Speech Enhancement in Cochlear Implants

MASTER THESIS

Ioanna Avramidou

Zurich, September 2012

SUPERVISORS:

USZ: Prof. Norbert Dillier

Phonak AG: Dr. Manuela Feilner

TRACK ADVISOR:

ETH ZURICH: Prof. Janos Vörös

ACKNOWLEDGEMENTS

First and foremost, to my supervisors, Prof. Norbert Dillier from the USZ and Dr. Manuela Feilner from Phonak, for their experienced guidance and continuous support throughout this master thesis. To Dr. Peter Derleth, head of the Algorithm Concepts group of Phonak, for the opportunity he offered me by including me among the members of his research team. To all my colleagues at Phonak, who happily shared their knowledge with me. Special acknowledgements to my good teacher Dr. Wai-Kong Lai from the USZ, for his valuable assistance during the clinical tests. Finally, to all the people who participated in the clinical tests, for their willingness to contribute to my work and to science, in general.

The images of the cover picture originated from:

http://www.englishtodaymagazine.com/noticia.php?id=86

http://www.spookypop.com/news/wp-content/uploads/2011/03/FactoryDetail3Lorez.jpg

http://www.iclipart.com/search.php?x=97&y=9&keys=267144&andor=AND&cat=All&tl=clipart&id=111_10_3_17

http://www.blueskymusicstudio.com/Music.html

http://iceyboard.no-ip.org/projects/white_noise_generator/

http://www.flickr.com/photos/sketchaway/6601823611/sizes/l/in/photostream/

Table of Contents

I. INTRODUCTION

The present master thesis, which was completed for the award of the ETH MSc Degree in Biomedical Engineering, was conducted in joint collaboration with the ORL Department of the University Hospital in Zurich and Phonak AG. The initial study plan of the thesis can be found in Appendix A.

The topic addressed in this work is Speech Enhancement (SE) in Hearing Instruments with special emphasis on Cochlear Implants. A Dictionary Learning algorithm for SE, developed by Dr. Christian Sigg and Tomas Dikk, served as a basis framework. The aforementioned algorithm was first optimized, then clinically tested and finally modified using wavelets.

As the performance of the algorithm in relation to Cochlear Implants cannot be directly evaluated by Normal Hearing people, a Cochlear Implant (CI) Simulator provided by Advanced Bionics was employed. The CI Simulator processes the output of the SE algorithm and simulates how it would be perceived by a CI user.

The following chapters include the project's theoretical background, together with an analytic description of its three main parts that were mentioned above.

More specifically, in Chapter 2, the original version of the Dictionary Learning algorithm for Speech Enhancement is presented, by explaining its working scheme as well as its main functions. Furthermore, in the same chapter, the principle of the CI Simulator is analyzed.

Chapter 3 is associated with the optimization of the SE algorithm. First of all, the main parameters to be optimized are described. Moreover, the results of the optimization procedure, from the perspective of both an individual file and many files, are presented. Furthermore, alternative objective measures for the evaluation of the algorithm's performance are discussed. In addition, the algorithm's computational time is investigated with respect to the parameters. Finally, besides the optimization conclusions, additional observations and comments concerning the algorithm are included.

Chapter 4 is dedicated to the evaluation of the algorithm by clinical adaptive SRT tests with the Oldenburg sentences. In the beginning, the selection procedure of the parameter sets used for testing is presented. Next, the results of clinical testing are offered for discussion. These involve both experiments with CI patients as well as with Normal Hearing people using the CI Simulator.

Finally, in Chapter 5, a modification of the original SE algorithm is described. In the proposed implementation, the signals are transferred into the wavelet domain instead of the Fourier domain. Two versions of the wavelet method are analyzed: one that uses separate dictionaries for each scale of the wavelet transform and one that uses a unified dictionary for all scales. The goal of this part of the thesis was to investigate the possible benefits that may arise from an alternative implementation, such as the reduction of the algorithm's computational cost.

II. BASIC THEORETICAL BACKGROUND

A. Dictionary Learning Algorithm

Before describing the algorithm under investigation, we should first define its goal: Speech Enhancement. Let's consider a degraded speech signal x as a linear additive mixture of a clean speech signal s and an interferer signal i,

$$x = s + i \,. \tag{1}$$

The aim of Speech Enhancement is to find an estimation $\hat{s}$ of the clean speech signal such that

$$\left\|\hat{s} - s\right\| << \left\|x - s\right\| . \tag{2}$$

The most prevalent methods for Speech Enhancement that have been proposed in literature are Spectral Subtraction [1] and Vector Quantization [2]. The former calculates an estimation of the clean speech spectrum by subtracting the estimate of the interferer spectrum from the degraded speech signal spectrum. As far as the estimation of the interferer spectrum is concerned, it is calculated during speech inactivity. The major drawback of Spectral Subtraction is that it is unsuitable for non-stationary interferers, for which it results in the generation of musical noise artifacts. On the other hand, in the Vector Quantization method, a codebook of vectors which serves as a model for speech, is trained in the STFT domain. In the enhancement phase, the mixture is projected on the closest clean speech prototype of the trained codebook. This method, however, involves a large error due to quantization. Finally, a recently proposed algorithm for audio noise reduction [3] applies sparse coding shrinkage on the principal components of the noisy signal. This method compensates for auditory deficits of NH people, such as reduced frequency selectivity.

The algorithm under investigation [4] trains dictionaries as models of the speech and the interferer signal classes. During enhancement, the degraded speech signal is sparsely coded on the concatenation of a speech and an interferer dictionary. In that way, the mixture can be separated into its underlying speech and interferer components. By discarding the interferer component, Speech Enhancement is accomplished. Figure 1 presents a schematic overview of both the learning and the enhancement steps of the Dictionary Learning method. $D^{(s)}$ and $D^{(i)}$ represent the speech and the interferer dictionaries, respectively, while D their concatenation. The coding matrix that results from the sparse coding of the mixed signal on D, is denoted by c. The feature space where the algorithm operates is the STFT domain.

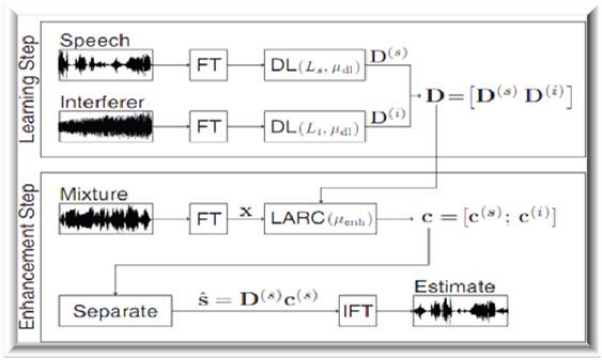


Figure 1: A schematic overview of the DL method [4].

Sparse coding is fundamental both for the training and the enhancement phase of the DL algorithm. Its goal is to approximate a given signal $x \in R^D$ as a linear combination of only a few elements (atoms) of a dictionary $D \in R^{D \times L}$. The coefficients of the linear combination constitute the coding vector $c \in R^L$. Sparsity lies in the fact that there is a restriction in the maximum number of dictionary atoms that can be used for the representation of the signal. In other words, the number of non-zero elements of the coding vector, referred to as cardinality (K), is limited. The sparse coding problem can be formulated as

$$\begin{aligned} c^* &= \arg\min_c \|x - Dc\|_2 \\ s.t. \quad & \|c\|_0 \le K \end{aligned} \quad . \tag{3}$$

The algorithm in [4] that provides a solution to the sparse coding problem is the Batch Least Angle Regression with Coherence Criterion (LARC). Its steps are illustrated in Figure 2. The LARC algorithm is a modification of the Least Angle Regression algorithm (LARS) [5]. Similarly to LARS, each iteration of LARC consists of an atom selection and a coding coefficient update step. The atom that is selected in every iteration is the most coherent to the current residual ($r = x - Dc$) one. The coding coefficient update proceeds in the equiangular direction of the selected atoms. What differentiates LARC from LARS, is that it is not terminated when a new atom has equal correlation with the residual as all atoms in the active set, but when the maximum residual coherence reaches a specified down threshold. The value of the coherence threshold determines the sparsity of the coding.

Input: $\mathbf{x} \in \mathbb{R}^D$; $\mathbf{D} \in \mathbb{R}^{D \times L}$; $\mathbf{G} = \mathbf{D}^\top \mathbf{D}$; μ_{dl}
Output: $\mathbf{c} \in \mathbb{R}^L$

$\mathbf{c} \leftarrow \mathbf{0}$; $\mathbf{y} \leftarrow \mathbf{0}$; $\mathcal{A} \leftarrow \{\}$
$\boldsymbol{\mu}^{(x)} \leftarrow \mathbf{D}^\top \mathbf{x}$; $\boldsymbol{\mu}^{(y)} \leftarrow \mathbf{0}$
while $|\mathcal{A}| < D$ **do**
  $\boldsymbol{\mu} \leftarrow \boldsymbol{\mu}^{(x)} - \boldsymbol{\mu}^{(y)}$
  $j^* \leftarrow \arg\max_j |\mu_j|,\ j \in \mathcal{A}^c$
  $\mathcal{A} \leftarrow \mathcal{A} \cup \{j^*\}$
  if $\mu_{j^*} / ||\mathbf{x} - \mathbf{y}||_2 < \mu_{\text{dl}}$ **then**
    break
  end if
  $\mathbf{s} \leftarrow \text{sign}(\mu_{\mathcal{A}})$
  $\mathbf{g} \leftarrow \mathbf{G}^{-1}_{(\mathcal{A},\mathcal{A})} \mathbf{s}$
  $b \leftarrow (\mathbf{g}^\top \mathbf{s})^{-\frac{1}{2}}$
  $\mathbf{w} \leftarrow b\mathbf{g}$
  $\mathbf{u} \leftarrow \mathbf{D}_{(:,\mathcal{A})} \mathbf{w}$
  $\mathbf{a} \leftarrow \mathbf{G}_{(:,\mathcal{A})} \mathbf{w}$
  $\gamma \leftarrow \min^+_{k \in \mathcal{A}^c} \left(\frac{\mu_{j^*} - \mu_k}{b - a_k}, \frac{\mu_{j^*} + \mu_k}{b + a_k} \right)$
  $\mathbf{y} \leftarrow \mathbf{y} + \gamma \mathbf{u}$
  $\mathbf{c}_{\mathcal{A}} \leftarrow \mathbf{c}_{\mathcal{A}} + \gamma \mathbf{w}$
  $\boldsymbol{\mu}^{(y)} \leftarrow \boldsymbol{\mu}^{(y)} + \gamma \mathbf{a}$
end while

Figure 2: The LARC algorithm [4].

Regarding the training of the dictionaries, the K-SVD algorithm [6] is used. The aforementioned algorithm, proposes an efficient way for the factorization of the training data matrix $X \in R^{D \times N}$ into a dictionary $D \in R^{D \times L}$ and a coding matrix $C \in R^{L \times N}$. The factorization aims at minimizing

$$D, C = \arg\min_{D,C} \|X - DC\|_2$$
$$s.t. \qquad . \qquad (4)$$
$$\|C\|_0 \le K(cardinality) \qquad and \qquad \|d_{(:,l)}\|_2 = 1, \forall l = 1,\ldots,L$$

The K-SVD algorithm is presented in Figure 3. It consists of three steps: the initialization, the coding update and the dictionary update step. In the initialization step, the initial dictionary is formed by sampling the training matrix X. In the coding update step, each column of the coding matrix C is separately updated by applying the LARC algorithm for each training sample, based on the current dictionary and using a specified coherence threshold. Here it becomes evident how sparse coding is involved in the training phase besides the enhancement phase. Finally, in the dictionary update step, each atom of the dictionary is separately updated by performing Singular Value Decomposition on the part of the residual norm where the current atom was involved. The algorithm alternates between the last two steps for several iterations.

Input: $\mathbf{X} = \mathbb{R}^{D\times N}$; $\mathbf{D} = \mathbb{R}^{D\times L}$; $\mathbf{C} = \mathbb{R}^{L\times N}$
Output: Updated dictionary $\mathbf{D}$
for $l \leftarrow 1$ to L **do**
  $\mathbf{d}_{(:,l)} \leftarrow \mathbf{0}$
  $\mathcal{N} \leftarrow \{n | C_{l,n} \neq 0, 1 \le n \le N\}$
  $\mathbf{R} \leftarrow \mathbf{X}_{(:,\mathcal{N})} - \mathbf{D}\mathbf{C}_{(:,\mathcal{N})}$
  $\mathbf{g} \leftarrow \mathbf{c}^{\top}_{(l,\mathcal{N})}$
  $\mathbf{h} \leftarrow \mathbf{R}\mathbf{g}$
  $\mathbf{h} \leftarrow \mathbf{h}/\|\mathbf{h}\|_2$
  $\mathbf{g} \leftarrow \mathbf{R}^{\top}\mathbf{h}$
  $\mathbf{d}_{(:,l)} \leftarrow \mathbf{h}$
  $\mathbf{c}_{(l,\mathcal{N})} \leftarrow \mathbf{g}^{\top}$
end for

Figure 3: The approximate K-SVD algorithm [4].

At this point, a couple of details should be mentioned with regard to the processing inside the Dictionary Learning algorithm. The first one is associated with the nature of the vectors that are sparsely coded either in the training or in the enhancement phase. As mentioned before, the feature

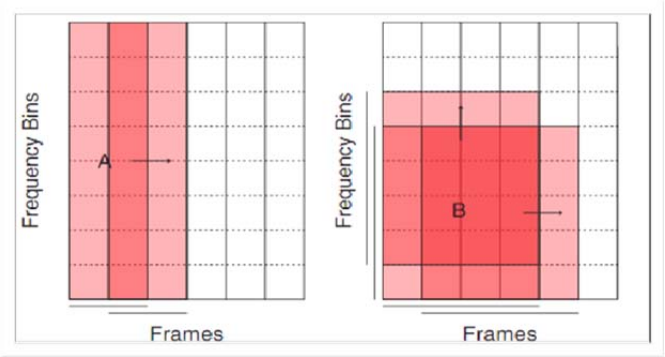


Figure 4: Extraction of overlapping blocks from the STFT domain. (A): Tall and narrow patch (B): Short and wide patch [4].

space is the STFT. Nevertheless, the STFT coefficients do not directly formulate the feature vectors. Instead, the STFT domain is tiled resulting into the extraction of overlapping blocks which later on are vectorized giving rise to the feature vectors e.g. the columns of the training matrix X. The tiling process is illustrated in Figure 4. Various geometries for the overlapping blocks (patches) can be chosen.

The second detail is related to an additional processing step that is introduced right before the inverse STFT of the estimated clean speech spectrum, the final step of the algorithm. There, filtering that aims at the reduction of the musical noise artifact takes place, using the instantaneous geometric approach (GA) estimator [7].

In conclusion, having incorporated the two details mentioned above, the training and enhancement processing steps of the DL algorithm are presented in Figures 5 and 6, respectively.

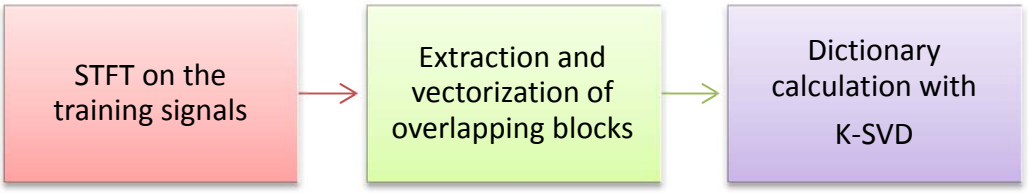


Figure 5: Training of either the speech or the interferer dictionary.

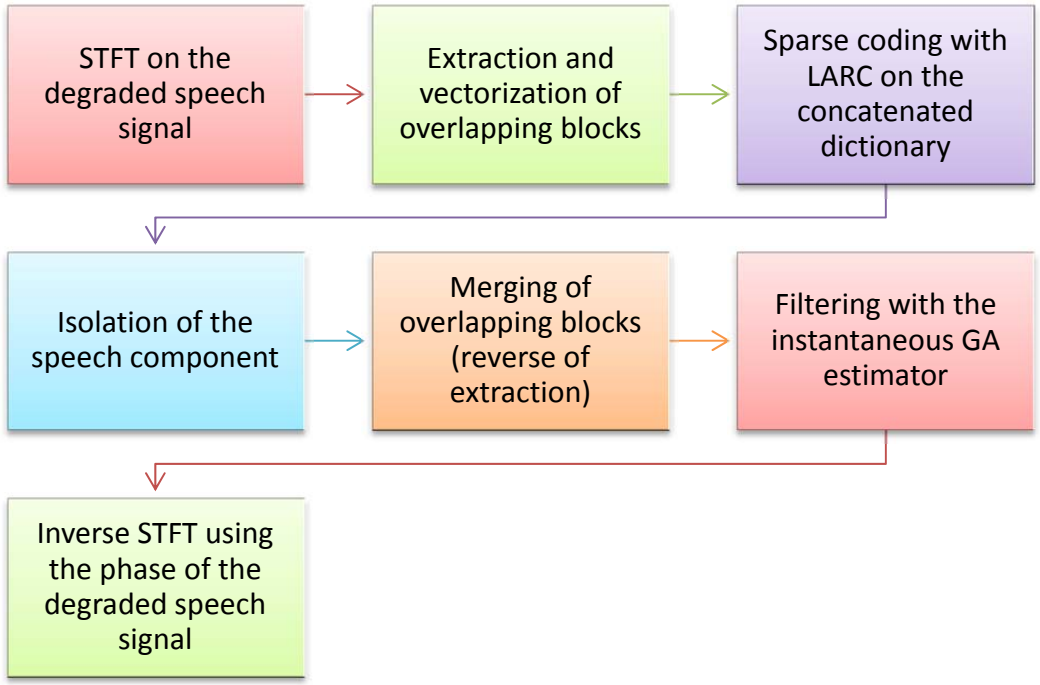


Figure 6: Enhancement of the degraded speech signal.

B. Cochlear Implant Simulator

The Cochlear Implant Simulator provided by Advanced Bionics [8], models both the CI processor and the spread of excitation that takes place after electrical stimulation of the cochlea. Its aim is to cause to Normal Hearing people similar impairments in speech understanding to those that are observed in CI patients. A high correlation between the confusion matrices that resulted from vowel and consonant recognition tests to NH people with the Simulator and the confusion matrices that resulted from the same tests to CI patients, has proven the effectiveness of the Simulator.

The working principle of the CI Simulator is that it decomposes its input into 15 logarithmically spaced frequency channels and then reconstructs it after multiplying the envelope of each channel with noise. Various degrees of impairment can be simulated by adjusting the drop-off of the noise spectrum away from its peak (40-5 dB/octave). A small noise spectrum slope (e.g. 5 dB) leads to a wide spread of excitation and therefore to a larger deterioration in speech understanding.

The core of the CI Simulator is a Vocoder. The processing steps inside the Vocoder are illustrated in Figure 7.

