - 20k$ Hz frequency response, 112 dB of sensitivity. The headphones were plugged into an audio interface, namely a M-Audio Fasttrack Pro.

Figure 4.2: Urbanears Plattan

The audio interface was connected to a computer, in this case a MacBook Pro from the year 2014. The excerpts had a duration of approximately between 2 and 3 seconds, and were prepared properly with MATLAB and then brought to Logic Pro X for reproduction (figure 4.3).

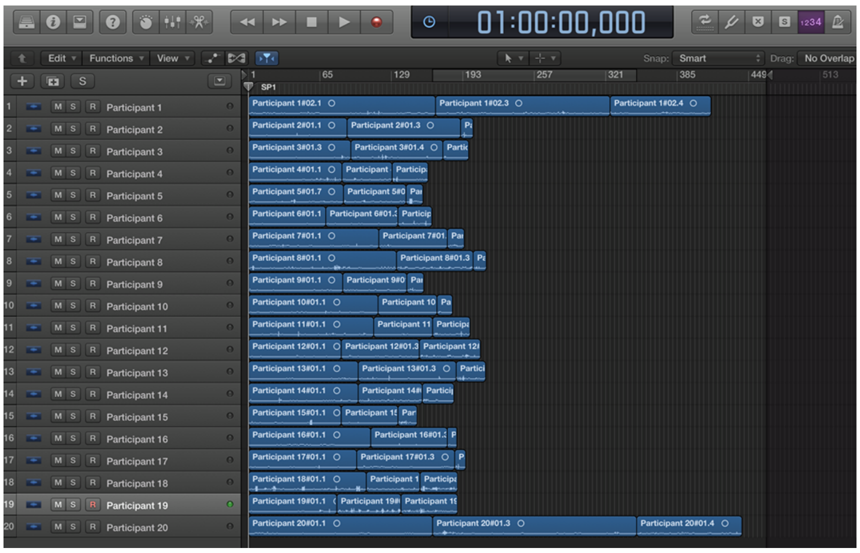


Figure 4.3: Logic Pro X interface

Chapter 5

Intelligibility test

In this chapter, an extensive description of the experimental part of the project will be presented.

Overview

Our initial hypothesis was that adding noisy signal to an already processed (by noise suppression) could improve its intelligibility. Following this reasoning, we carried out an experiment to test this. Hence, the purpose of the experiment was to see which the optimal amount of noisy signal to be added a posteriori that gives us maximum intelligibility would be, but keeping it at reasonable SNR's, this is, keeping the noise levels low. In this way, we could achieve reasonable levels of quality and intelligibility at the same time.

Therefore, we decided to test five different kinds of signals: an unprocessed signal (original noisy signal), a processed one (original signal Wiener filtered) and three different more signals. These last three were created mixing back our processed signal with the initial noisy one. The chosen ratios were 15, 30 and 45 percent. For instance, the first one of the three would be a signal with 85 and 15 percent of processed and unprocessed (noisy) signal respectively.

The reason for this choice was not easy for us. As we saw in previous chapters, the STOI has shown that intelligibility is positively correlated with percentage of added noisy signal, proving our initial hypothesis to be, at least, coherent. However, the optimal percentage of mixing appeared to be, in the majority of the cases, at non-desired SNR values. Therefore, our first criterion was to choose ratios below the 50 percent ratio for testing. Another thing we took into account was the threshold of distinction between to different mixing ratios. This is important because we wanted the participant to be able to tell the difference between two different mixing ratios. That said, and for the sake of the experiment, we agreed upon testing three different kinds of mixing, all of them spaced by 15 percent. The choice was of the different mixing ratios was done considering the criteria mentioned above as well as taking into account the results of STOI.

For creating the excerpts at the ratios mentioned, we took the audio samples from the Noizeus database (http://ecs.utdallas.edu/loizou/speech/noizeus/). This database consists of 8 different groups with 30 different sentences in each group corresponding to 8 different types of background noise. Each sentence consists of a short set of words in the form of a sentence such us 'The lazy cow lay on the cool grass'. All of the samples are spoken in English, some by a male voice and others by a female voice. We chose half of each gender voice for equity.

For each of the noises we had 4 different SNR's: 0, 5, 10 and 15dB. Here we made two choices. The first concerns the SNR. After hearing all the different SNR's, we agreed upon using the best one (15dB), for the others were of a too poor quality. The second choice concerns background noise. Since our main interest was to compare different mixing ratios, using a single background noise is enough for our interests. According to the algorithm we used, we wanted to look for a fairly stationary background noise. At the same time, we wanted to work under realistic conditions, so we avoided choosing ideal noises such as Gaussian. Therefore, car noise seemed to be a balanced choice that met the requirements.

Later on, we made a choice on which sentences to use in the experiment. We chose 10 different sentences among the 30 which had medium difficulty yet overall not too difficult so as to avoid the participants not being able to understand anything whatsoever. The five mixing ratios were tested twice, each time with a different sentence to avoid bias in the answers. A total of 20 participants performed the test, thus providing 40 different measures for each mixing ratio. To avoid getting biased results based on using the same order of reproduction of the sentences for all the participants, both the sentences and the mixing ratios were randomised and gathered into different reproduction lists.

Moreover, we created a method for detecting false positives. This was made to avoid participants guessing the answer, as opposed to actually understanding the speech. We did this by creating a scale on how well a participant thaught he or she understood what they heard. This scale was created from two established intelligibility scales [13], [1].

Test procedure

First of all, we selected the place where we wanted to carry out our experiment. The sound lab was the perfect space:a quiet environment without distractions and with all the necessary equipment available.

Once we selected the place, we prepared the set up and called upon participants. They then were asked to fill in a quick consent form (to ask for permission to be recorded) along with some simple questions which included: name (optional), nationality (optional), hearing problems (yes or no; mandatory), English level (intermediate, advanced, native; mandatory). Asking about whether the participant suffers from any kind of hearing problem seems quite obvious in an experiment like this, but asking about the English level may not seem that obvious. We had reasons to believe that the language played a non-disposable role in this experiment. We will discuss about this matter after we present the results of the experiment.