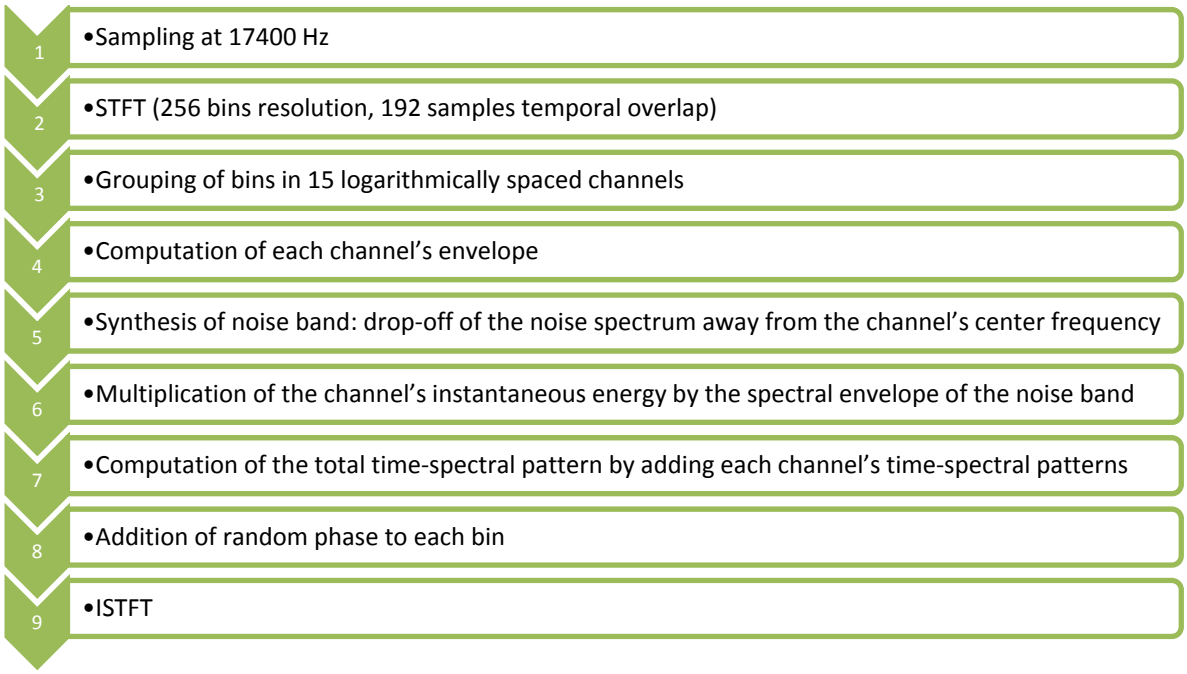


Figure 7: Processing steps in the Vocoder of the CI Simulator.

Figures 8 and 9 depict the spectrogram of a clean speech signal before and after the CI Simulator, respectively. In Figure 9, the left spectrogram corresponds to noise spectrum slope of 40 dB/octave, while the right one to 5 dB/octave. For an audible impression of the functionality of the CI Simulator Audio_1, Audio_2 and Audio_3 are additionally provided. Audio_1 is the clean speech signal before the CI Simulator. Audio_2 is the clean speech signal after the CI Simulator, where noise with spectrum slope of 40 dB/octave was used. Finally, Audio_3 is the clean speech signal after the CI Simulator, where noise with spectrum slope of 5 dB/octave was used. From Figure 9 as well as from

the Audio files, it becomes evident that noise is spread all over the frequency channels. After the CI Simulator, the sound is no longer clean and gives the impression of whispering. The left and right spectrograms in Figure 9 don't appear a lot different from each other. However, in Audio_3, where a smaller slope was used for the noise spectrum in comparison to Audio_2, the speech is much more distorted. In all the following simulations, a slope of 40 dB/octave has been used.

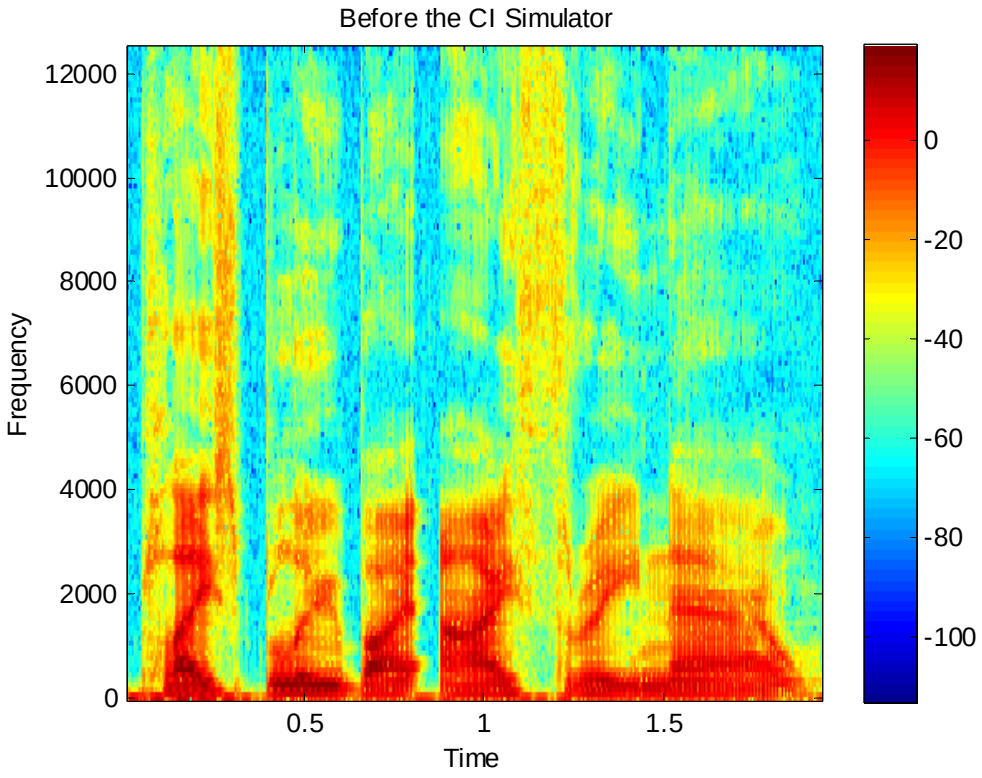


Figure 8: Spectrogram of a clean speech signal before the CI Simulator.

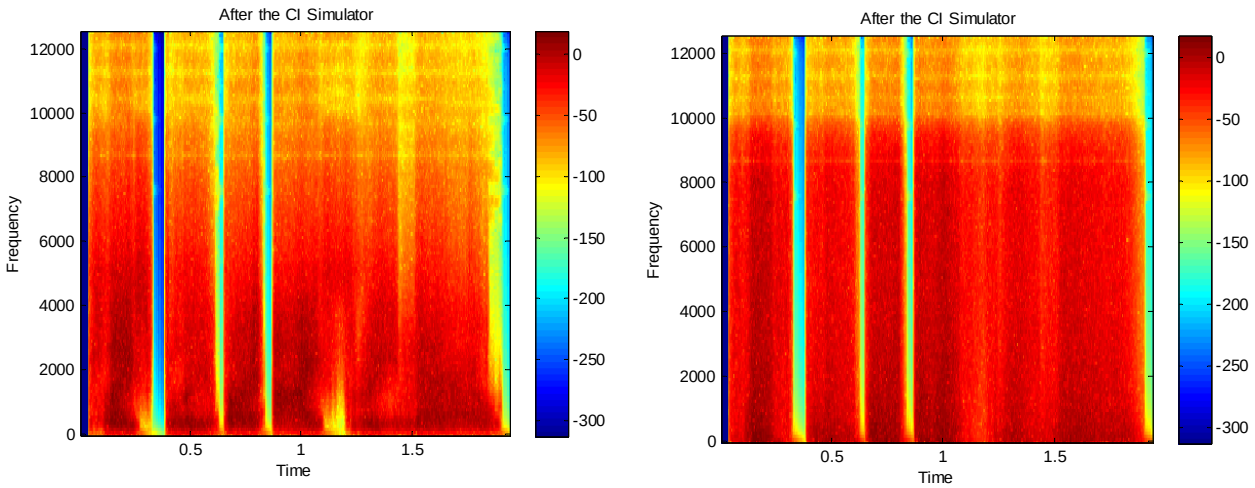


Figure 9: Spectrogram of a clean speech signal after the CI Simulator. (Left): noise spectrum slope 40 dB/octave. (Right): noise spectrum slope 5 dB/octave.

Attention needs to be paid to the fact that the input of the CI Simulator has to be properly scaled. To begin with, the CI Simulator is differently calibrated from the Media Data Base of Phonak. In the Media Data Base (system 1) the following relation holds

$$\begin{aligned} &80dBSPL == -43dBFSrms \\ &i.e. SPL_1 = 20\log x_1 + 123 \end{aligned} . \tag{5}$$

In the CI Simulator (system 2) a different relation holds

$$\begin{aligned} &60dBSPL == -12dBFSpeak = -15dBFSrms \\ &i.e. SPL_2 = 20\log x_2 + 75 \end{aligned} . \tag{6}$$

From (5) and (6) it can be inferred that

$$calibration \quad scale = \frac{x_2}{x_1} = 10^{\frac{48}{20}} . \tag{7}$$

Furthermore, the fractional value within the range of [0 1] of a .wav file, should be converted to actual level. This depends on the number of bits with which the .wav file is written. For example, if the number of bits equals 16, then a fractional value of 1 corresponds to actual level 2^{15}. Therefore, the input should also be multiplied by the

$$bitwidth \quad scale = 2^{nbits} . \tag{8}$$

Finally, since AGC is not incorporated in the CI Simulator, the input is not allowed to exceed 60 dB SPL. For this reason, the input should be normalized such that its maximum value is reduced to 60 dB SPL. This value according to (5) corresponds to

$$loudness \quad scale = x_1 = 10\frac{-63}{20} . \tag{9}$$

In conclusion, if a signal s is read from a written with 16-bits .wav file, that originates from the Media Data Base of Phonak, in order to be processed by the CI Simulator it should first be scaled according to the following relation

$$\begin{aligned} s_{scaled} &= s \times calibration \quad scale \times bitwidth \quad scale \times loudness \quad scale / \max(abs(s)) \\ &= s \times 5.8271 \times 10^3 / \max(abs(s)) \end{aligned} . \tag{10}$$

The value 5.8271×10^3 is very close to 5×10^3 that was determined by trial as the necessary total scaling coefficient.

III. PARAMETER OPTIMIZATION AND INVESTIGATION

A. List of Parameters

The performance of the SE algorithm was optimized with respect to the following 4 parameters:

Residual Coherence Threshold (μ): This parameter serves as the stopping criterion of the sparse coding algorithm (LARC), which was presented in paragraph II.A. It precisely appears in line 9 of LARC (Figure 2). When a signal is sparsely coded on a dictionary, its components that are more coherent to the dictionary will be coded before the less coherent ones. Furthermore, in every iteration of LARC, as one more atom of the dictionary is added to the active set of atoms, the maximum residual coherence decreases. Therefore, by determining a residual coherence threshold as the stopping criterion of sparse coding, the coherent components of the signal can be separated from the incoherent ones. Moreover, the value of μ defines the degree of sparsity in the coding. A larger value leads to a more sparse coding. Additionally, when the coding is very sparse, the speech component of the mixture might be explained by too few atoms of the speech dictionary. This results to source distortion, i.e. inadequate representation of the clean speech signal by the active set of speech dictionary atoms. On the other hand, when the coding is very dense, some parts of the speech component might be explained by atoms of the interferer dictionary and vice versa. This phenomenon is called source confusion. The residual coherence threshold thus controls the trade-off between the unwanted phenomena of source distortion and source confusion by regulating the sparsity of coding. Besides this, μ is highly associated with the computational time required for sparse coding. The larger the value of μ, the faster the execution of LARC. Finally, sparse coding is involved both in dictionary learning and in enhancement phases. Nevertheless, the selection of the value of μ has a much larger impact on the enhancement phase, for which it was optimized. For the training of the dictionaries μ was set to 0.2.

Beta (b): This parameter is a smoothing constant of the instantaneous geometric approach (GA) estimator that was mentioned in paragraph II.A. In the GA estimator, the final estimated clean speech spectrum is provided by multiplying the degraded speech signal spectrum with a gain function equal to

$$H = \left(\sqrt{\frac{1 - \frac{(\gamma + 1 - \xi)^2}{4\gamma}}{1 - \frac{(\gamma - 1 - \xi)^2}{4\xi}}} \right)^b . \tag{11}$$

γ is the instantaneous a posteriori SNR defined as

$$\gamma = \frac{\mathrm{X}^2}{\hat{\mathrm{I}}^2}, \tag{12}$$

where X is a sample of the spectrum of the degraded speech signal and $\hat{\mathrm{I}}$ is a sample of the spectrum of the estimated by the algorithm interferer signal.

ξ is the instantaneous a priori SNR defined as

$$\xi = \frac{\hat{S}^2}{\hat{I}^2}, \tag{13}$$

where $\hat{S}$ is a sample of the spectrum of the enhanced by the algorithm speech signal and $\hat{I}$ is a sample of the spectrum of the estimated by the algorithm interferer signal.

Therefore,

$$\hat{S}_{GA} = HX \ . \tag{14}$$

From (14) it is obvious that when beta equals zero, the enhanced signal is the same as the degraded speech signal. For beta equal to 1,

$$\hat{S}_{GA} \approx \hat{S} \ . \tag{15}$$

The choice of the parameter beta does not influence the computational time.

Geometric Index: The parameter geometric index defines the morphology of the overlapping blocks which are extracted from the STFT space both during dictionary training and enhancement (Figure 4). The range of available indices is 1-16, each one corresponding to a different patch geometry. The available patch geometries are listed in Table 1. A tall and narrow patch captures better the harmonic content of a signal, while a short and wide patch favors the temporal dynamics of the signal.

INDEX	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
HEIGHT	1024	512	256	128	256	128	64	128	64	32	64	32	16	32	16	8
WIDTH	2	4	4	4	8	8	8	16	16	16	32	32	32	64	64	64

Table 1: Geometric indices and corresponding patch morphologies.

The morphology of the patches not only determines the performance of the algorithm, but also highly affects the computational time of LARC. This can be justified by the fact that various tiling rules of the STFT space defined by different geometric indices, lead to various dimensionalities of the matrix that will be sparsely coded, as well as of the dictionaries.

FFT Size: A STFT is applied both on the training signals during the dictionary learning phase and on the degraded speech signal given for enhancement. In the context of an STFT transform, the signal is divided in time frames of certain duration, on which a DFT is applied with the FFT algorithm. The number of points of the DFT is equal to the length of the signal contained in one time frame. Therefore, by the parameter FFT Size, we refer to the length of the signal segment that corresponds to one time frame. The aforementioned are connected to each other with the following relation

$$FFT \quad size = time \quad frame \quad duration \quad \times \quad sampling \quad frequency \ . \tag{16}$$

From the above it becomes obvious that the FFT size would be critical in a real time implementation of the algorithm, as the time frame duration determines the delay of the system.

B. Objective Measure

The performance of the algorithm for various values of the parameters subject to optimization could be measured by subjective listening. However, this requires long time as well as trained listeners. For this reason, in order to determine the optimal parameters, the output of the algorithm was objectively evaluated by the "frequency weighted segmental SNR (fwSegSNR)". This objective measure has shown to correlate well with both subjective speech quality and subjective speech intelligibility scores [9]. The fwSegSNR is defined as [10]

$$fwSegSNR(S,\hat{S}) = \frac{10}{M}\sum_{m=0}^{M-1}\frac{\sum_{j=1}^{K} W(j,m)\log_{10}\dfrac{|S(j,m)|^2}{(|S(j,m)|-|\hat{S}(j,m)|)^2}}{\sum_{j=1}^{K} W(j,m)}, \tag{17}$$

where M is the total number of frames, K is the number of frequency bands, $|S(j,m)|$ is the weighted (by a Gaussian-shaped window) clean signal spectrum in the j[th] frequency band at the m[th] frame, and $|\hat{S}(j,m)|$ is the weighted enhanced signal spectrum in the same frequency band and frame. More specifically, for the calculation of the weighted spectra $|S(j,m)|$, the signal bandwidth is first divided in 25 bands spaced in proportion to the ear's critical bands. The fast spectra are then multiplied by overlapping Gaussian-shaped windows and the weighted spectra are summed up within each band. Finally, $W(j,m)$ is the weighting function placed on the j[th] band and is computed according to:

$$W(j,m) = |S(j,m)|^{\gamma}, \tag{18}$$

with γ equal to 0.2 in order to obtain maximum correlation.

In fact, the objective measure of the algorithm's enhancement performance was not the absolute fwSegSNR, but the fwSegSNR gain in relation to when the degraded speech signal, instead of the enhanced speech signal, is compared to the clean speech signal. This can be formulated as

$$fwSegSNR\ \ gain = fwSegSNR(S,\hat{S}) - fwSegSNR(S,X), \tag{19}$$

where X is the weighted spectrum of the degraded speech signal.

C. One-file Based Optimization

The effect of changing the parameters described in paragraph III.A during enhancement, was investigated for a single speech file. The speech file was mixed with 7 different noise types (babble, factory, piano, street, volvo car, white noise, wind) at SNRs ranging between -6 and 6 dB with step 3 dB. The 4 parameters were treated as independent. When varying one parameter the remaining 3 were set to default values, which were approximately decided as optimal by roughly observing the algorithm's behavior before starting the standard optimization procedure. The default values for the 4 parameters are listed in Table 2. The ranges of variation of the 4 parameters during optimization are listed in Table 3.

Res_Coh_Thr (μ)	Beta (b)	Geom_Idx	FFT Size
0.1	0.9	5	512

Table 2: Default parameter values.

Res_Coh_Thr (μ)	Beta (b)	Geom_Idx	FFT Size
0.1 – 0.9 with step 0.2	0 - 1 with step 0.2	1-16	256,512,1024

Table 3: Ranges of variation of the parameters' values.

The objective measure of the algorithm's performance, fwSegSNR gain, with respect to the 4 parameters, is presented in the following figures for 3 (babble, piano, white) of the 7 types of noise, both with and without the CI Simulator.