Then participants were asked to sit down and listen to the explanation of the experiment. Its structure is fairly simple: each participant was asked to hear a sequence of speech phrases with headphones and to repeat what they heard. Secondly, they were asked to rate how well they thought they understood what they heard according to a scale we provided.

Before doing the actual test, the participants were granted with a training session to get familiar with how the test was going to be. In this training session, participants were asked to hear to two different sentences different from the ones used in the test in order to adjust the volume of the system and rehearse on giving the answers.

Following the steps described above, the test was done. The gathering of the answers was done by the note taker, who had the answer sheet with the transcription of the sentences. The questionnaire and the answer sheet are provided in the Appendix.

Chapter 6

Results and analysis

The first part of the results will include STOI measures and the second part concerns the experimental test.

We performed the STOI on several sentences and in figure (ref) a sample can be found.

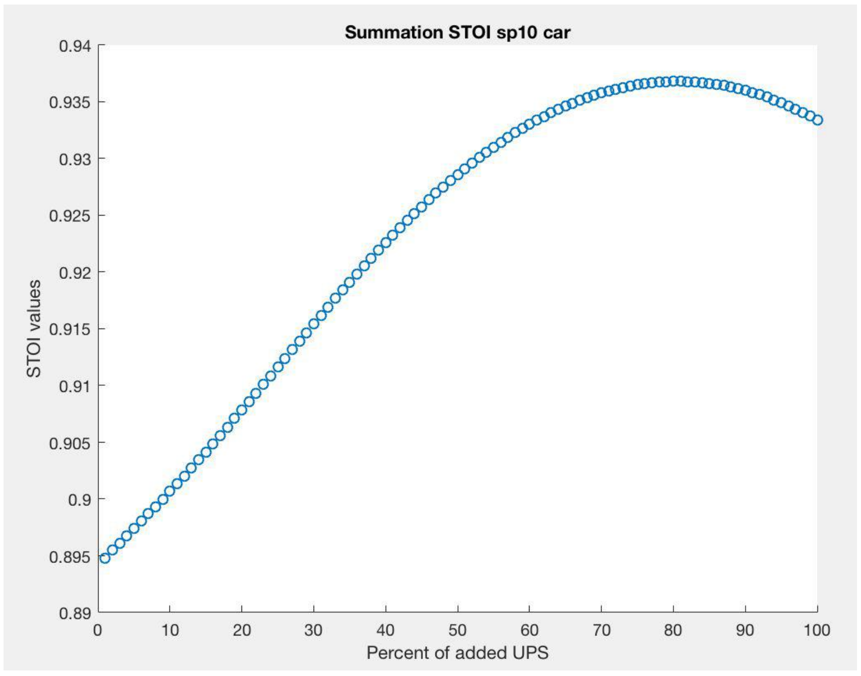
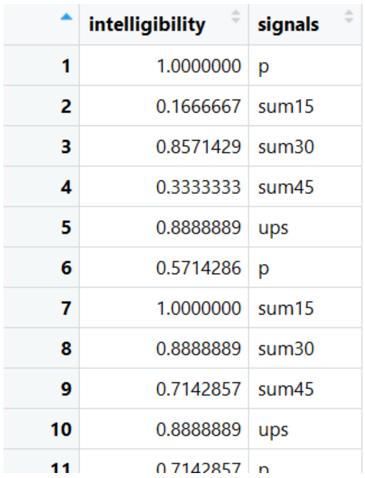


Figure 6.1: STOI scores for one of the sentences used in the test varying the amount of noisy unprocessed signal (UPS) from 0 to 100 percent in the tested signal

The usage of STOI was initially intended to help to decide upon the optimal choice for the mixing ratios. The sample above represents how intelligible the sentences were according to STOI depending on how much of the noisy signal was added. The maximum point in the graph was in most of the cases around 70 percent. As explained in the last chapter, we wanted to keep SNRs high enough, so in the end we only used the results of the STOI to check what the tendency of added noisy signal and intelligibility was. That is, to check whether adding more noisy signal increased intelligibility or not. The results showed that the correlation of these two variables was positive (at least until it reaches the maximum), as can be seen in the figure above.

In the following chapter, all the statistical calculations were done with R Studio statistical toolbox. In the results, 200 observations taken from 20 participants were obtained in total. The dataset was modelled into a data frame which is an object with rows and columns. The data frame can be seen in figure 6.2.



	intelligibility	signals
1	1.0000000	p
2	0.1666667	sum15
3	0.8571429	sum30
4	0.3333333	sum45
5	0.8888889	ups
6	0.5714286	p
7	1.0000000	sum15
8	0.8888889	sum30
9	0.7142857	sum45
10	0.8888889	ups
11	0.7142857	p

Figure 6.2: Representation of the data frame (segment)

The rows contain the different observations (intelligibility) in the range from 0 to 1 while the columns (signals) contain a factor of five different levels (processed, sum15, sum30, sum45, unprocessed). The results showed that from the five presented speech signals (namely processed, unprocessed and combination of these signals in three predetermined proportions), sum15 and sum45 scored the highest means of intelligibility, as can be seen in figure 6.3.