Residual Coherence Threshold:

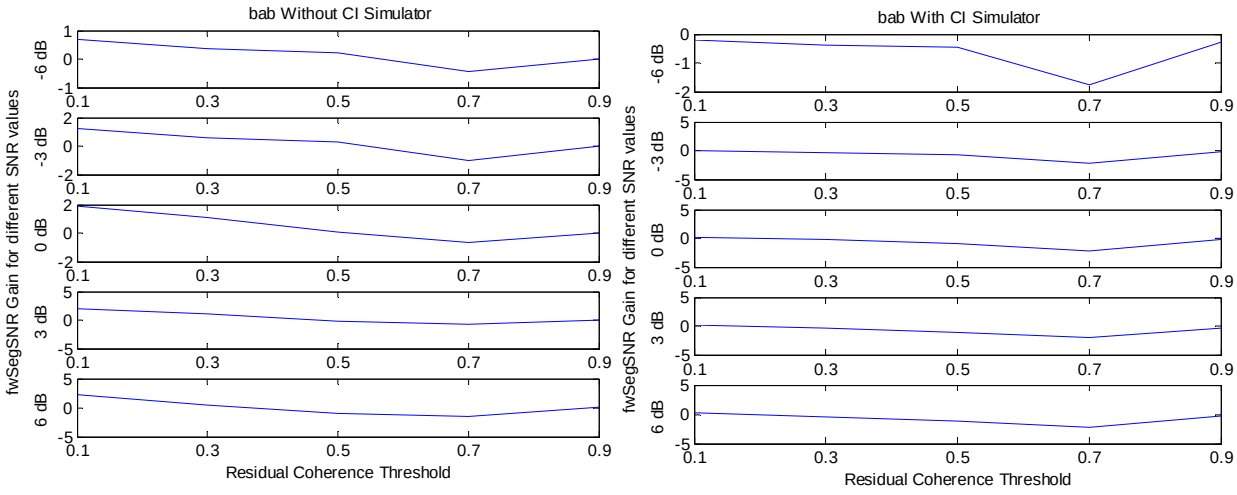


Figure 10: Optimization of the Residual Coherence Threshold for babble noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

From Figure 10, it becomes obvious that 0.1 is the optimal value for μ. A small value of μ corresponds to dense coding on the dictionaries. A too dense coding, however, can lead to source confusion where parts of the interferer component are explained by elements of the speech dictionary and are included in the enhanced signal. For this reason, a judgment for the optimal degree of sparsity in coding and thus for the value of μ, can only be made by subjective listening, as the objective measure is not a proper indication of source confusion. In this case, subjective listening verified that 0.1 is the optimal value for μ.

As μ increases, the enhanced signal is more noisy. In the extreme case of 0.9, the enhanced signal sounds almost like the degraded signal. For values within the range [0.3-0.7], the enhanced file is not only noisy but also corrupted. This phenomenon exhibits a peak for 0.7 which also reflects on the objective measure. At this point, it is worth to mention that a smaller μ results in a larger computational time, since the LARC is forced to run for more iterations until the down threshold is reached. However, without doubt 0.1 is the optimal value. The maximum gain achieved is approximately 2 dB.

Furthermore, the effect of changing μ on the objective measure follows the same trend regardless of the SNR of the degraded speech signal and independent of whether we measure the performance with or without the CI Simulator.

To obtain an impression of the influence of μ on the quality of the enhanced sound Audio_4-6 are included. Audio_4 is a degraded speech file with babble noise at 0 dB SNR. Audio_5 is the same file enhanced with μ=0.1 and Audio_6 with μ=0.7.

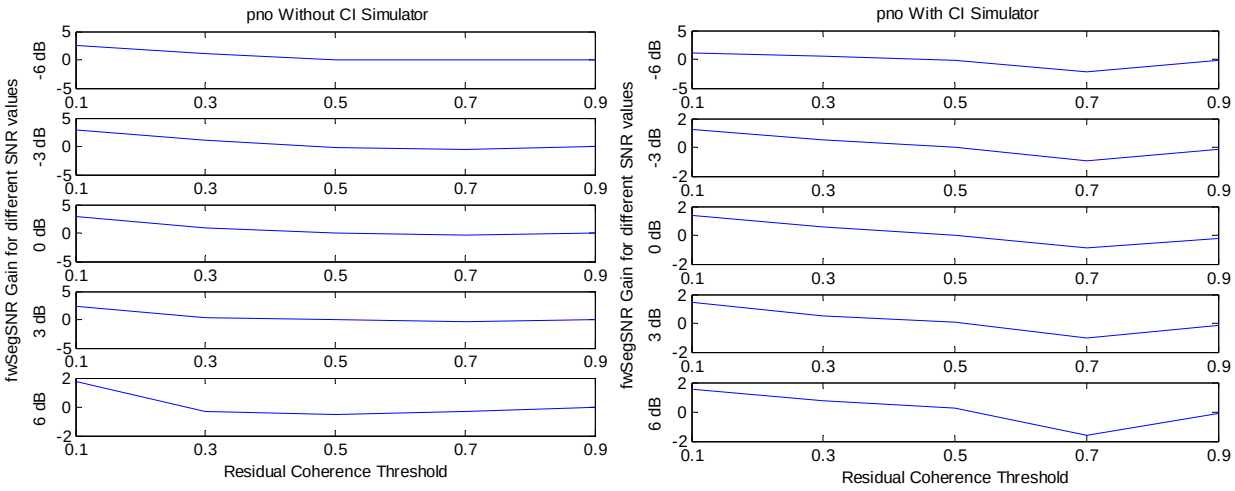


Figure 11: Optimization of the Residual Coherence Threshold for piano noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

Figures 11 and 12, illustrate the results for piano and white noise respectively. It is apparent that 0.1 is the optimal value for μ regardless of the use of the CI Simulator and of the SNR of the degraded file. The maximum gain achieved for piano is approximately 3 dB, while for white noise approximately 6 dB. As piano is a very well structure signal class, it is possible to create a successful

dictionary for it. Moreover, the characteristics of a piano signal differ a lot from the characteristics of speech and thus the enhancement algorithm exhibits a high performance for speech in piano noise. White noise, on the other hand, is incoherent to the speech dictionary and, therefore, rejected by it.

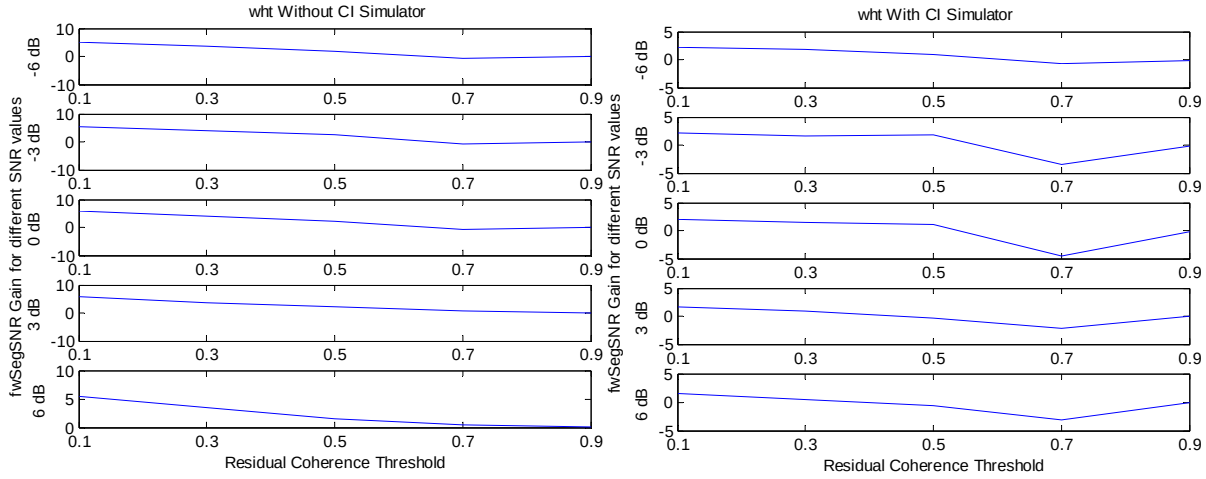


Figure 12: Optimization of the Residual Coherence Threshold for white noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

Beta:

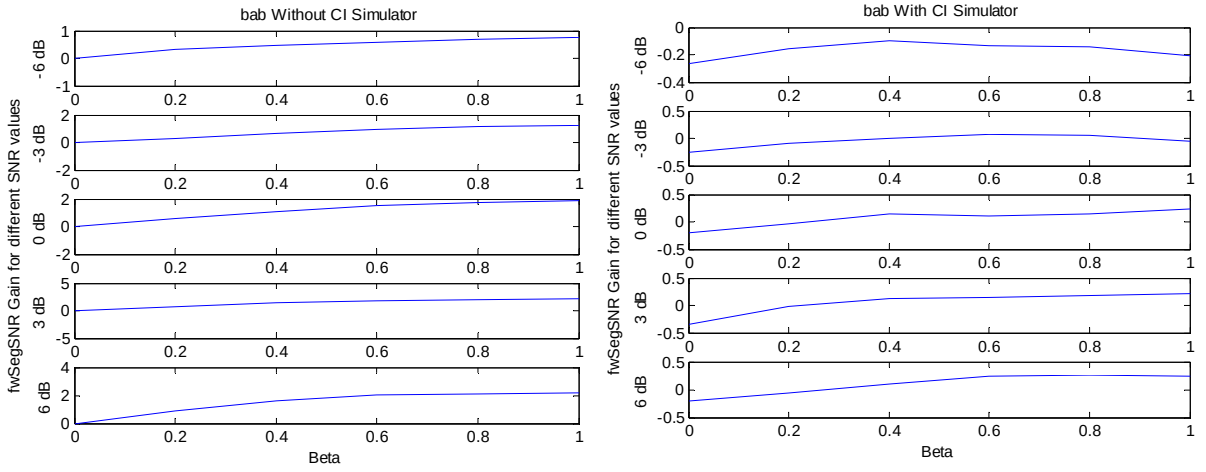


Figure 13: Optimization of Beta for babble noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

In Figure 13, it can be seen that the objective measure increases as beta increases, indicating better performance. By subjective evaluation, it can be observed that a larger value of beta leads to better enhancement. However, for beta=1 the sound is more artificial, while for a slightly smaller value the sound becomes more natural. Therefore, the choice of beta should optimize the trade-off between sound quality and enhancement. A choice between 0.8-1 is optimal as it provides good results in both senses, while the differences in quality or in enhancement are minor within this range. When the CI Simulator is included in the evaluation, a large beta is again preferred. Objectively, the optimal

value is slightly SNR dependent, while subjectively, beta=1 offers the largest degree of speech enhancement. The sound is anyway artificial after the CI Simulator and the degree of speech enhancement is more important in this case than speech quality. When beta=0, there is no enhancement.

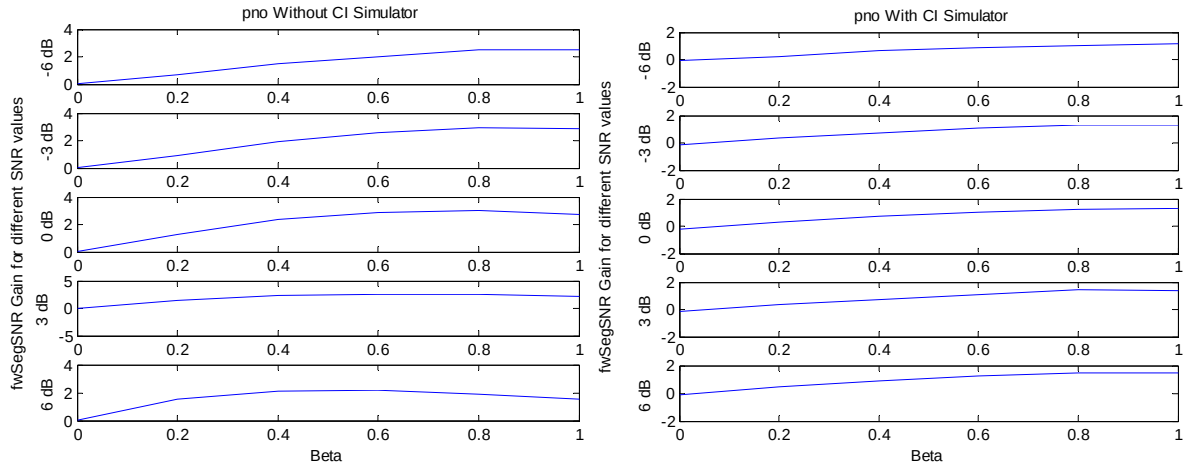


Figure 14: Optimization of Beta for piano noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

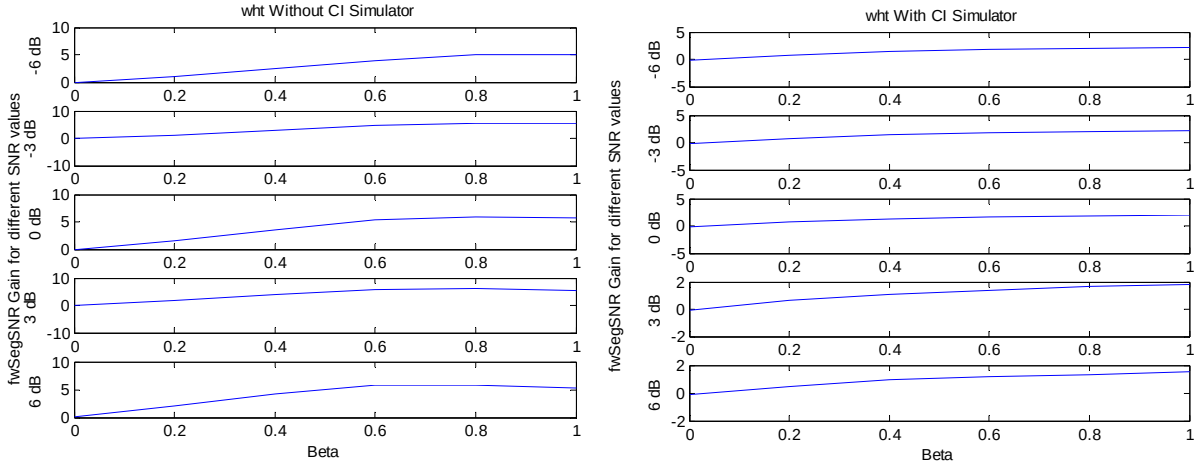


Figure 15: Optimization of Beta for white noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

As it can be seen in Figures 14 and 15, when the performance is measured after the CI Simulator, the optimal value for beta is 1 both for piano and white noise. However, when the Simulator is not taken into account, the optimal beta depends on the SNR of the degraded speech file. Two examples are included: Audio_7 is speech degraded with piano noise at 0 dB SNR and enhanced with beta=0.6, while in Audio_8 beta=1.

Geometric Index:

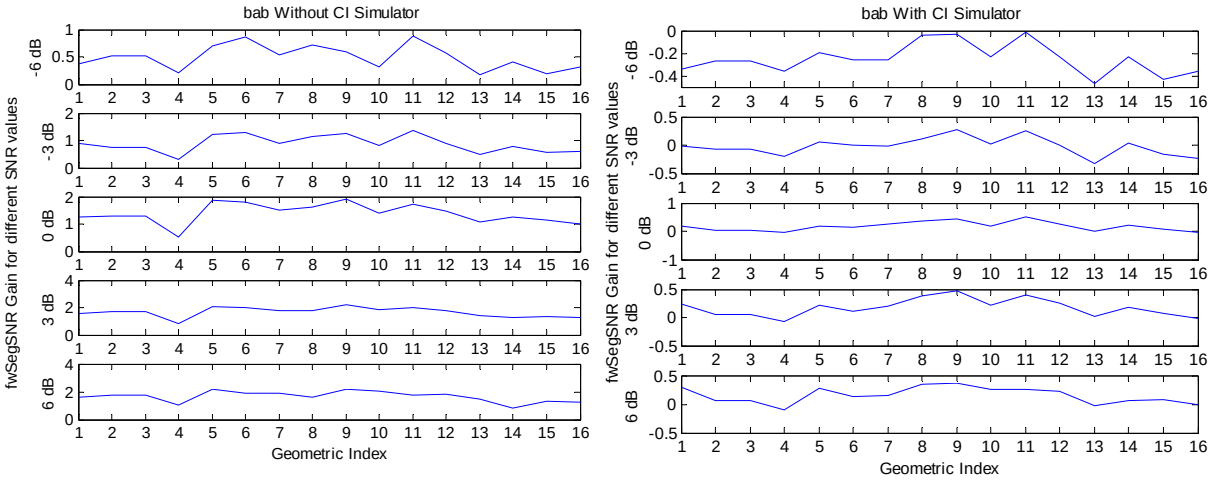


Figure 16: Optimization of Geometric Index for babble noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

The Geometric Index is the most complicated parameter given for optimization. Besides affecting the degree of enhancement, it highly influences the quality of the enhanced sound. For this reason, the selection of the value of this parameter cannot be based on the objective measure, but requires subjective listening as well. For example, in the case of 0 dB SNR without the CI Simulator, patch 5 appears to perform very well (Figure 16). However, the corresponding audio file sounds distorted. On the other hand, patch 15 in the same example corresponds to a low value for the objective measure. Nevertheless, by listening it can be observed that it results into good enhancement although the speech appears artificial.

Regarding the selection of the Geometric Index without the CI Simulator, it can be observed by subjective evaluation that for narrow patches (1-3), the enhanced file is distorted and there is low degree of enhancement. This is more evident for patch 4, which is also short except for narrow. For patch 4 the enhanced file is very noisy. This can also be reflected in the objective measure.

The width of the patch is critical, because it controls the amount of temporal information that is captured. 1 point in the STFT space corresponds to 32 msec when the default FFT of 512 points is applied and the sampling frequency equals 16 kHz. Therefore, a width of maximum 4 points (patches 1-4) captures the information of only 128 msec. It can be observed from the input file, that a word corresponds to approximately 280 msec. From this it can be derived that in order to have good enhancement, the patch should not capture less temporal information than what is contained in one word. When the patch is very wide (64 points for patches 15-16), there is very good noise suppression. However, the aforementioned patches are short and thus don't capture the spectral information very well, resulting into artificial speech quality. A good compromise of all the above is patch 9, which is adequately wide (16 points) and tall (64 points). For this patch, the noise suppression is a little bit lower than for patch 15, but the speech sounds very natural.

Moreover, the optimal Geometric Index is different without and with the CI Simulator. This can be mainly detected by subjective listening. While the degree of noise suppression is similar in both cases

for the same patch, artifacts that appear without the Simulator for certain patches, have a minor impact when the CI Simulator is used. For example, patch 11 leads to the appearance of an unpleasant artifact, which is diffused after the CI Simulator. Therefore, since patch 11 also provides good speech enhancement, it becomes the optimal choice for CIs.

Finally, the Geometric Index is associated with the computational time of sparse coding, but this will not be discussed at the moment.