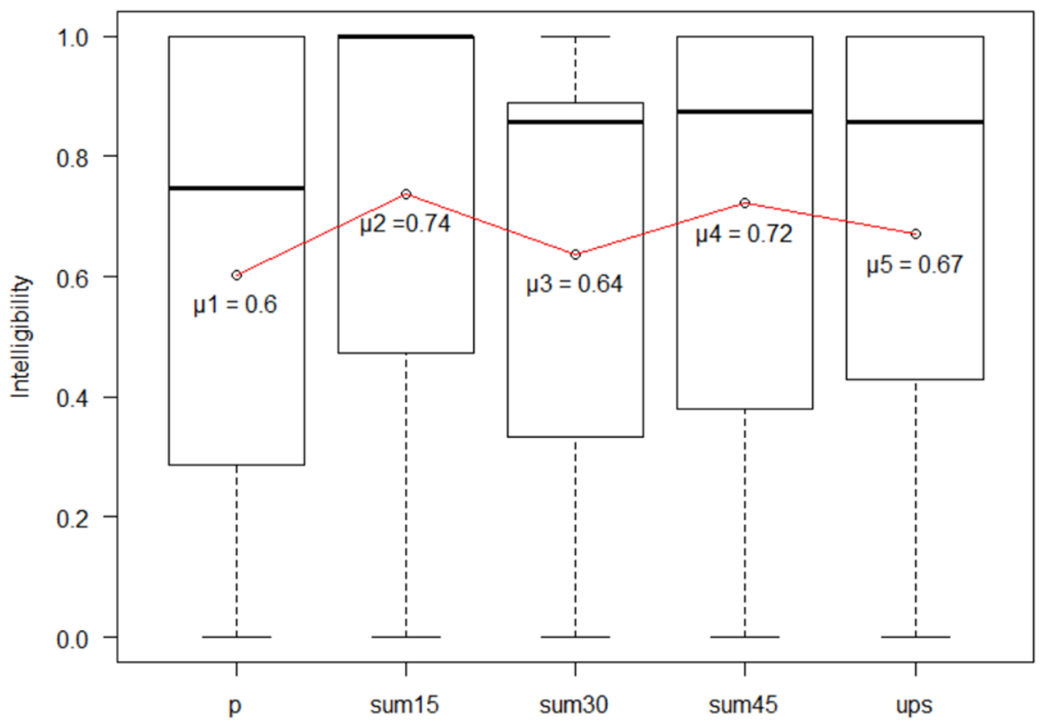


Figure 6.3: Box plot of scores for the audio intelligibility test.

The intelligibility of sum30 was considered to be somewhere between sum15 and sum45, however, the results showed a significantly lower value.

In order to assess significant differences between the intelligibility scores obtained from each level, the listeners' scores are subjected to statistical analysis. This can be done by either ANOVA or t-test. However, t-test applies only when the means of two populations are compared. On the other hand, when using ANOVA(a parametric test), normal distribution and equal variances must be verified. To verify whether the data is normally distributed both Q-Q plot (6.4) analysis and Shapiro-Wilk Test were used.

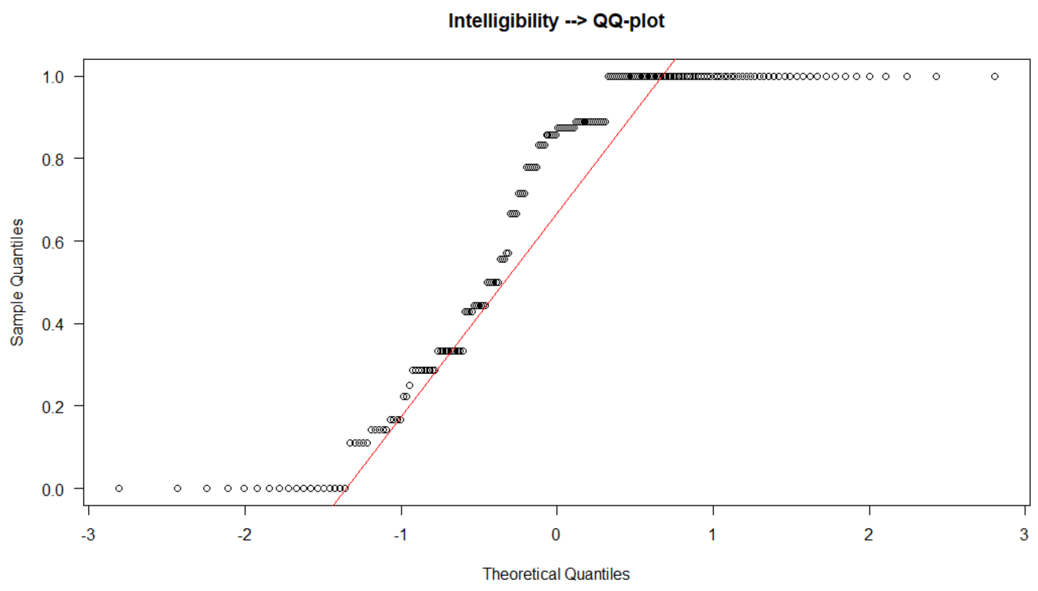
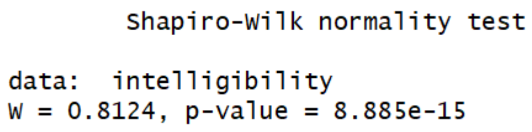


Figure 6.4: Quantile-Quantile Plot for quick check of normality.

If the data is normally distributed, the points in the q-q plot follow a straight diagonal line. Observably, the outcome indicates abnormality. Plotting the data in this way is useful to check for normality at first glance, but it is not an accurate statistical test. However, Shapiro-Wilk Test (see figure 6.5) ensured that the data is not normally distributed.

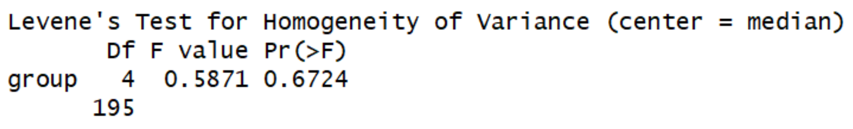


```
                Shapiro-Wilk normality test

data:  intelligibility
W = 0.8124, p-value = 8.885e-15
```

Figure 6.5: Shapiro-Wilks test for normality.

The null hypothesis of Shapiro-Wilk Test states that the sample has a normal distribution. Accordingly, the p-value that results from the test represents the chance that the sample originates from a normal distribution. The low p-value ($p < 0.05$) thus indicates that the sample is not likely to be normally distributed. The Levene's Test (figure 6.6) uses an F-test to test the null hypothesis, which indicates that the variance is equal across levels. The outcome ($p > F - value$) accpets H_0 which indicates the assumption that variance is equal across levels.

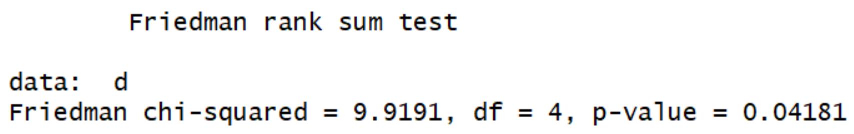


```
Levene's Test for Homogeneity of Variance (center = median)
       Df F value Pr(>F)
group   4  0.5871 0.6724
      195
```

Figure 6.6: Levene's Test for homogeneity of variance.

Since the obtained data is not normally distributed, a non-parametric test called Friedman's Test was used. The purpose of the test is to examine that whether the processed speech signal summed with the unprocessed signal back are significantly different from the processed speech signal.

The results showed that if more unprocessed signal is introduced back to the processed signal, intelligibility is increased compared to the processed signal. To test if this result is statistically significant, Friedman and Post-hoc Test were done. The Friedman test (figure 6.7) determines if there are differences among levels, precisely, whether the groups selected from populations have equal means ($H_0 = \mu_1 = \mu_2 = \mu_3 = \mu_4 = \mu_5$) or not.

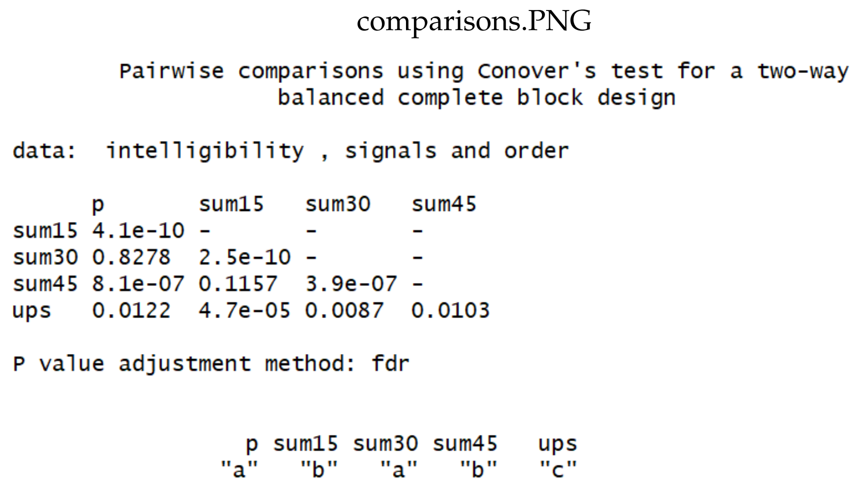


```
        Friedman rank sum test

data:  d
Friedman chi-squared = 9.9191, df = 4, p-value = 0.04181
```

Figure 6.7: Friedman rank sum test in R Studio.

This means that the null hypothesis, that all means of the scores are the same, was rejected. Since the confidence level ($\alpha = 0.05$) is greater than the p-value ($p < \alpha$), there are some significant differences among levels. The outcome of the Friedman test tells whether there are significant differences among the levels or not, but does not tell which levels are different from each other. In order to determine which levels are different from others, post-hoc testing was conducted (figure 6.8):


comparisons.PNG

```
        Pairwise comparisons using Conover's test for a two-way
                    balanced complete block design

data:  intelligibility , signals and order

      p       sum15   sum30   sum45
sum15 4.1e-10 -       -       -
sum30 0.8278  2.5e-10 -       -
sum45 8.1e-07 0.1157  3.9e-07 -
ups   0.0122  4.7e-05 0.0087  0.0103

P value adjustment method: fdr


                          p sum15 sum30 sum45   ups
                        "a"   "b"   "a"   "b"   "c"
```

Figure 6.8: Pairwise comparisons using Conover's test (Post-hoc) with compact letter display, groups sharing a letter are not significantly different.

It performs a two-sample paired sign test on each pair of levels, where each combination is assigned with a p-value. For simplicity and readability, the result was converted into compact letter display. The outcome shows that the two levels (sum15 and sum45) with highest intelligibility scores are not statistically different. Furthermore, level p and sum30 are not significantly different either. Nevertheless, the intelligibility scores of unprocessed signals are recognizably different in nature from the others. Despite sum15 and sum45 being distinct from processed speech signal, the null-hypothesis:

$H_0 =$ Composition of processed signal and unprocessed signal within the range of 15 to 45 percent DOES NOT improve subjective intelligibility, compared to processed signal.

could not be rejected due to the homogeneity of processed and sum30 speech signals. On the other hand, the result showed that the intelligibility is significantly improved with the proportion of 15 percent unprocessed to 85 percent processed speech signal. However, the validity and reliability of the observations could have depended on the English language levels (native, advanced and intermediate) of the participants. Although most of the participants had advanced levels in English, other language levels could have influenced the results. Figure 6.9 represents the language levels among the participants.

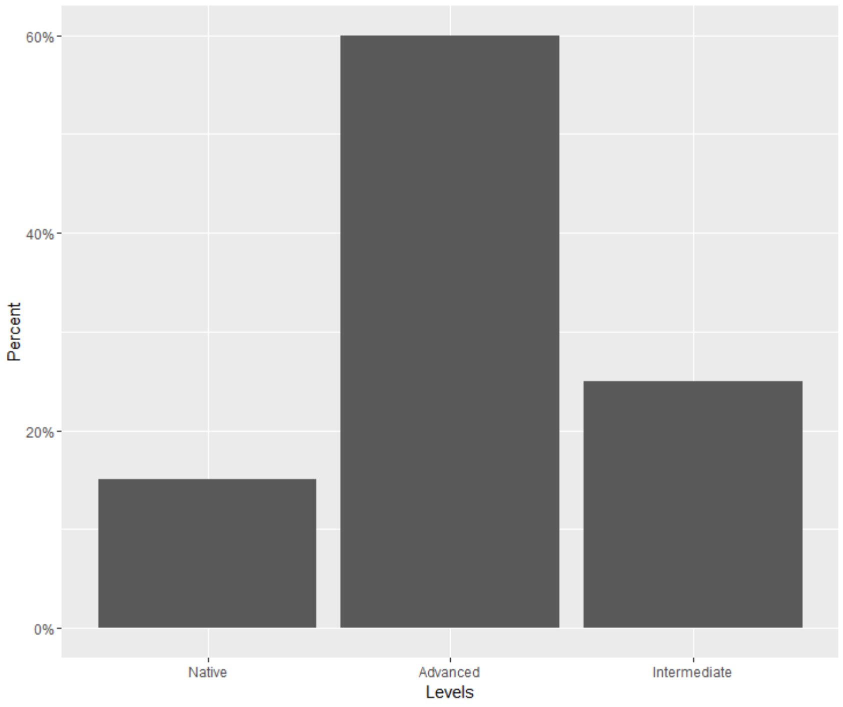


Figure 6.9: English levels of the participants of the test

The participants with higher English level performed better and more accurately than others with lower English skills. This observation is identical for participants with intermediate proficiency but with the opposite effect. The influence of English levels on intelligibility scores can be seen in figure 6.10:

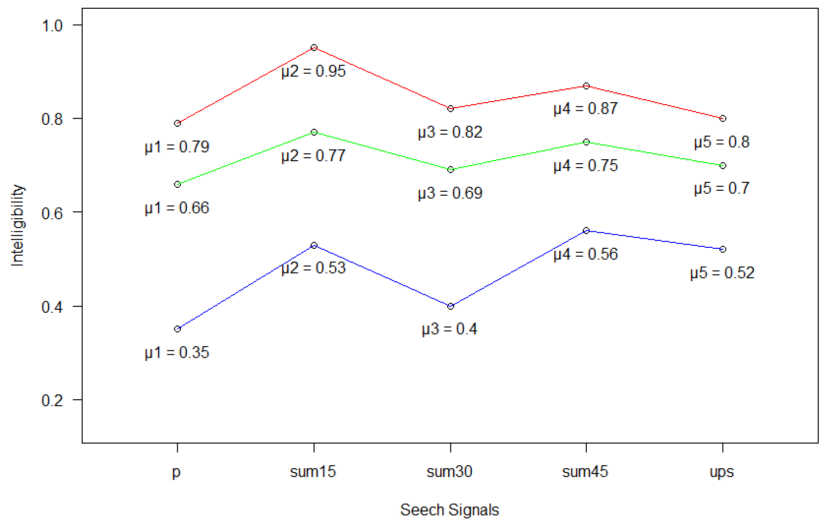


Figure 6.10: Influence of English levels on intelligibility scores. From the top to bottom: Native (red); Advanced (green); Intermediate (blue).

However, the outcome of the separated levels is very similar compared to the analysis done when they are not isolated. Consequently, the hypothesis stating that language proficiency significantly influenced the results can be rejected.

Bibliography

[1] C Bowen. *Table 1: Intelligibility*. `http://www.speech-language-therapy.com/`. 2011.

[2] MATLAB Central. *Wiener Filter*. `https://se.mathworks.com/matlabcentral/fileexchange/24462-wiener-filter-for-noise-reduction-and-speech-enhancement?focused=5123482&tab=function`.

[3] Cunningham Shao Cooke Barker. *An audio-visual corpus for speech perception and automatic speech recognition*. 2005.

[4] Zurek P.M Duchnowski P. *Villchur revisited: Another look at AGC simulation of recruiting hearing loss*. 1995.

[5] Malah Ephraim. *Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator*. 1984.

[6] Jesper Jensen. *A short-time objective intelligibility measure for time-frequency weighted noisy speech*. 2010.

[7] James M. Kates. *Digital Hearing Aids*. 2008.

[8] M. G. Christensen J.B. Boldt M. S. Kavelekalam J. K. Nielsen. *Model Based Speech Enhancement for Intelligibility Improvement in Binaural Hearing Aids*. 2017.

[9] Kollmeier Marzinzik. *Speech pause detection for noise spectrum estimation by tracking power envelope dynamics*. 2002.

[10] World Health Organization. *Deafness and hearing loss*. `http://www.who.int/mediacentre/factsheets/fs300/en/`. 2017.

[11] Chiquito Scalart Filho. *On speech enhancement algorithms based on MMSE estimation*. 1996.

[12] Heudsens Jensen Taal Hendriks. *An algorithm for intelligibility prediction of time-frequency weighted noisy speech*. 2011.

[13] Eochester Institute of Technology. *Intelligibility*. `https://www.ntid.rit.edu/slpros/assessment/speechvoice/training/1`.

[14] Esfandiar Zavarehi. *Wiener Filter - Ephraim-Malah algortihm*. `https://se.mathworks.com/matlabcentral/fileexchange/7673-wiener-filter?focused=5061187&tab=function`.