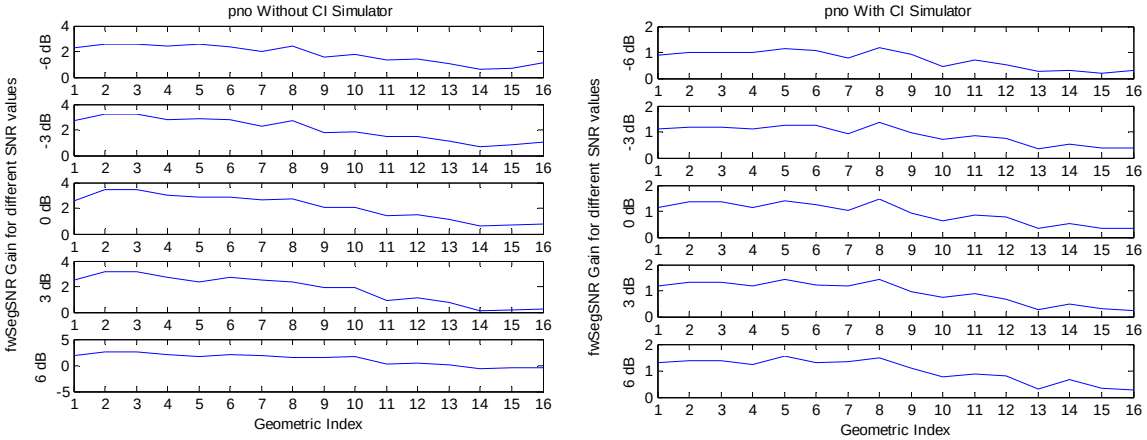


Figure 17: Optimization of Geometric Index for piano noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

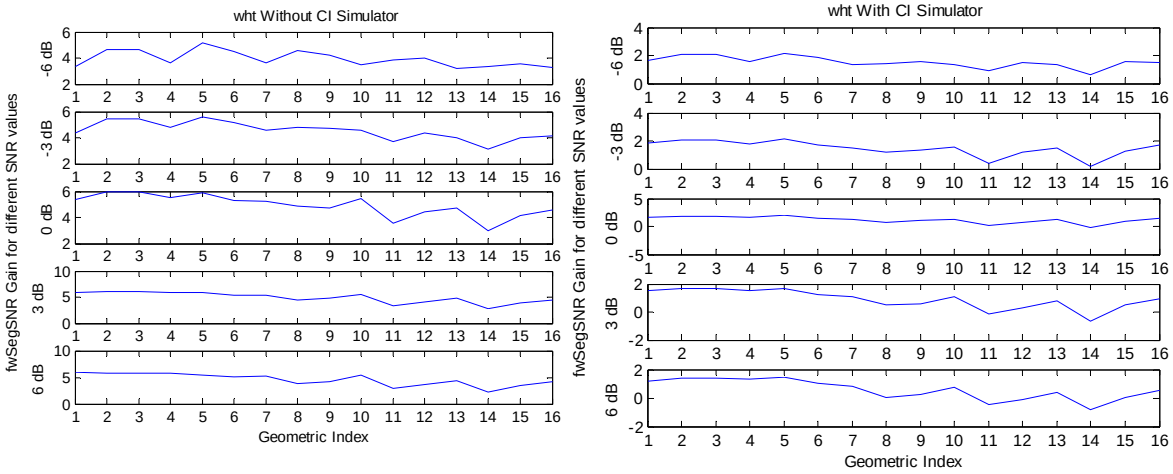


Figure 18: Optimization of Geometric Index for white noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

The selection of the patch depends highly on the type of noise that needs to be suppressed. For piano noise a tall patch is needed to capture the harmonic content. This can also be seen in the objective measure (Figure 17). In addition, the patch should fit the characteristics of speech so that the speech component is well represented and not distorted. Patch 6 (height:128 width:8) offers a

good compromise. When the CI Simulator is included, by subjective listening is can be observed that a tall and narrow patch is the ideal choice for suppression of piano noise. For example patch 2 (height:512 width:4) results in good speech enhancement for CIs.

For white noise a tall and narrow patch (e.g. 1) offers very good representation of speech in the enhanced file. However, it involves a high amount of musical noise. Patch 5 on the other hand (height: 256 width:8) offers good speech enhancement. But for this patch an unpleasant artifact appears, making a shorter and wider patch, such as 8 (height: 128 width:16), more appropriate for speech enhancement without the CI Simulator, since it produces a more smooth result at the cost inevitably of speech clarity. Nevertheless, when the CI Simulator is included, the artifact produced by patch 5 disappears and the sharpness of speech provided by this patch remains, making it optimal for white noise in CIs. The reduction of the importance of this artifact when the CI Simulator is introduced can be audible in Audio_9 and Audio_10. Audio_9 is speech degraded with white noise at 0 dB SNR, which has been enhanced with patch 5 before the CI Simulator. Audio_10 is the same, but after the CI Simulator.

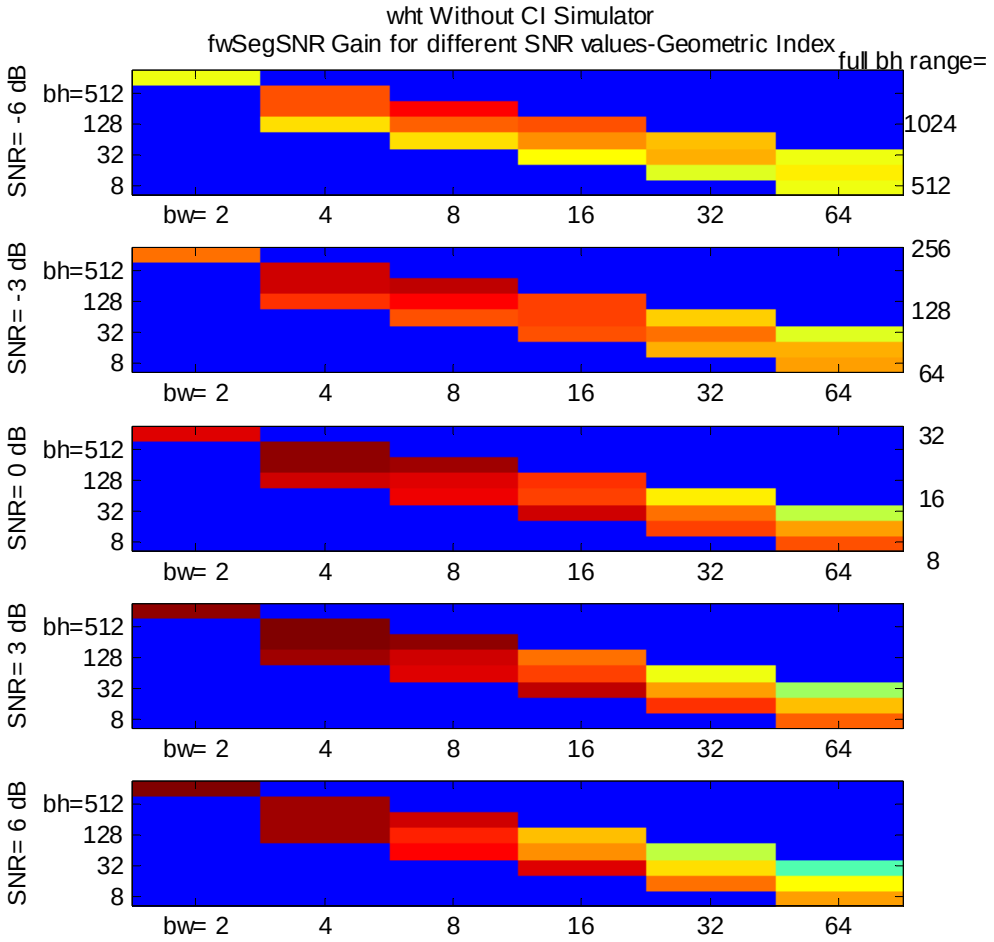


Figure 19: fwSegSNR gain for white noise without the CI Simulator with respect to the patch height and width.

To illustrate the objective measure with respect to the geometric index, figures like Figure 19 could alternatively be used instead of figures of the form of Figure 18. In Figure 19, the objective measure is presented in relation to both the patch height and width instead of the geometric index of the patch. There, it can be seen that for white noise, as the patch becomes shorter and wider the objective measure is reduced.

FFT Size:

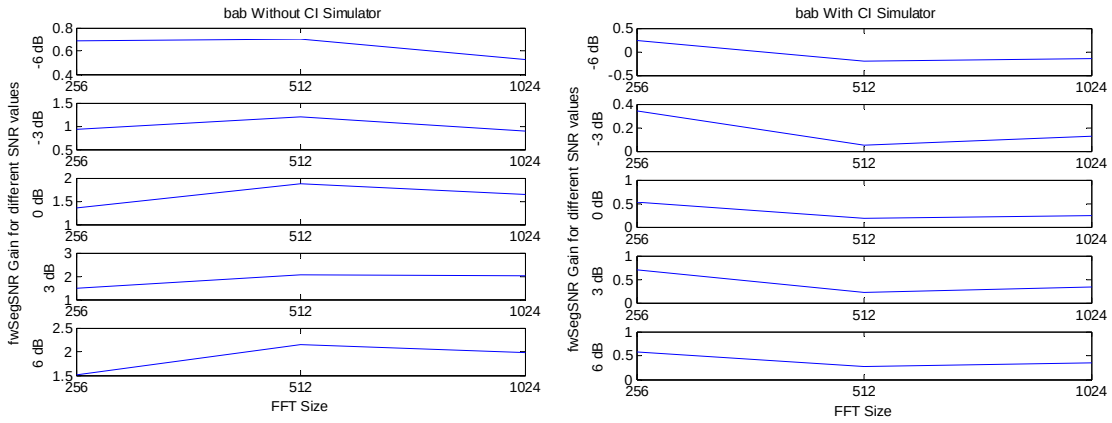


Figure 20: Optimization of FFT Size for babble noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

Figure 20 presents the fwSegSNR gain with respect to the FFT Size for babble noise. The first impression regarding the case without the CI Simulator is that the choice of FFT Size is SNR dependent. A larger SNR in the degraded speech file requires a larger FFT Size according to the objective measure. For this parameter, the objective measure correlates extremely well with subjective observations. Indeed, for 6 dB, an FFT Size equal to 256 leads to a more dull sound and to less speech enhancement. On the other hand, for -6 dB, an FFT Size equal to 1024 might provide a more clear sound, but a smaller FFT Size is preferable in terms of intelligibility. In conclusion, a larger FFT Size results in a more crystal sound, but it does not follow that it also leads to better speech intelligibility.

Furthermore, what is evident is Figure 20, is that unlike in the case of the previous parameters, the optimization pattern for FFT Size is reversed when the CI Simulator is introduced. In other words, a smaller FFT Size appears more preferable than a larger one. This can be subjectively verified for the case of very small SNR. For larger SNR the subjective differences are minor. The ineffectiveness of a large FFT Size in CIs can probably be justified by the fact that a large amount of the spectral information that is gained by using a large FFT during enhancement is lost in the CI, where an FFT of 256 points is applied.

As illustrated in Figure 21, for piano noise a larger FFT is preferred without doubt both without and with the CI Simulator. This can be verified subjectively as well and can be explained by the fact that the capture of the harmonic content by a large FFT is important for piano signals.

For white noise (Figure 22) without the CI Simulator, an FFT Size of 512 is preferred, while when the CI Simulator is used, a smaller FFT is optimal. Indeed, by subjective listening 512 is a good

compromise of intelligibility and artifacts before the Simulator, while after the Simulator an FFT of 256 points results in sharper speech.

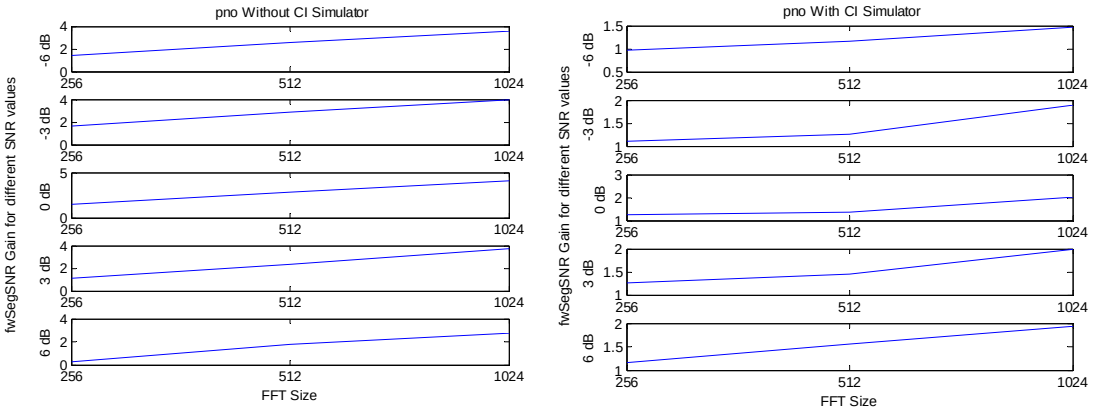


Figure 21: Optimization of FFT Size for piano noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

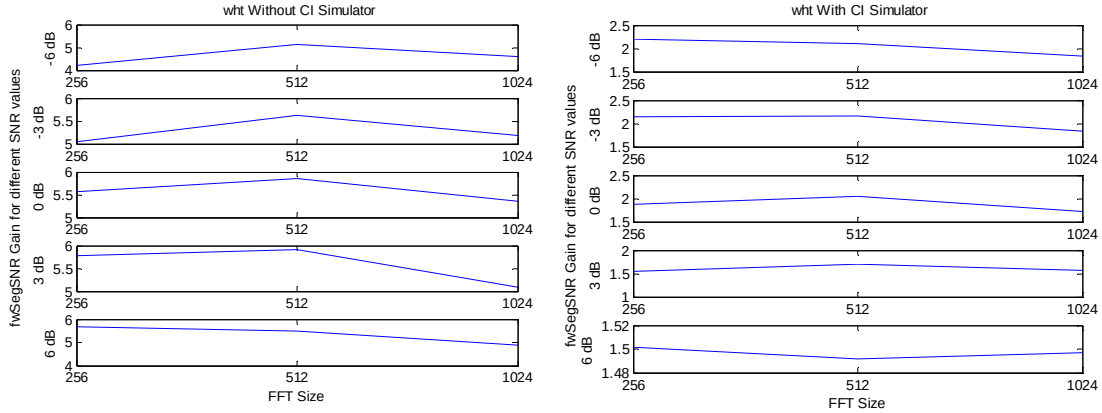


Figure 22: Optimization of FFT Size for white noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

Two suggestions for optimal enhancement of a speech file degraded with babble noise at 0 dB, are provided for before and after the CI Simulator. The parameters that were chosen as optimal are listed in Table 4 together with the resulting fwSegSNR gains. Audio_11 is the degraded speech file. Audio_12 is Audio_11 enhanced with the parameters for "without CI Simulator". Audio_13 is Audio_11 after the CI Simulator. Finally, Audio_14 is Audio_11 enhanced with the parameters for "with CI Simulator", after having gone through the CI Simulator.

	Res_Coh_Thr (μ)	Beta (b)	Geom_Idx	FFT Size	FwSegSNR gain
Without CI Simulator	0.1	0.8	9	512	1.803
With CI Simulator	0.1	1	11	256	0.337

Table 4: Optimal parameters and fwSegSNR gains for speech degraded with babble noise at 0 dB SNR with and without the CI Simulator.

The optimization figures for the remaining 4 types of noise are included in Appendix B.

D. Multiple-files Based Optimization

The aim of this paragraph is to investigate whether the objective results from the parameter optimization based on a single file (III.C) can be generalized. For this reason, 12 clean speech files (2 from 6 speakers), instead of 1, were mixed with 7 different noise types (babble, factory, piano, street, volvo car, white noise, wind) at 0 dB SNR. Again the parameters were treated as independent and varied within the ranges provided in Table 3 (III.C) having the default values of Table 2 (III.C).

A histogram based approach was followed to present the results. For a given parameter, investigated with respect to a certain noise type, a histogram depicts how many times among the 12 files a specific value of the parameter was the first (blue), the second (green) and third (red) in preference. In that way, a visual impression of the optimal parameter among the 12 files is given. Moreover, the total number of files for which a certain value was within the first 3 orders of preference is calculated. If this number is large for a value that possesses the highest first order of preference (highest blue bar), then it can be ensured that even if this value is not the most preferred one for a certain file, it will still result in a good performance. The following figures illustrate the results for all 4 parameters, for 3 types of noise (babble, piano, white), both before and after the CI Simulator.

Residual Coherence Threshold:

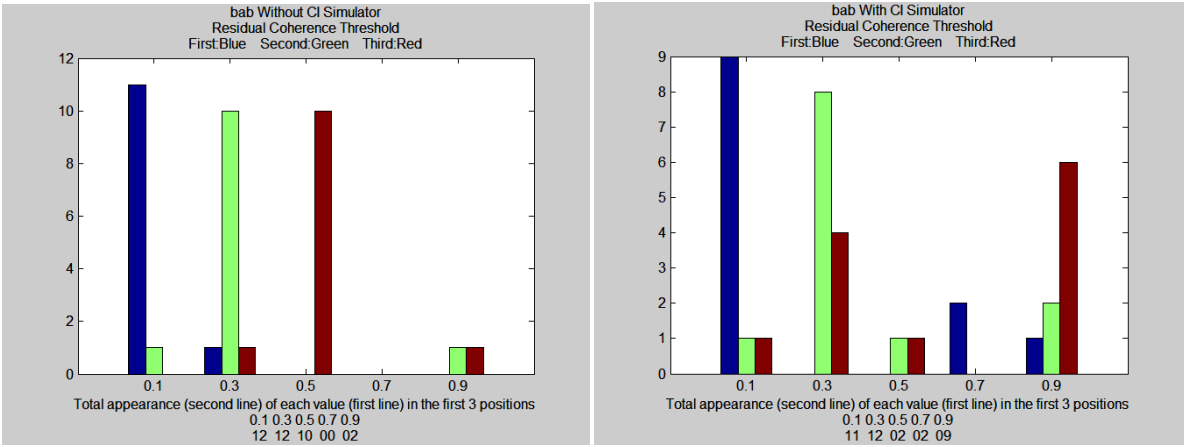


Figure 23: Optimization of Residual Coherence Threshold for babble noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

In Figure 23 left, it is shown that the value of 0.1 for the Residual Coherence Threshold in the case of babble noise without the CI Simulator, was 11 times in the first order of preference (blue) among the 12 files and 1 time in the second order of preference (green). Moreover, for all the 12 files it was always within the first 3 orders of preference as it is written under the histogram. Another value that was preferred, was 0.3. However, it only appeared 1 time in the first order, when 0.1 was second. For all the above reasons, 0.1 is without doubt the optimal value in terms of the objective measure. The selection of 0.1 as the optimal value for the Residual Coherence Threshold agrees with the one-file case of paragraph III.C (Figure 10-left for 0 dB SNR). At this point it should be remarked that in this paragraph only objective evaluation is considered and no subjective listening is involved, as the goal is to generalize the objective results of the previous paragraph (III.C).

When the CI Simulator is used (Figure 23-right), 0.1 is again the value that appears the most times in the first order of preference. However, fewer times (9) than without the CI Simulator (11). The 0.7 is twice the most preferred but appears in the first 3 orders if preference only 2 times. On the other hand, the value 0.3 might never be the most preferred, but it is always within the first 3 orders of preference. Therefore, 0.1 is definitely the optimal value, with 0.3 the second best. The selection of the 0.1 as the optimal value agrees with the one-file optimization results (Figure 10-right for 0 dB SNR).