Appendix A

Appendix A name

Wiener Filter (Scalart's)

```
function output=WienerScalart96(signal,fs,IS)

% output=WIENERSCALART96(signal,fs,IS)
% Wiener filter based on tracking a priori SNR usingDecision-Directed
% method, proposed by Scalart et al 96. In this method it is assumed that
% SNRpost=SNRprior +1. based on this the Wiener Filter can be adapted to a
% model like Ephraims model in which we have a gain function which is a
% function of a priori SNR and a priori SNR is being tracked using Decision
% Directed method.
% Author: Esfandiar Zavarehei
% Created: MAR-05



if (nargin<3 | isstruct(IS))
    IS=.25; %Initial Silence or Noise Only part in seconds
end

W=fix(.025*fs); %Window length is 25 ms

SP=.4; %Shift percentage is 40% (10ms) %Overlap-Add method

works good with this value(.4)

wnd=hamming(W);
```

```
%IGNORE FROM HERE ..............................
if (nargin>=3 & isstruct(IS)) %This option is for

compatibility with another programme

    W=IS.windowsize
    SP=IS.shiftsize/W;
    %nfft=IS.nfft;
    wnd=IS.window;
    if isfield(IS,'IS')
        IS=IS.IS;
    else
        IS=.25;
    end
end
% .....................................UP TO HERE

pre_emph=0;
signal=filter([1 -pre_emph],1,signal);

NIS=fix((IS*fs-W)/(SP*W) +1);%number of initial silence segments

y=segment(signal,W,SP,wnd); % This function chops the signal into frames
Y=fft(y);
YPhase=angle(Y(1:fix(end/2)+1,:)); %Noisy Speech Phase
Y=abs(Y(1:fix(end/2)+1,:));%Specrogram
numberOfFrames=size(Y,2);
FreqResol=size(Y,1);

N=mean(Y(:,1:NIS)')'; %initial Noise Power Spectrum mean
LambdaD=mean((Y(:,1:NIS)').^2)';%initial Noise Power Spectrum variance
alpha=.99; %used in smoothing xi

(For Decision Directed method for estimation of A Priori SNR)
NoiseCounter=0;
NoiseLength=9;%This is a smoothing factor for the noise updating
G=ones(size(N));%Initial Gain used in calculation of the new xi
Gamma=G;

X=zeros(size(Y)); % Initialize X (memory allocation)
```

```
h=waitbar(0,'Wait...');

for i=1:numberOfFrames
    %AD and Noise Estimation START

    if i<=NIS % If initial silence ignore VAD
        SpeechFlag=0;
        NoiseCounter=100;
    else % Else Do VAD
        [NoiseFlag, SpeechFlag, NoiseCounter, Dist]=vad(Y(:,i),N,NoiseCounter);

        %Magnitude Spectrum Distance VAD
    end

    if SpeechFlag==0  %If not Speech Update Noise Parameters
        N=(NoiseLength*N+Y(:,i))/(NoiseLength+1); %Update and smooth noise mean
        LambdaD=(NoiseLength*LambdaD+(Y(:,i).^2))./(1+NoiseLength); %Update and

        smooth noise variance
    end
    %VAD and Noise Estimation END

    gammaNew=(Y(:,i).^2)./LambdaD; %A postiriori SNR
    xi=alpha*(G.^2).*Gamma+(1-alpha).*max(gammaNew-1,0); %Decision Directed

    Method for A Priori SNR
    Gamma=gammaNew;

    G=(xi./(xi+1));

    X(:,i)=G.*Y(:,i); %Obtain the new Cleaned value

    waitbar(i/numberOfFrames,h,num2str(fix(100*i/numberOfFrames)));
end

close(h);
output=OverlapAdd2(X,YPhase,W,SP*W); %Overlap-add Synthesis of speech
output=filter(1,[1 -pre_emph],output); %Undo the effect of Pre-emphasis

function ReconstructedSignal=OverlapAdd2(XNEW,yphase,windowLen,ShiftLen);
```

```
%Y=OverlapAdd(X,A,W,S);
%Y is the signal reconstructed signal from its spectrogram. X is a matrix
%with each column being the fft of a segment of signal. A is the phase
%angle of the spectrum which should have the same dimension as X. if it is
%not given the phase angle of X is used which in the case of real values is
%zero (assuming that its the magnitude). W is the window length of time
%domain segments if not given the length is assumed to be twice as long as
%fft window length. S is the shift length of the segmentation process ( for
%example in the case of non overlapping signals it is equal to W and in the
%case of %50 overlap is equal to W/2. if not givven W/2 is used. Y is the
%reconstructed time domain signal.
%Sep-04
%Esfandiar Zavarehei

if nargin<2
    yphase=angle(XNEW);
end
if nargin<3
    windowLen=size(XNEW,1)*2;
end
if nargin<4
    ShiftLen=windowLen/2;
end
if fix(ShiftLen)~=ShiftLen
    ShiftLen=fix(ShiftLen);
    disp('The shift length have to be an integer as it is the number of
samples.')
    disp(['shift length is fixed to ' num2str(ShiftLen)])
end

[FreqRes FrameNum]=size(XNEW);

Spec=XNEW.*exp(j*yphase);

if mod(windowLen,2) %if FreqResol is odd
    Spec=[Spec;flipud(conj(Spec(2:end,:)))];
else
    Spec=[Spec;flipud(conj(Spec(2:end-1,:)))];
end
sig=zeros((FrameNum-1)*ShiftLen+windowLen,1);
```

```
weight=sig;
for i=1:FrameNum
    start=(i-1)*ShiftLen+1;
    spec=Spec(:,i);
    sig(start:start+windowLen-1)=sig(start:start+windowLen-1)+
    real(ifft(spec,windowLen));
end
ReconstructedSignal=sig;

function Seg=segment(signal,W,SP,Window)

% SEGMENT chops a signal to overlapping windowed segments
% A= SEGMENT(X,W,SP,WIN) returns a matrix which its columns are segmented
% and windowed frames of the input one dimentional signal, X. W is the
% number of samples per window, default value W=256. SP is the shift
% percentage, default value SP=0.4. WIN is the window that is multiplied by
% each segment and its length should be W. the default window is hamming
% window.
% 06-Sep-04
% Esfandiar Zavarehei

if nargin<3
    SP=.4;
end
if nargin<2
    W=256;
end
if nargin<4
    Window=hamming(W);
end
Window=Window(:); %make it a column vector

L=length(signal);
SP=fix(W.*SP);
N=fix((L-W)/SP +1); %number of segments

Index=(repmat(1:W,N,1)+repmat((0:(N-1))'*SP,1,W))';
hw=repmat(Window,1,N);
Seg=signal(Index).*hw;

function [NoiseFlag, SpeechFlag, NoiseCounter, Dist]=
vad(signal,noise,NoiseCounter,NoiseMargin,Hangover)

%[NOISEFLAG, SPEECHFLAG, NOISECOUNTER, DIST]=
vad(SIGNAL,NOISE,NOISECOUNTER,NOISEMARGIN,HANGOVER)
%Spectral Distance Voice Activity Detector
%SIGNAL is the the current frames magnitude spectrum which is to labeld as
%noise or speech, NOISE is noise magnitude spectrum template (estimation),
%NOISECOUNTER is the number of imediate previous noise frames, NOISEMARGIN
%(default 3)is the spectral distance threshold. HANGOVER ( default 8 )is
%the number of noise segments after which the SPEECHFLAG is reset (goes to
%zero). NOISEFLAG is set to one if the the segment is labeld as noise
%NOISECOUNTER returns the number of previous noise segments, this value is
%reset (to zero) whenever a speech segment is detected. DIST is the
%spectral distance.
%Saeed Vaseghi
%edited by Esfandiar Zavarehei
%Sep-04

if nargin<4
    NoiseMargin=3;
end
if nargin<5
    Hangover=8;
end
if nargin<3
    NoiseCounter=0;
end

FreqResol=length(signal);

SpectralDist= 20*(log10(signal)-log10(noise));
SpectralDist(find(SpectralDist<0))=0;

Dist=mean(SpectralDist);
if (Dist < NoiseMargin)
    NoiseFlag=1;
    NoiseCounter=NoiseCounter+1;
else
    NoiseFlag=0;
    NoiseCounter=0;
end
```

```
% Detect noise only periods and attenuate the signal
if (NoiseCounter > Hangover)
    SpeechFlag=0;
else
    SpeechFlag=1;
end
```

STOI

```
function d = stoi(x, y, fs_signal)
%   d = stoi(x, y, fs_signal) returns the output of the short-time
%   objective intelligibility (STOI) measure described in [1, 2], where x
%   and y denote the clean and processed speech, respectively, with sample
%   rate fs_signal in Hz. The output d is expected to have a monotonic
%   relation with the subjective speech-intelligibility, where a higher d
%   denotes better intelligible speech. See [1, 2] for more details.
%
%   References:
%      [1] C.H.Taal, R.C.Hendriks, R.Heusdens, J.Jensen 'A Short-Time
%      Objective Intelligibility Measure for Time-Frequency Weighted Noisy
%      Speech', ICASSP 2010, Texas, Dallas.
%
%      [2] C.H.Taal, R.C.Hendriks, R.Heusdens, J.Jensen 'An Algorithm for
%      Intelligibility Prediction of Time-Frequency Weighted Noisy Speech',
%      IEEE Transactions on Audio, Speech, and Language Processing, 2011.
%
%
% Copyright 2009: Delft University of Technology, Signal & Information
% Processing Lab. The software is free for non-commercial use. This program
% comes WITHOUT ANY WARRANTY.
%
%
%
% Updates:
% 2011-04-26 Using the more efficient 'taa_corr' instead of 'corr'

if length(x)~=length(y)
```

```
    error('x and y should have the same length');
end

% initialization
x = x(:);   % clean speech column vector
y = y(:);   % processed speech column vector

fs = 10000;   % sample rate of proposed intelligibility measure
N_frame = 256;   % window support
K = 512;   % FFT size
J = 15;    % Number of 1/3 octave bands
mn = 150;   % Center frequency of first 1/3 octave band in Hz.
H = thirdoct(fs, K, J, mn);   % Get 1/3 octave band matrix
N = 30;    % Number of frames for intermediate intelligibility measure
(Length analysis window)
Beta  = -15;    % lower SDR-bound
dyn_range  = 40;  % speech dynamic range

% resample signals if other samplerate is used than fs
if fs_signal ~= fs
    x = resample(x, fs, fs_signal);
    y  = resample(y, fs, fs_signal);
end

% remove silent frames
[x y] = removeSilentFrames(x, y, dyn_range, N_frame, N_frame/2);

% apply 1/3 octave band TF-decomposition
x_hat = stdft(x, N_frame, N_frame/2, K); % apply short-time DFT
to clean speech
y_hat = stdft(y, N_frame, N_frame/2, K); % apply short-time DFT
to processed speech

x_hat = x_hat(:, 1:(K/2+1)).';   % take clean single-sided spectrum
y_hat = y_hat(:, 1:(K/2+1)).';   % take processed single-sided spectrum

X = zeros(J, size(x_hat, 2));  % init memory for clean speech 1/3 octave
band TF-representation
Y  = zeros(J, size(y_hat, 2));  % init memory for processed speech
1/3 octave
band TF-representation
```

```
for i = 1:size(x_hat, 2)
    X(:, i) = sqrt(H*abs(x_hat(:, i)).^2);  % apply 1/3 octave bands
    as described in Eq.(1) [1]
    Y(:, i) = sqrt(H*abs(y_hat(:, i)).^2);
end

% loop al segments of length N and obtain intermediate intelligibility measure
for all TF-regions
d_interm = zeros(J, length(N:size(X, 2)));   % init memory for intermediate
intelligibility measure
c = (-Beta/20);   % constant for clipping procedure

for m = N:size(X, 2)
    X_seg   = X(:, (-N+1):m);   % region with length N of clean TF-units
    for all j
    Y_seg   = Y(:, (-N+1):m);     % region with length N of processed TF-units
    for all j
    alpha = sqrt(sum(X_seg.^2, 2)./sum(Y_seg.^2, 2));   % obtain scale factor
    for normalizing processed TF-region for all j
    aY_seg  = Y_seg.*repmat(alpha, [1 N]);  % obtain \alpha*Y_j(n) from Eq.(2) [1]
    for j = 1:J
     Y_prime = min(aY_seg(j, :), X_seg(j, :)+X_seg(j, :)*c); %apply clipping
     from Eq.(3)
     d_interm(j, m-N+1)  = taa_corr(X_seg(j, :).', Y_prime(:));  % obtain
     correlation coeffecient from Eq.(4) [1]
    end
end

d = mean(d_interm(:));    % combine all intermediate intelligibility measures
as in Eq.(4) [1]

%%
function  [A cf] = thirdoct(fs, N_fft, numBands, mn)
%   [A CF] = THIRDOCT(FS, N_FFT, NUMBANDS, MN) returns 1/3 octave band matrix
%   inputs:
%   FS:         samplerate
%   N_FFT:      FFT size
%   NUMBANDS:   number of bands
%   MN:         center frequency of first 1/3 octave band
%   outputs:
```

```
%   A:          octave band matrix
%   CF:         center frequencies

f = linspace(0, fs, N_fft+1);
f = f(1:(N_fft/2+1));
k = 0:(numBands-1);
cf = 2.^(k/3)*mn;
fl = sqrt((2.^(k/3)*mn).*2.^((k-1)/3)*mn);
fr = sqrt((2.^(k/3)*mn).*2.^((k+1)/3)*mn);
A = zeros(numBands, length(f));

for i = 1:(length(cf))
    [a b] = min((f-fl(i)).^2);
    fl(i) = f(b);
    fl_ii = b;

[a b] = min((f-fr(i)).^2);
    fr(i) = f(b);
    fr_ii = b;
    A(i,fl_ii:(fr_ii-1)) = 1;
end

rnk = sum(A, 2);
numBands = find((rnk(2:end)>=rnk(1:(end-1)))
& (rnk(2:end)~=0)~=0, 1, 'last' )+1;
A = A(1:numBands, :);
cf = cf(1:numBands);

%%
function x_stdft = stdft(x, N, K, N_fft)
%   X_STDFT = X_STDFT(X, N, K, N_FFT) returns the short-time
% hanning-windowed dft of X with frame-size N, overlap K and DFT size
%   N_FFT. The columns and rows of X_STDFT denote the frame-index and
%   dft-bin index, respectively.

frames = 1:K:(length(x)-N);
x_stdft = zeros(length(frames), N_fft);

w = hanning(N);
x = x(:);
```

```
for i = 1:length(frames)
    ii = frames(i):(frames(i)+N-1);
x_stdft(i, :) = fft(x(ii).*w, N_fft);
end

%%
function [x_sil y_sil] = removeSilentFrames(x, y, range, N, K)
%   [X_SIL Y_SIL] = REMOVESILENTFRAMES(X, Y, RANGE, N, K) X and Y
%   are segmented with frame-length N and overlap K, where the maximum energy
%   of all frames of X is determined, say X_MAX. X_SIL and Y_SIL are the
%   reconstructed signals, excluding the frames, where the energy of a frame
%   of X is smaller than X_MAX-RANGE

x = x(:);
y = y(:);

frames = 1:K:(length(x)-N);
w = hanning(N);
msk = zeros(size(frames));

for j = 1:length(frames)
    jj = frames(j):(frames(j)+N-1);
    msk(j) = 20*log10(norm(x(jj).*w)./sqrt(N));
end

msk = (msk-max(msk)+range)>0;
count = 1;

x_sil = zeros(size(x));
y_sil = zeros(size(y));

for j = 1:length(frames)
    if msk(j)
        jj_i = frames(j):(frames(j)+N-1);
        jj_o = frames(count):(frames(count)+N-1);
        x_sil(jj_o) = x_sil(jj_o) + x(jj_i).*w;
        y_sil(jj_o) = y_sil(jj_o) + y(jj_i).*w;
        count = count+1;
    end
end
```

```
x_sil = x_sil(1:jj_o(end));
y_sil = y_sil(1:jj_o(end));

%%
function rho = taa_corr(x, y)
%   RHO = TAA_CORR(X, Y) Returns correlation coeffecient between column
%   vectors x and y. Gives same results as 'corr' from statistics toolbox.
xn = x-mean(x);
xn = xn/sqrt(sum(xn.^2));
yn = y-mean(y);
yn = yn/sqrt(sum(yn.^2));
rho = sum(xn.*yn);
```