Figures 24 and 25 present the results for piano and white noise, respectively. It becomes obvious, especially for the case without the CI Simulator, that 0.1 is the optimal value in terms of the objective measure. This is consistent with the results of paragraph III.C (Figures 11 and 12 for 0 dB SNR).

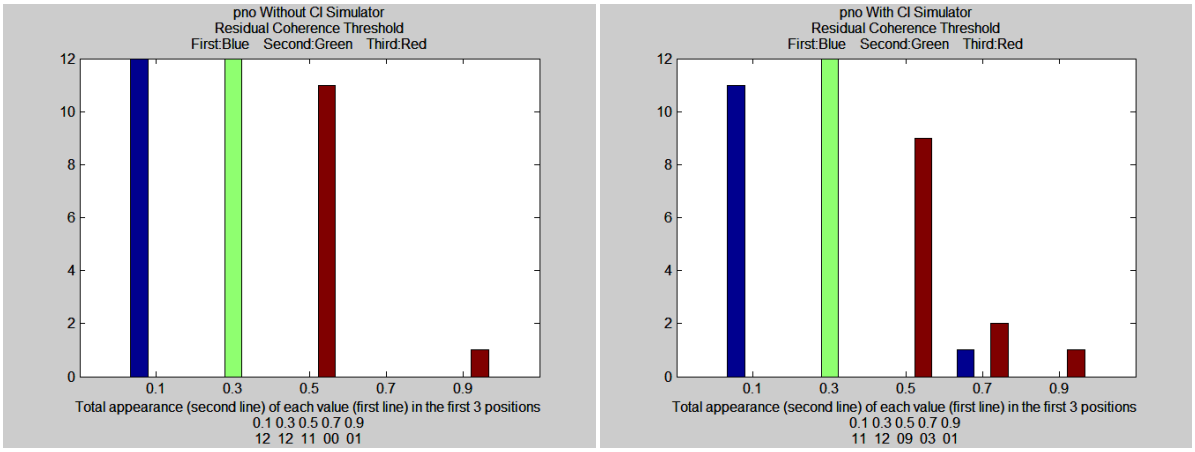


Figure 24: Optimization of Residual Coherence Threshold for piano noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

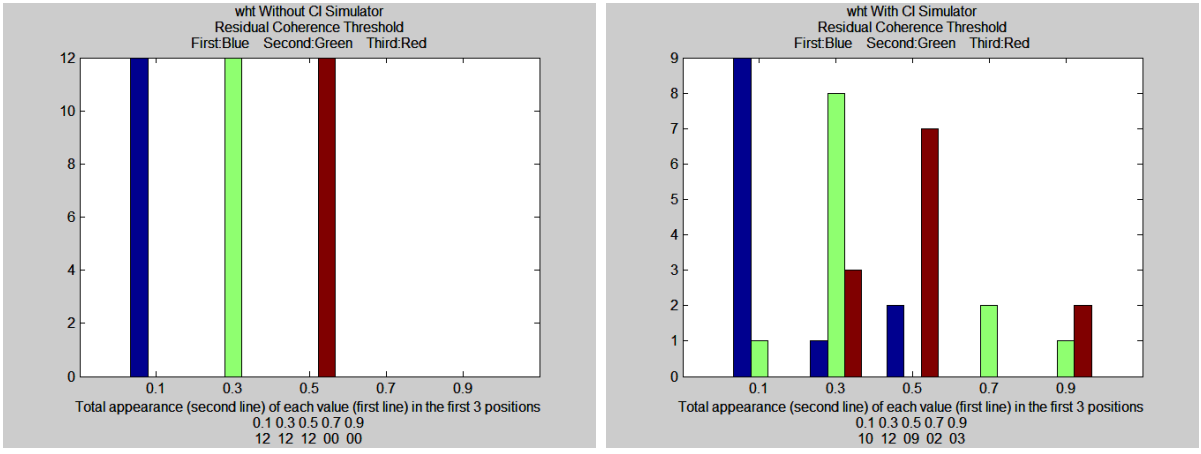


Figure 25: Optimization of Residual Coherence Threshold for white noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

Beta:

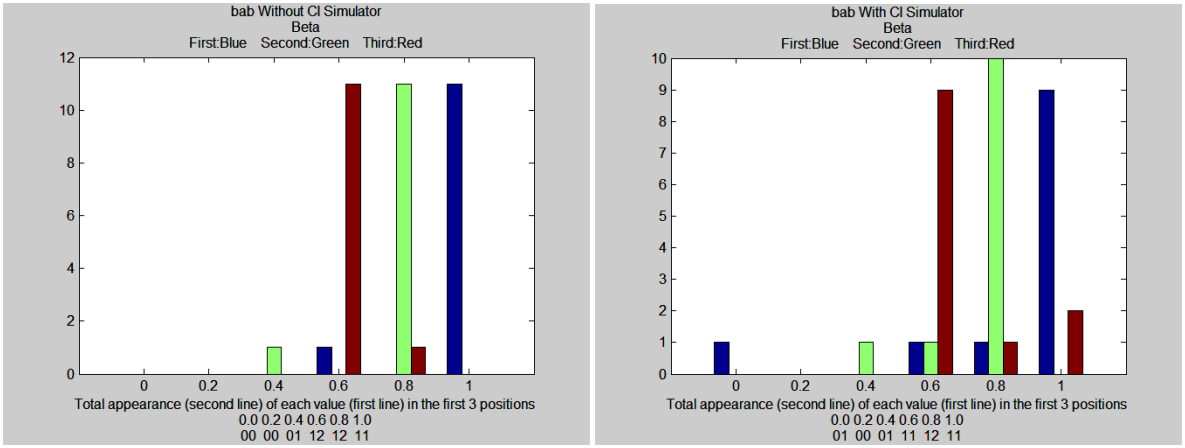


Figure 26: Optimization of Beta for babble noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

From Figure 26, it can be inferred that the optimal value of Beta for babble noise both before and after the CI Simulator is 1. This value was first in preference for 11 and 9 of the 12 files, for the cases without and with the Simulator, respectively. Furthermore, almost for all files (11 of the 12) Beta=1 was within the first 3 orders of preference. The next most preferred values for Beta are 0.8 and 0.6. For the one-file case (Figure 13 for 0 dB SNR), these values appear in the same order of preference. Beta=1 is the optimal choice only in terms of the objective measure. A large value of beta leads to better enhancement, expressed by a large fwSegSNR gain, but also to more artificial sound quality. Therefore, subjective evaluation is required as well in order to detect the optimal trade-off between speech enhancement and sound quality by slightly reducing Beta, especially without the CI Simulator.

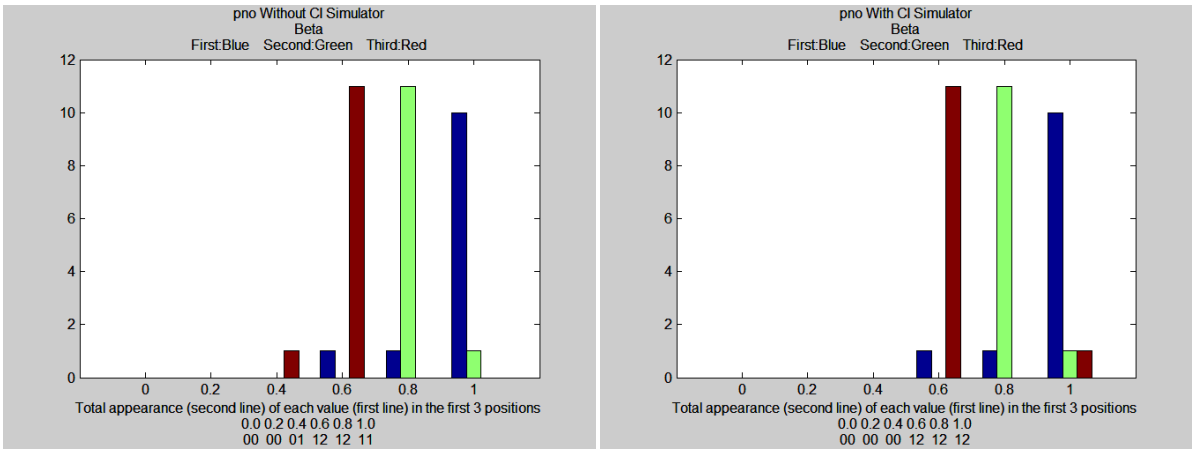


Figure 27: Optimization of Beta for piano noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

For piano noise (Figure 27), the optimal value of Beta is 1 both without and with the CI Simulator. This is consistent with the one-file optimization only for the second case (Figure 14 for 0dB SNR). For the case without the CI Simulator, the optimal value for one file is 0.8 instead of 1. However, on one hand, Beta=1 gives a large value to the objective measure in the one-file optimization and on the

other hand, 0.8 is always within the first 3 orders of preference in the histograms. Therefore, both the one-file and the multiple-files based optimizations reveal the tendency of the objective measure to be maximized for large values of beta. The only difference lies in the optimal value.

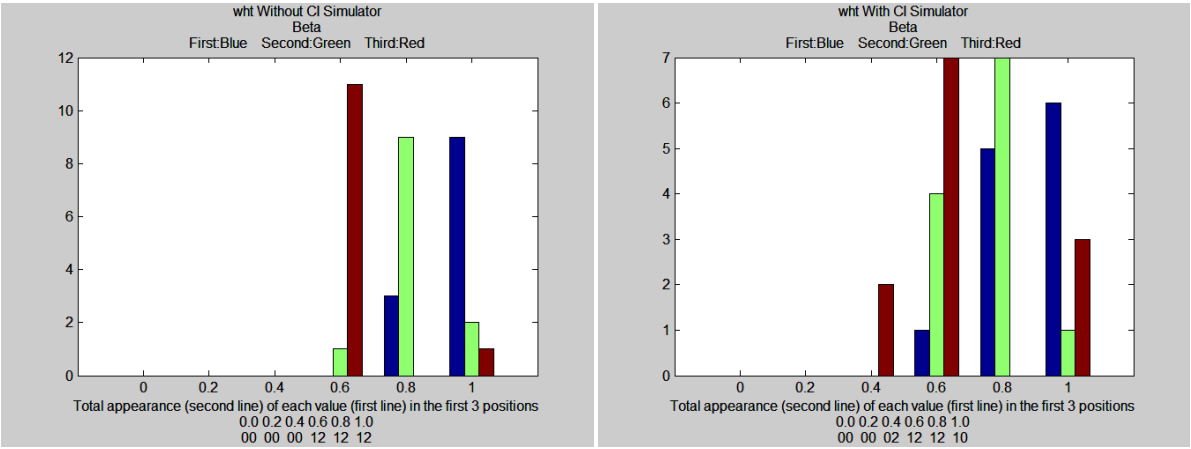


Figure 28: Optimization of Beta for white noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

For white noise without the CI Simulator (Figure 28-left), the optimal value is Beta=1, as in the one-file optimization (Figure 15-left for 0 dB SNR). As far as the optimal Beta when measuring after the CI Simulator (Figure 28-right) is concerned , both 1 and 0.8 appear preferable. Beta=1 overcomes 0.8 at number of appearances in the first position only by 1, while 0.8 is 2 more times within the first 3 positions than Beta=1. Regarding the one-file optimization (Figure 15-right for 0 dB SNR), the values 0.8 and 1 give exactly the same fwSegSNR gain, thus being in agreement with the general results.

Geometric Index:

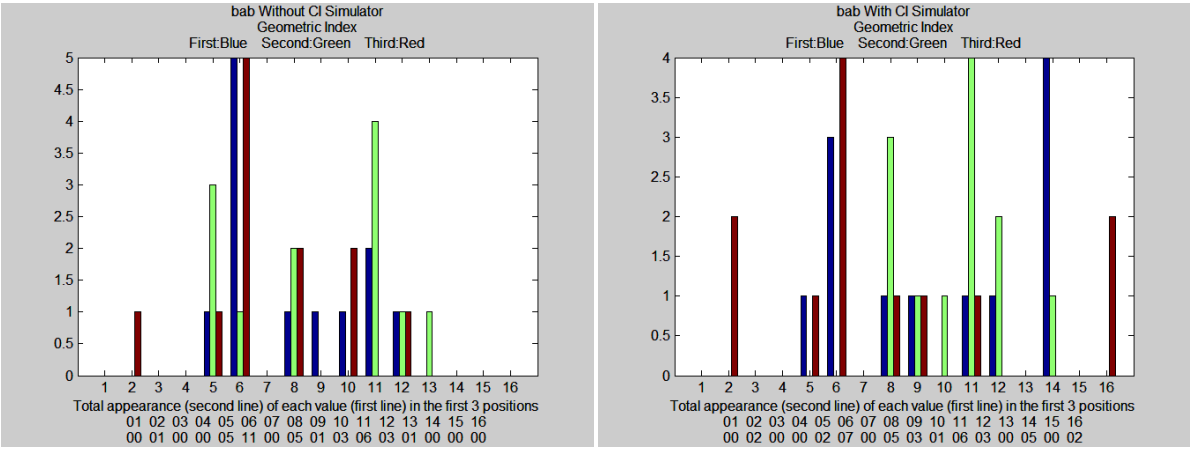


Figure 29: Optimization of Geometric Index for babble noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

From Figure 29 it appears that patch 6 is the optimal without the CI Simulator (6 times in the first position / 11 times within the first 3 positions) and patch 14 is the optimal without the CI Simulator (4 times in the first position / 5 times within the first 3 positions).

However, due to the complexity of choosing the optimal Geometric Index from the histograms, the following method was used to simplify the selection. Every Geometric Index was assigned a score based on the orders of preference where it appeared for all the 12 files. The assignment of scores took place according to the following rule: 16 points were given to the first position, 15 to the second and etc. The last position (16th) was given 1 point. For example, if a patch appeared 3 times in the first position, 2 in the second and 1 in the seventh, it would get a total score of 3x16+2x15+1x10=88. The scores of all the patches (1-16) are presented in Figure 30 both before and after the CI Simulator for babble noise.

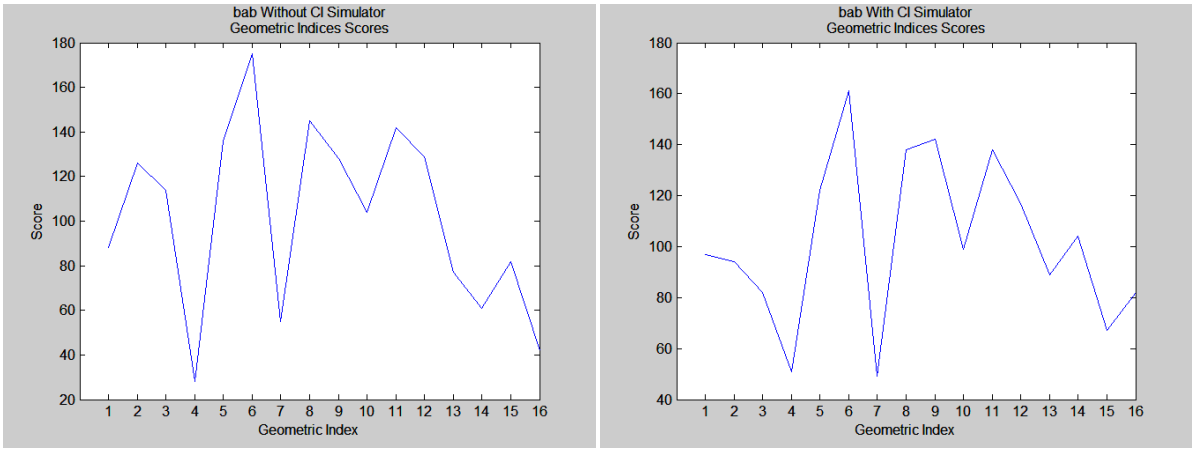


Figure 30: Patch scores for babble noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

Figure 30-left exhibits a very similar pattern to Figure 16-left (for 0 dB SNR) from the one-file optimization. Patches 4 and 7 are in both figures non-preferable. In addition, patches 1 and 13-16 result in a relatively bad performance in both figures. Very good performance is achieved with patches 5,6,8,9,11 for both cases. Finally, the optimal based on multiple files is patch 6, which also works well for one-file. Therefore, the generalized result can be applied to a single file. When the CI Simulator is used (Figure 30-right), patches 4 and 7 result again in very bad performance. For the one-file optimization (Figure 16-right for 0 dB SNR), the same patches are not preferable. However, the one-file optimization curve is more smooth than the one of Figure 30-right. Patches 9 and 11 that are optimal for one file, also exhibit good performance for multiple files. Finally, patch 6, which is the optimal for multiple files, could be successfully used also for one file.

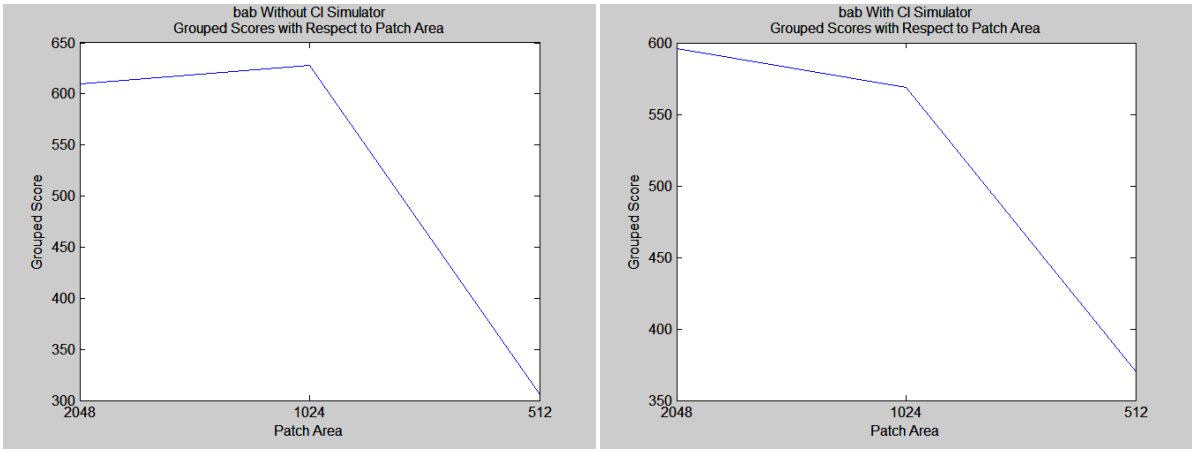


Figure 31: Area-grouped patch scores for babble noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

The scores of the Geometric Indices were also grouped and summed with respect to the patch area. Three groups were formed (area=2048, 1024 and 512). The grouped scores for babble noise are presented in Figure 31 both before and after the CI Simulator. It can be observed that a medium or large area is highly preferred to a small area. A small area is not adequate to capture the content of speech and babble noise. Furthermore, is seems that a larger area is slightly more preferable in the CI case. Indeed, the patches that were selected in paragraph III.C are 9 (area=1024) without the Simulator and 11 (area=2048) with the Simulator.

An alternative grouping of the scores of the Geometric Indices was done with respect to the patch width. The groups formed were (width=2-4, 8, 16, 32, 64). The results are presented in Figure 32. Both without and with the CI Simulator, a width of 16 points is preferred and a width of 64 points is extremely unfavorable, as it decreases the temporal resolution. Patch 9 that was selected for one file without the Simulator has, indeed, a width of 16 points. Patch 11 that was selected for one file with the Simulator has 32 points, the second best choice for multiple files.

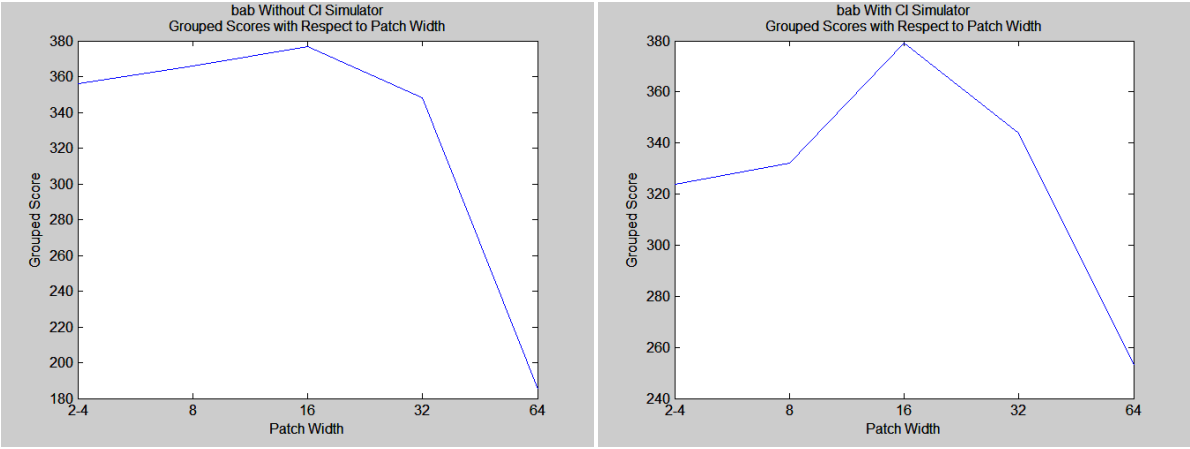


Figure 32: Width-grouped patch scores for babble noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

The results of the Geometric Index optimization based on multiple files are presented for piano noise in the following figures.

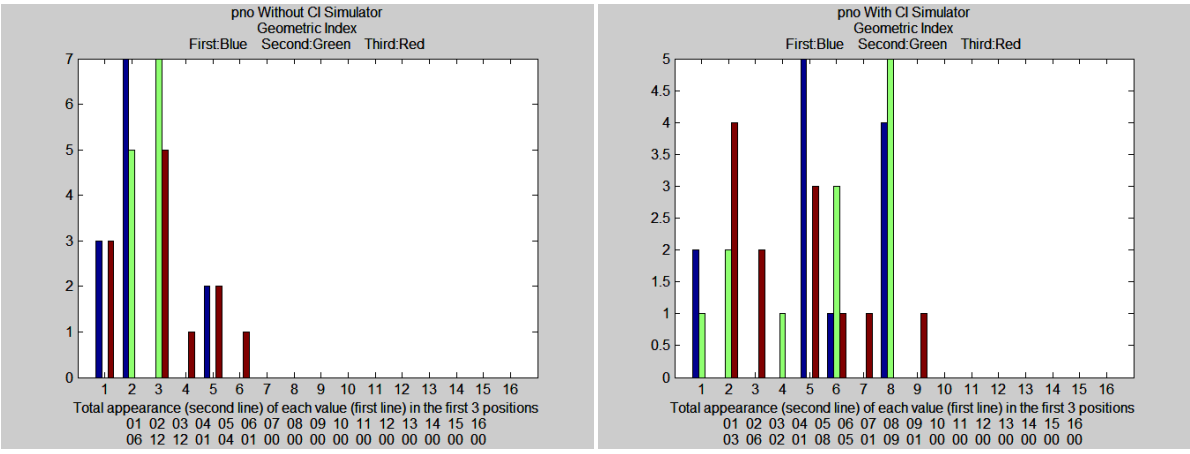


Figure 33: Optimization of Geometric Index for piano noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

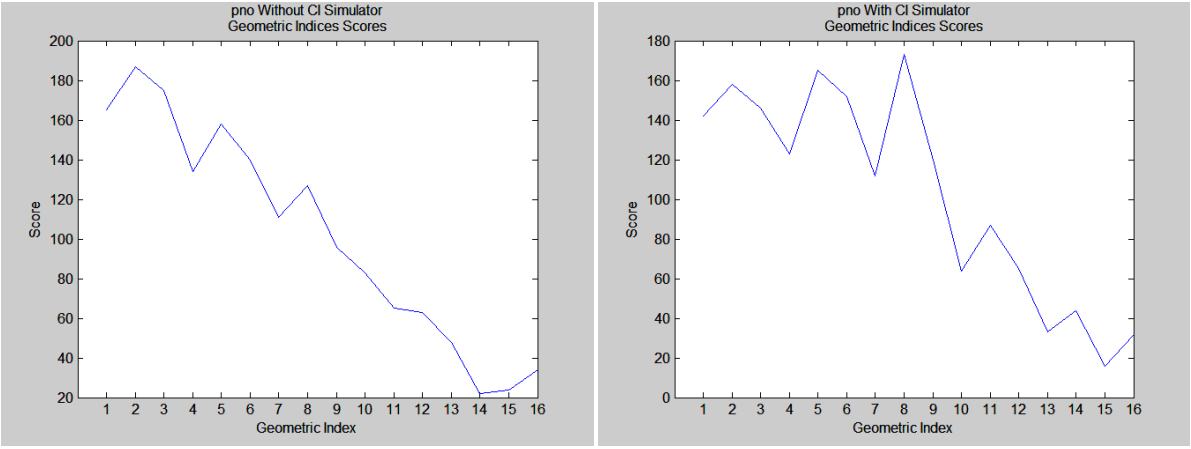


Figure 34: Patch scores for piano noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

From Figure 34, it can be inferred that patch 2 is the optimal without the CI Simulator and patch 8 is optimal with the CI Simulator. These results are in absolute agreement with the one-file optimization (Figure 17 for 0 dB SNR).

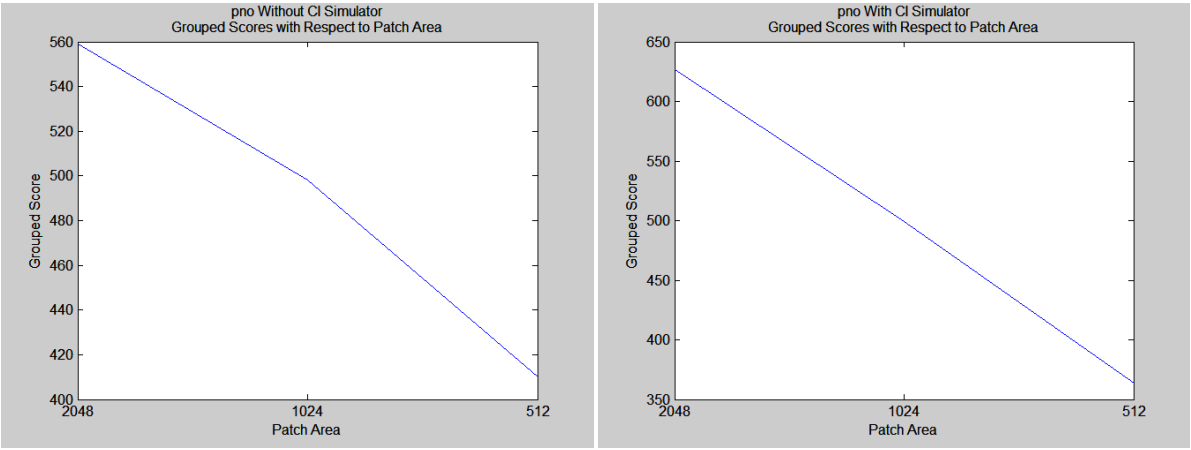


Figure 35: Area-grouped patch scores for piano noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

A large area is highly preferred in the case of piano noise as it can be seen in Figure 35. Both patches 2 and 8 fulfill this criterion as they have an area equal to 2048. The patch that was subjectively chosen for one-file with the CI Simulator, patch 2, has an area of 2048. However, patch 6 that was chosen subjectively for one-file without the CI Simulator, has an area of 1024. Patch 6 is slightly shorter and wider than patch 2 and it seems that it fits better the characteristics of speech. It was subjectively preferred, because for patch 6 the speech sounds more sharp.

By taking Figure 36 besides Figure 35 into account, it can be inferred that a tall and narrow patch favors the objective measure. This was also valid for the one-file optimization. For piano noise, it is more important to capture the spectrum than the temporal information.

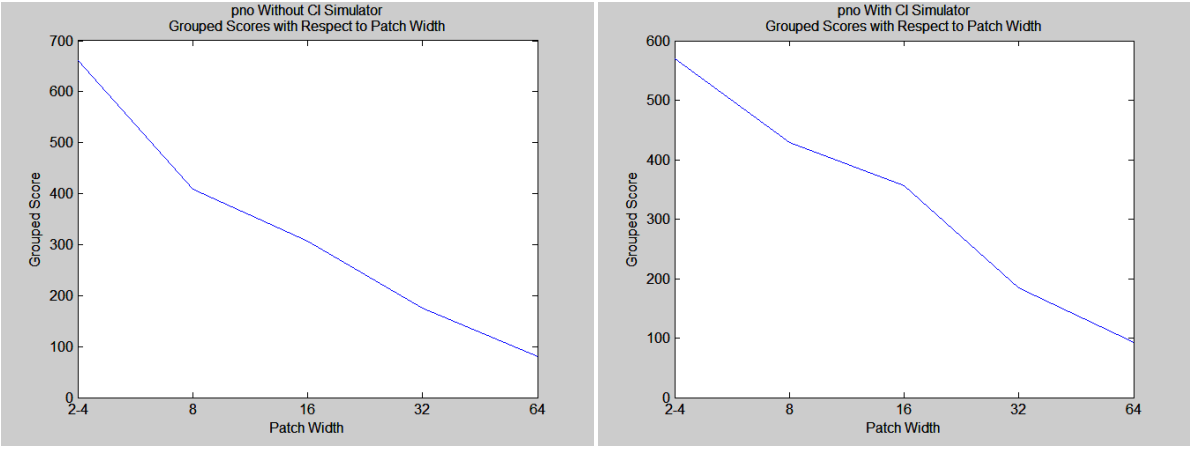


Figure 36: Width-grouped patch scores for piano noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

The results of the Geometric Index optimization based on multiple files are presented for white noise in the following figures.

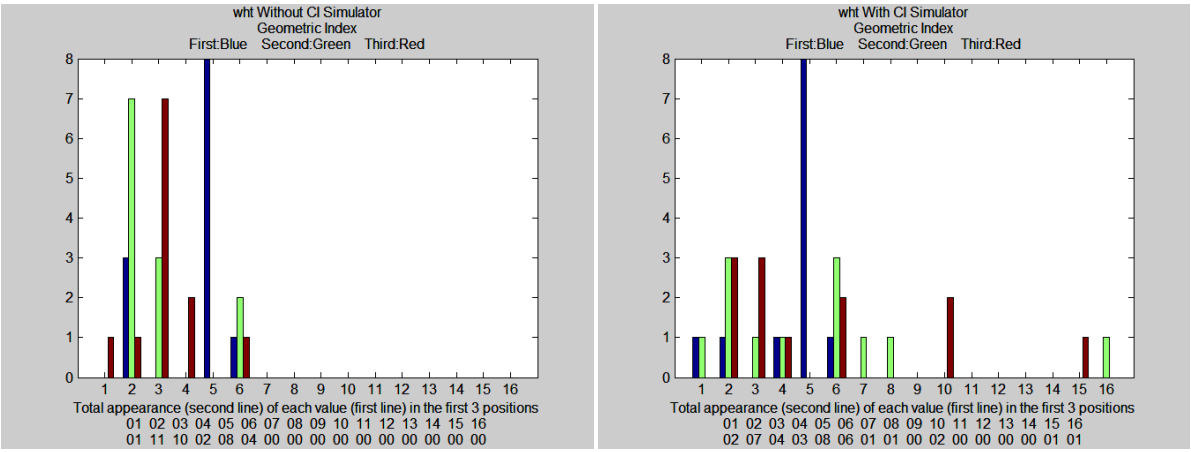


Figure 37: Optimization of Geometric Index for white noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

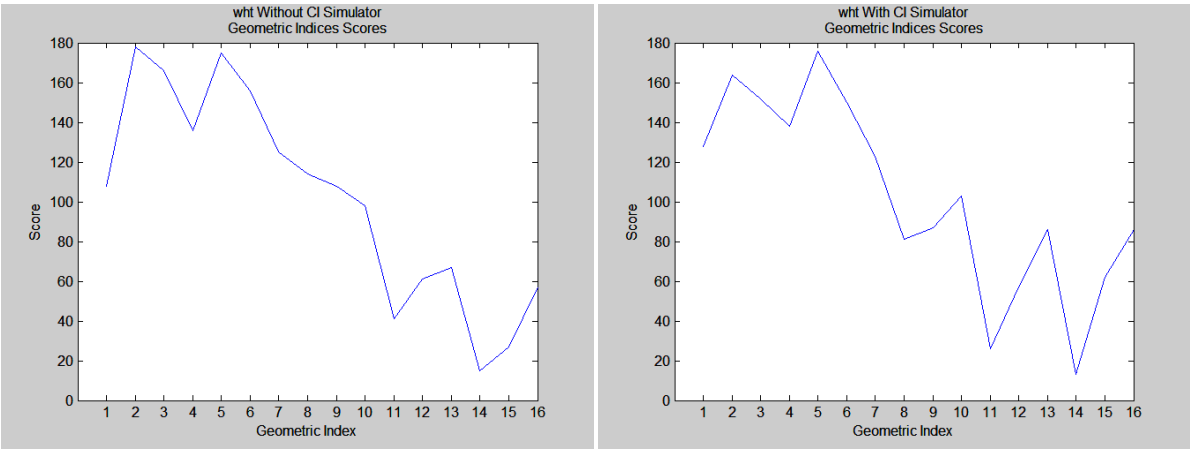


Figure 38: Patch scores for white noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

Like in the one-file optimization, the patches that favor the objective measure are 2 and 5. Therefore, the one-file results in terms of the fwSegSNR can be very well generalized.

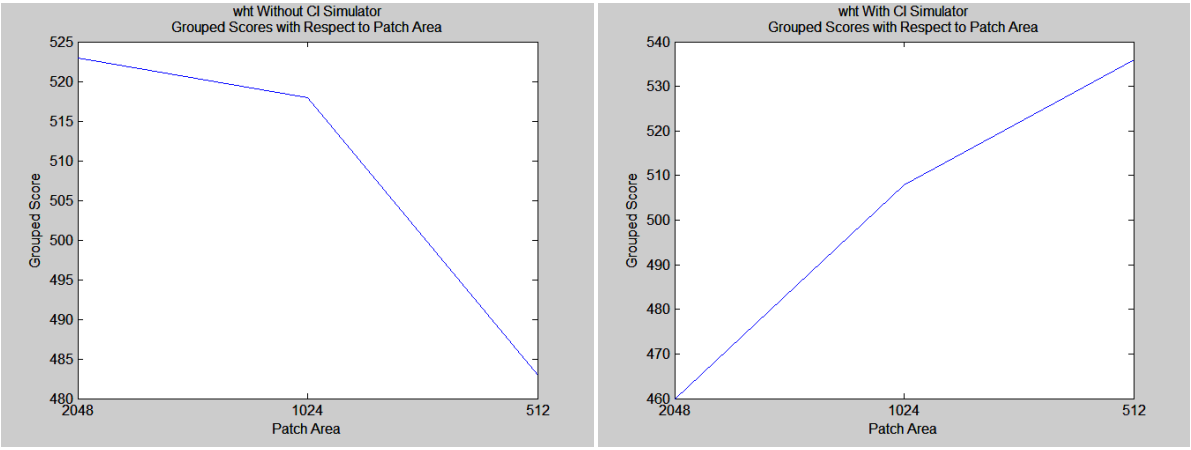


Figure 39: Area-grouped patch scores for white noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

As shown in Figure 39, a larger patch area leads to a better performance without the CI Simulator. This pattern is reversed when the CI Simulator is introduced. These results agree with the one-file optimization only for the case without the CI Simulator, as both patch 2 (optimal without Simulator) and 5 (optimal with Simulator) have an area equal to 2048.

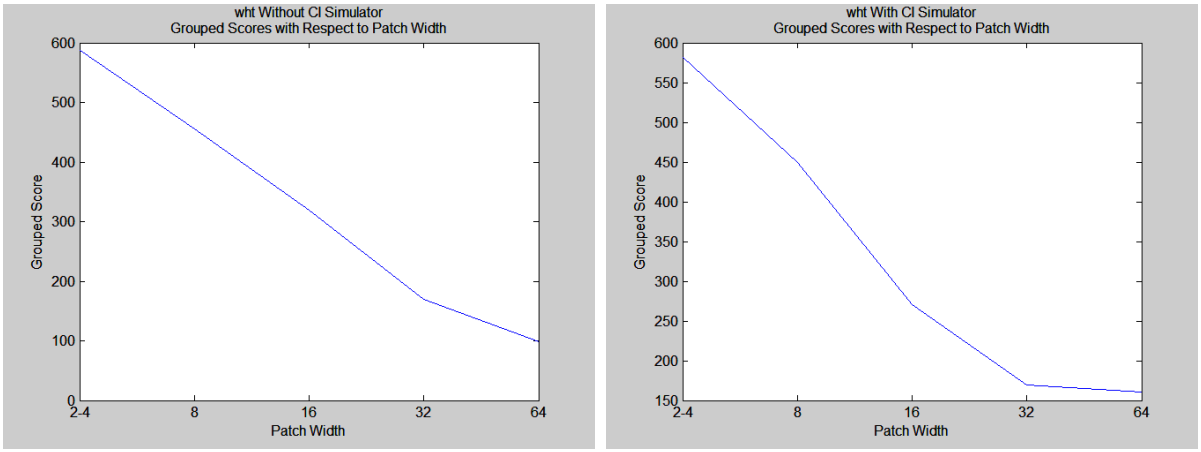


Figure 40: Width-grouped patch scores for white noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

Figure 40 shows that a small patch width favors the objective measure. This is compatible with the objectively selected patches 2 and 5 of width equal to 4 and 8, respectively. However, patch 8 that was selected by listening for without the CI Simulator, has a larger width equal to 16. What makes patch 8 subjectively optimal is that its smoothes an unwanted artifact that appears for patches like 5, which maximizes the objective measure.