Noise Suppression using Wiener Filter and Calculation of STOI Values

```
clc

clear

        %Select Clean and Noisy Speech Files to be Used

[CSFname,clFpath]=uigetfile('*.wav',
'Please Select the Clean Speech Audio File');

[NSFname,unFpath]=uigetfile('*.wav',
'Please Select the Noisy Speech Audio File');

prompt = 'Please input the desired initial silence duration of the noisy speech

(in seconds):';

x = input(prompt);

[YN, Fs] = audioread(NSFname);  %Extract Audio Information from
Noisy Speech File

plot(YN);  %Visualize Signal
```

```
soundsc(YN,Fs);   %Listen to Signal

pause(3.2)

IS = x;    %Input Initial noise duration for Wiener Filter

YPR = WienerScalart96(YN, Fs, IS); %Noise Supression of Noisy Speech

plot(YPR)    %Visualize Processed Signal

soundsc(YPR, Fs);   %Listen to Processed Signal


%Use code below if you want to save an audio file of the processed signal


proc = 'processed.wav';  %Assign filename for processed speech signal

audiowrite(proc,YPR,Fs);   %Create audio file of processed speech signal


[YC, Fs] = audioread(CSFname); %Extract Audio Information from

Clean Speech File


diff = length(YC)-length(YPR);    %Calculate difference in samples between
                                  %clean and processed speech signal

YCT = YC(1:end - diff ,:);    %Truncated Clean Speech Signal for STOI

StoiN = stoi(YC, YN , Fs)    %calculates STOI for Unprocessed

Noisy Speech Signal

StoiP = stoi(YCT, YPR , Fs)    %calculates STOI for Processed Speech Signal
```

Experiment's Questionnaire

Consent Form

Outline of Research: The purpose of this research is to measure the intelligibility of audio signals that were processed and summed in certain ratios. There will be 8 sentences in total. After each audio has been heard, firstly, you have to rate the intelligibility of the audio signal from 1 to 5 and secondly try to repeat the sentences of the audio signal. Your answers will be kept confidential and your responses will be kept anonymous.

```
1. I confirm that I have read and understand the above outline of research
and have had the opportunity to ask questions.

2. I understand that my participation is voluntary and that I am free
to withdraw at any time, without giving any reason.

3. I understand that my answers will be kept confidential and that I will
only be identified by a pseudonym in the
publications arising from this research.

4. I agree to take part in the above study and give permission to be audio
recorded and/or filmed during the test session(s).

---------------------------------------
Name of Participant (you do not have to
give your full name here if you prefer to be
known only by first name or pseudonym)

Date ---------------------------


Signature ----------------------


Nationality: ----------------------

Do you have hearing problems?  o Yes o No

English level: -----------------------  (native, advanced,  intermediate)
```

Intelligibility scale

```
1. Speech was completely unintelligible. No words or phrases whatsoever
could be understood.

2. Speech was mostly unintelligible. Occasional words and phrases are
intelligible, but in general, the content of the message could not be understood.

3. Speech was somewhat intelligible. The listener could understand approximately
half the content of the message. The general
content of the message could be understood
although the listener had some difficulty.

4. Speech was mostly intelligible. For the most part, speech was intelligible;
with the exception of few words or phrases.

5. Speech was completely intelligible. Every single word or phrase could be
understood.
```

Transcription of the sentences used in the experiment

Sentences:

- The drip of the rain made a pleasant sound.
- The lazy cow lay on the cool grass.
- Let's all join as we sing the last chorus.
- Stop whistling and watch the boys march.
- He wrote down a long list of items.
- The sky that morning was clear and bright blue.
- Her purse was full of useless trash.
- He knew the skill of the great young actress.
- We find joy in the simplest things.
- The friendly gang left the drug store.