FFT Size:

The optimization of FFT Size for babble noise, based on 12 files is presented in the following figure.

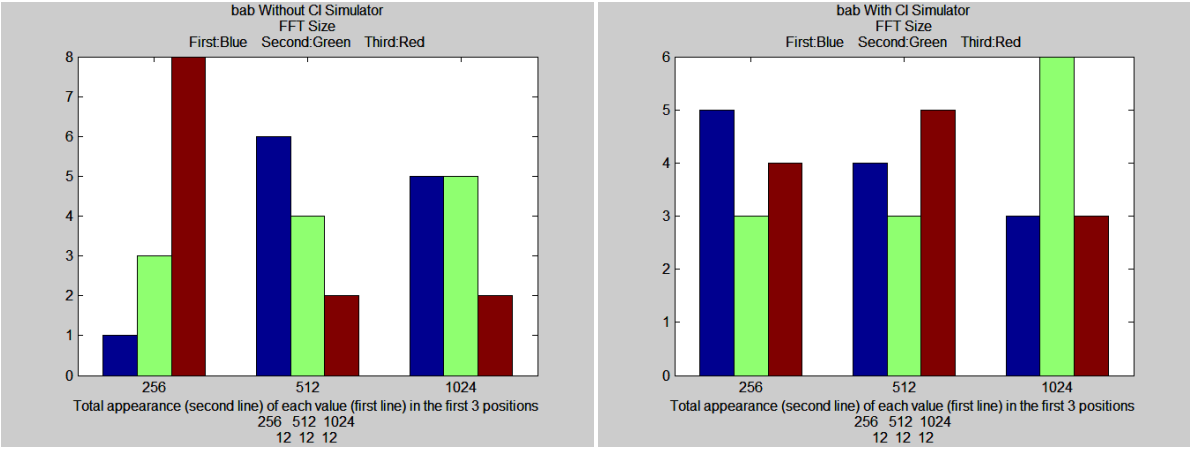


Figure 41: Optimization of FFT Size for babble noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

In Figure 41, it can be observed that the order of preference of FFT Size is 512,1024 and 256 without the CI Simulator and 256, 512, 1024 with the CI Simulator. This order is exactly the same as in the one-file optimization (Figure 20 for 0 dB SNR). The effectiveness of a 256 FFT for the CI case contrary to the case without the CI Simulator, is also verified in the multiple-files optimization, similarly to the one-file optimization.

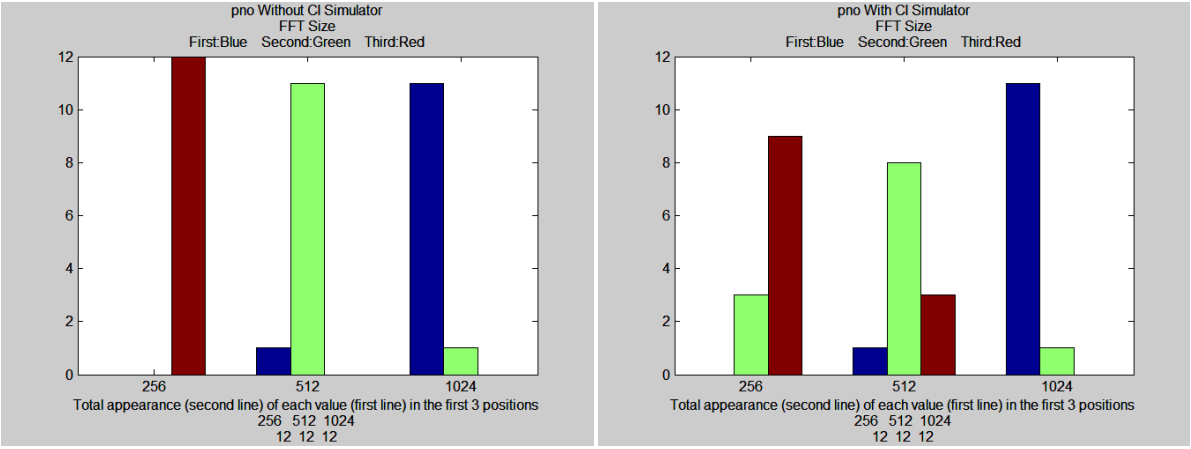


Figure 42: Optimization of FFT Size for piano noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

From Figure 42 it is clear that a large FFT is optimal for piano noise both without and with the CI Simulator. The same was derived from Figure 20 of the one-file case. As the spectral information is of high importance for piano noise, a large FFT is required.

In Figure 22 of the one-file case, the order of preference is 512, 256, 1024 both without and with the CI Simulator. The same applies also in the multiple-files case, as illustrated in Figure 43.

In conclusion, in this paragraph, it has been shown that the optimization results when 12 files were used, correlate very well with the corresponding results of one-file. Of course, it has to be clarified that the aforementioned has been investigated in terms of the objective measure. However, for

defining the optimal parameters, subjective evaluation is required as well, since the objective measure is not powerful in revealing quality artifacts than arise with certain parameterizations.

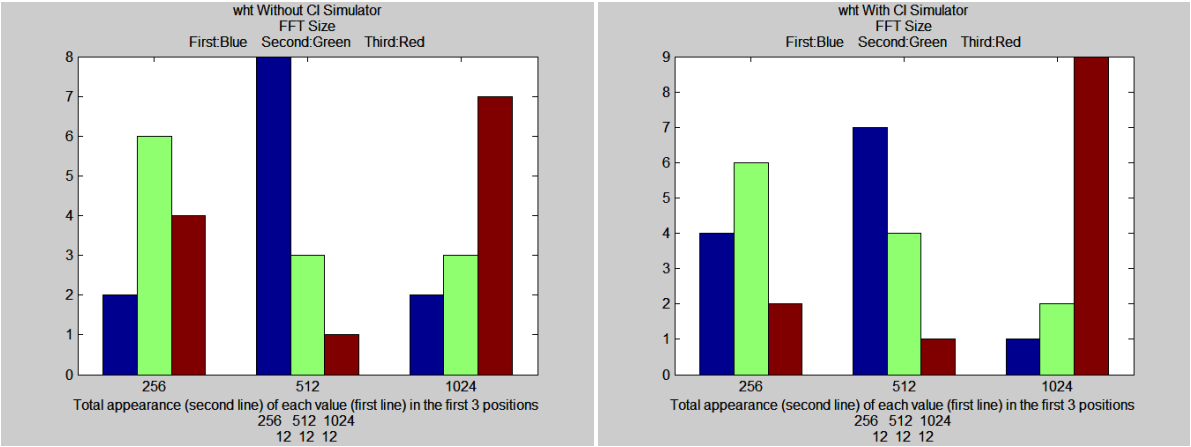


Figure 43: Optimization of FFT Size for white noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

A summary of the optimal parameterization for all the noise types, based on the multiple-files approach, for 0 dB SNR, for both without and with the CI Simulator and in terms of the objective measure is presented in Tables 5 and 6.

	Res. Coh. Thr.		Beta		Geom. Index		FFT Size	
	no CI	CI	no CI	CI	no CI	CI	no CI	CI
bab	0.1	0.1	1	1	6	6	512	256
fct	0.1	0.1	1	1	6	10	512	512
pno	0.1	0.1	1	1	2	8	1024	1024
str	0.1	0.1	1	1	12	8	512	512
vlv	0.1	0.1	1	1	8	12	1024	1024
wht	0.1	0.1	1	0.8	2	5	512	512
wnd	0.1	0.1	1	1	5	5	512	512

Table 5: Parameter optimization in terms of fwSegSNR based on multiple files for 0 dB SNR.

	Patch Area		Patch Width	
	no CI	CI	no CI	CI
bab	1024	2048	16	16
fct	1024	512	8	8
pno	2048	2048	2-4	2-4
str	1024	2048	16=32	32
vlv	1024	512	16	16
wht	2048	512	2-4	2-4
wnd	1024	2048	2-4	2-4

Table 6: Patch area and width optimization in terms of fwSegSNR based on multiple files for 0 dB SNR.

The same experiment was also repeated for a low SNR (-6 dB). The results are presented in Tables 7 and 8 similarly to Tables 5 and 6. The values that differ from the corresponding ones for 0 dB are indicated with red color.

	Res. Coh. Thr.		Beta		Geom. Index		FFT Size	
	no CI	CI	no CI	CI	no CI	CI	no CI	CI
bab	0.1	0.1	1	0.8	6	6	512	1024
fct	0.1	0.1	1	0.8	6	6	1024	1024
pno	0.1	0.1	1	1	2	8	1024	1024
str	0.1	0.1	1	1	12	5	512	512
vlv	0.1	0.1	1	1	12	8	1024	1024
wht	0.1	0.1	1	0.8	5	5	512	512
wnd	0.1	0.1	1	1	12	5	512	512

Table 7: Parameter optimization in terms of fwSegSNR based on multiple files for -6 dB SNR.

	Patch Area		Patch Width	
	no CI	CI	no CI	CI
bab	2048	2048	2-4	16
fct	1024	512	16	8
pno	2048	2048	2-4	2-4
str	2048	2048	32	64
vlv	1024	512	16	16
wht	1024	1024	2-4	2-4
wnd	2048	2048	64	64

Table 8: Patch area and width optimization in terms of fwSegSNR based on multiple files for -6 dB SNR.

E. Alternative Objective Measures

Besides the fwSegSNR that was described in paragraph III.B, there is a wide range of objective measures that can be used for the evaluation of the Speech Enhancement algorithm. The most widely spread ones are the Cepstrum Distance [11], which provides an estimate of the log spectral distance between two spectra, the Weighted Spectral Slope [12], which computes the weighted difference between two spectral slopes in each frequency band, the Itakura-Saito distance [11], which is a measure of the perceptual difference between two spectra, the Overall SNR, the time-domain Segmental SNR [13] and, finally, PESQ [14], which is a model of subjective quality.

Figure 44 depicts all the above objective measures with respect to the 4 optimization parameters. Withinin each of the 4 subfigures, the enhancement performance is presented with regard to the same degraded speech file. Better enhancement is expressed by a large value for fwSegSNR, PESQ, Overall SNR and Segmental SNR, while for Cepstrum Distance, Weighted Spectral Slope and Itakura-Saito Distance, a small value is an indication of good performance.

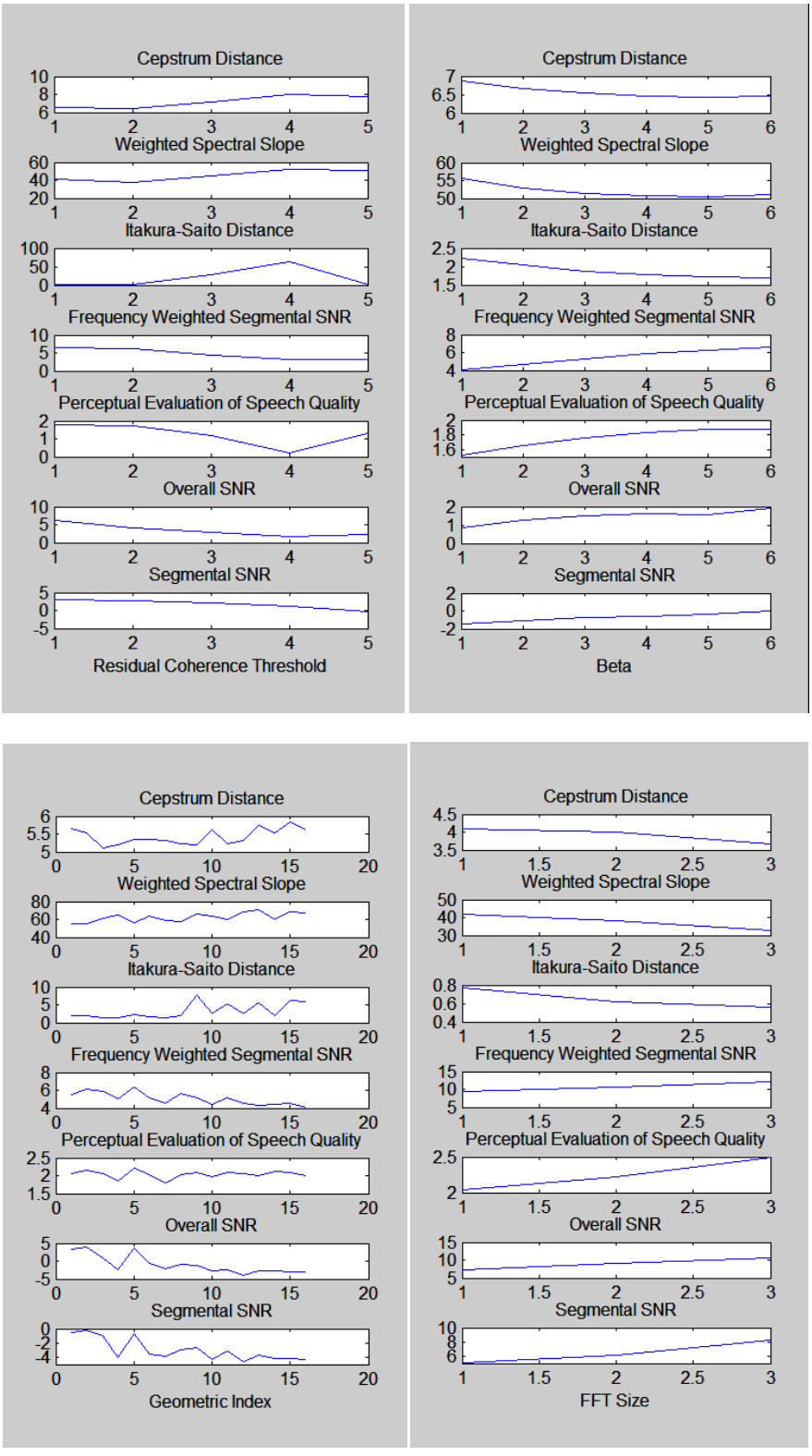


Figure 44: Parameter optimization with various objective measures.

This paragraph focuses on the PESQ (Perceptual Evaluation of Speech Quality). This measure was developed for the assessment of the end-to-end quality of narrowband telephone networks and speech codecs. It correlates well with subjective evaluation of speech distortion, noise distortion and overall quality. By properly selecting its parameters, it can emphasize on the desired quality criterion mentioned before. The PESQ is also recommended by ITU-T.

Here, PESQ is compared to fwSegSNR in measuring the performance of the Speech Enhancement algorithm with respect to the 4 optimization parameters. Similarly to paragraph III.D, 12 speech files were mixed with babble noise at -6 dB, 0 dB and 6 dB SNR. Both the fwSegSNR gain and PESQ gain were measured after enhancing the aforementioned files with the algorithm under investigation. The 4 parameters varied independently within the ranges of Table 3, having the default values of Table 2. The following figures present the results with respect to the 4 parameters, after averaging each measure for the 12 files. Both before and after the CI Simulator results are depicted. The markers indicate the maxima (objectively optimal values).

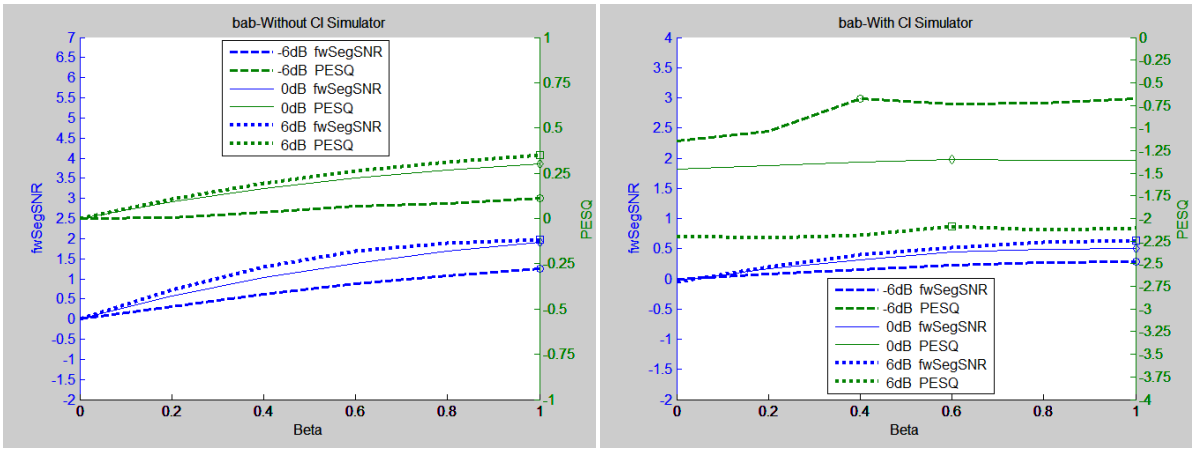


Figure 45: Comparison between fwSegSNR amd PESQ for babble noise with respect to Beta. (Left): without the CI Simulator. (Right): with the CI Simulator.

In Figure 45, it can be observed that when the CI Simulator is not used, both objective measures produce consistent results. However, with the CI Simulator, while fwSegSNR shows that a larger SNR leads to better enhancement, as shown also without the Simulator, PESQ reverses this principle. Furthermore, PESQ prefers Beta=0.4 (green circle) instead of 1 (blue circle) in the CI case. A smaller value for Beta results in lower noise suppression, but to a more natural and smooth sound. This might explain why it is preferred by a speech quality measure, such as PESQ. By subjective listening, it can be said that Beta=1 is the optimal for CI. Therefore, fwSegSNR is more credible in this case.

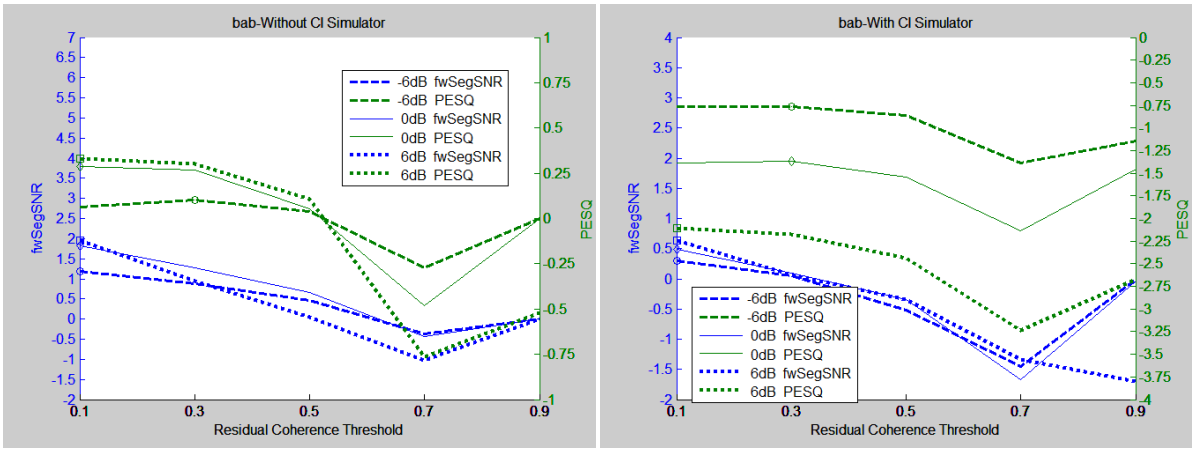


Figure 46: Comparison between fwSegSNR amd PESQ for babble noise with respect to Residual Coherence Threshold. (Left): without the CI Simulator. (Right): with the CI Simulator.

As it can be seen in Figure 46, in general, the two objective measures are in agreement with each other. They both highlight the improvement of performance for small values of the Residual Coherence Threshold and also indicate the value 0.7 as problematic. However, in some cases, the optimal with PESQ is 0.3 instead of 0.1. This probably happens because PESQ is concerned with quality rather than intelligibility. A value of 0.3 leads to an unclear speech, which, nevertheless, contains less source confusion. For this reason, it is slightly preferred by PESQ. The fwSegSNR correlates better with subjective listening, according to which 0.1 is the optimal value for this application.

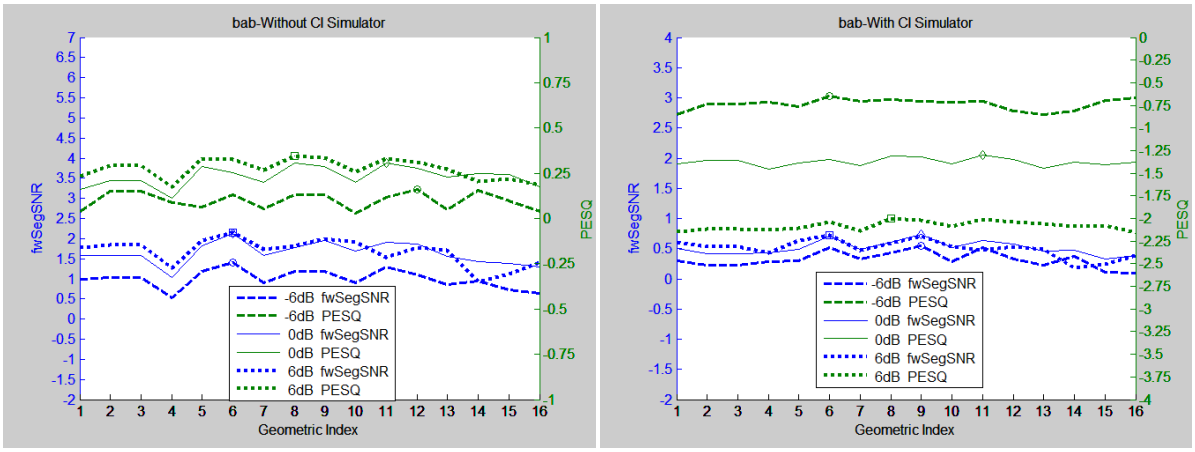


Figure 47: Comparison between fwSegSNR amd PESQ for babble noise with respect to Geometric Index. (Left): without the CI Simulator. (Right): with the CI Simulator.

From Figure 47, it can be derived that although the two objective measures don't indicate the same patches as optimal, they have similar fluctuation patterns (peaks and valleys). Furthermore, according to PESQ with the CI Simulator (Figure 47-right), better enhancement is achieved for worse SNRs on the contrary to fwSegSNR, similarly to Figure 45-right.

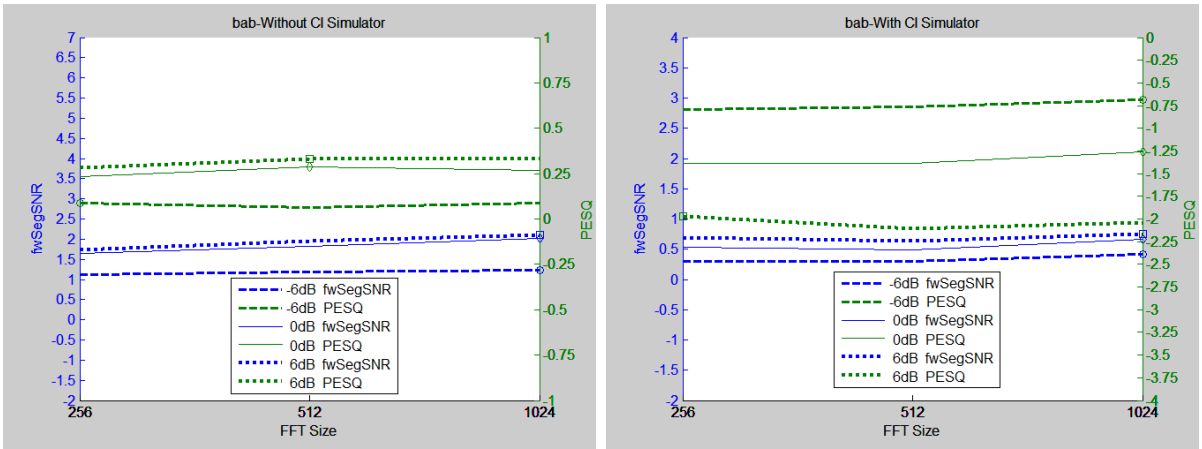


Figure 48: Comparison between fwSegSNR amd PESQ for babble noise with respect to FFT Size. (Left): without the CI Simulator. (Right): with the CI Simulator.

The variation of both the objective measures with respect to the FFT Size is small. This is mainly consistent with subjective evaluation. Again, as observed in Figure 48-right, according to PESQ, a bigger enhancement gain is achieved for low SNRs after the CI Simulator.

In conclusion, fwSegSNR is a more credible objective measure, as it correlates better with subjective evaluation in this application, which focuses more on speech intelligibility improvement by noise suppression and less on the generation of a smooth and pleasant sound, especially after the CI Simulator.

F. Computational Time

The computational time of the algorithm's enhancement step is crucial, as the algorithm is aimed to be incorporated in a real-time system. The "heaviest" computation in this algorithm takes place in the LARC function, which sparsely codes the feature matrix X of the degraded speech file on the dictionary D, as described in II.A.

The processing time in LARC depends on the value of the Residual Coherence Threshold. The smaller the threshold, the longer the time. A small threshold imposes a stricter termination criterion on LARC, which will need more iterations until it stops. Beta, on the other hand, does not affect the computational time. It is just an exponent in the GA instantaneous estimator. In this paragraph, the computational time of LARC will be investigated with respect to the Geometric Index and the FFT Size. Table 1 is presented here as Table 9, for convenience of observation, having added the patch area information.

INDEX	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
HEIGHT	1024	512	256	128	256	128	64	128	64	32	64	32	16	32	16	8
WIDTH	2	4	4	4	8	8	8	16	16	16	32	32	32	64	64	64
AREA	2048	2048	1024	512	2048	1024	512	2048	1024	512	2048	1024	512	2048	1024	512

Table 9: Geometric indices and corresponding patch morphologies.

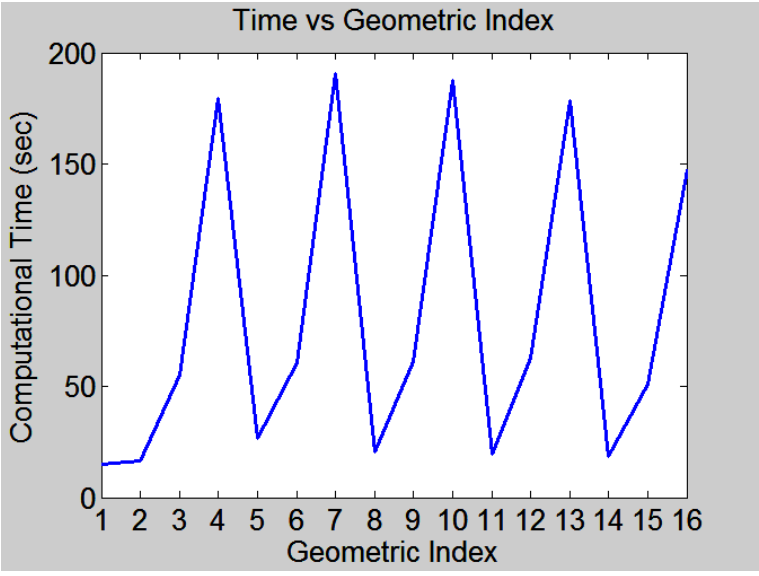


Figure 49: LARC computational time with respect to Geometric Index.

Figure 49 illustrates the computational time of LARC with respect to the Geometric Index. It is obvious that there is a pattern which needs to be highlighted. It can be observed that patches 1,2,5,8,11,14, which have a short computational time, are the ones with large area of 2048 (Table 9), while patches 4,7,10, 13,16, which have a long computational time, are the ones with small area of 512. Figure 50 presents the computational time with respect to the patch area, where it is clear that the larger the patch area, the shorter the computational time of LARC. A certain patch area (e.g. 512) is repeated 5 times in Figure 50, as 5 patches have an area of 512 etc.

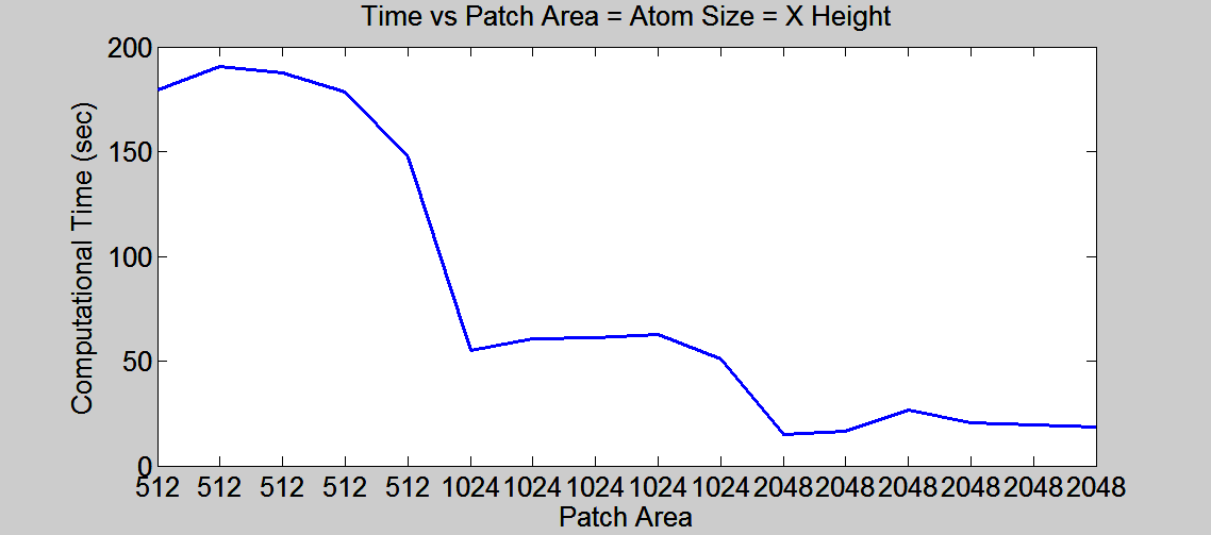


Figure 50: LARC computational time with respect to patch area.

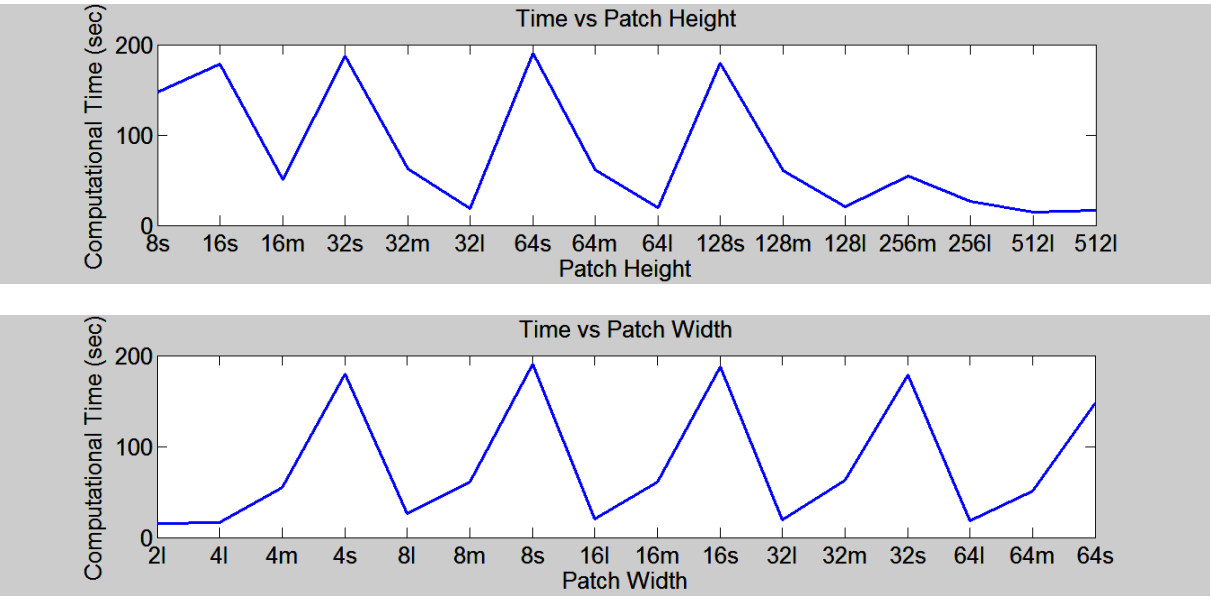


Figure 51: LARC computational time with respect to patch height (UP) and patch width (DOWN).

Figure 51 illustrates how the computational time varies with respect to the patch height and width. The letters on the x axis indicate the area (s: small=512, m: medium=1024 and l: large=2048). The area pattern lies underneath Figure 51, verifying the conclusion derived from Figure 50.

It remains to be explained, why a large area leads to short computational time and vice versa. In the second step of enhancement, before LARC coding, overlapping patches are extracted from the STFT space and vectorized, leading to the formation of X. Because of vectorization, the number of rows in X equals the patch area, as one column of X corresponds to one patch. Furthermore, the number of rows in D equals the patch area too. One would expect that a large patch area would lead to larger computational time, because of the larger height of X. However, this is not valid, because what plays a more important role in LARC's time is the width of X and not its height. The width of X obviously increases as the patch area decreases, because the STFT space is tiled in more blocks. Therefore, a large patch area leads to a small width in X and thus to a short computational time. Figure 52 depicts the computational time with respect to the X width.

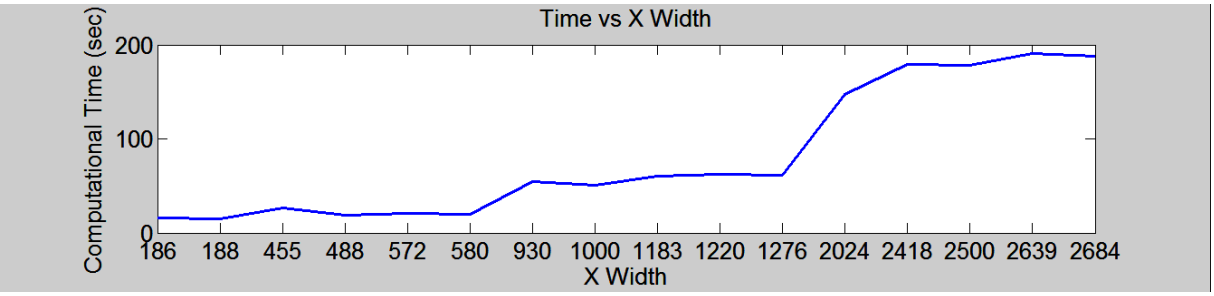


Figure 52: LARC computational time with respect to X width.

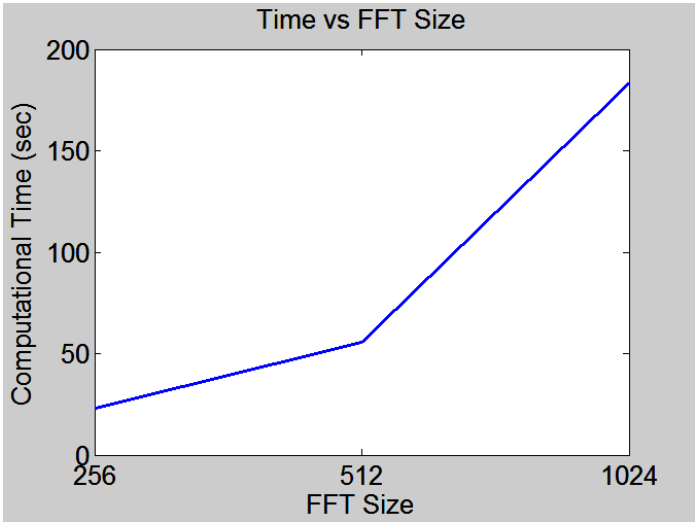


Figure 53: LARC computational time with respect to FFT Size.

Figure 53 shows the dependence of computational time on the FFT Size. It is clear that a large FFT Size leads to a long computational time. This can be justified by the fact that the width of X grows with a larger FFT Size, while its height remains constant and equal to the patch area. The reason why the width of X grows with a larger FFT, is that a larger STFT space needs to be tiled by the same patch. For example, with patch 10, an FFT of 256 leads to an STFT space of 128x194 and to an X of size 512x585. An FFT of 512 leads to an STFT space of 256x192 and to an X of size 512x1305. Finally, an FFT of 1024 leads to an STFT space of 512x189 and to an X of size 512x1684.

G. Optimization Conclusions

Residual Coherence Threshold:

The Residual Coherence Threshold is the termination criterion of LARC and controls the sparsity of coding. A small value leads to dense coding, while a large one leads to sparse coding. In theory, when the coding is too sparse, the speech component of the signal is not adequately represented by the atoms of the speech dictionary, resulting in the so called source distortion. On the other hand, when the coding is too dense, parts of the interferer component of the mixed signal are explained by atoms of the speech dictionary and are thus included in the estimated speech, resulting in source confusion. The Residual Coherence Threshold is involved both in the dictionary training and in the enhancement phase. However, it is not necessary that it has the same value in both phases. Moreover, it plays a more important role in enhancement, for which it was optimized.

As far as the computational time of LARC with regard to this parameter is concerned, a smaller value leads to longer time. Due to the fact that the Residual Coherence Threshold is the termination criterion of sparse coding, in order for LARC to reach a lower threshold, it needs to operate for more iterations, which require longer processing time.

Independently of the noise type and regardless of the SNR of the mixed file, it was proven that a small value of the Residual Coherence Threshold leads to better performance. A value around 0.1 is the optimal. The extreme case of 0.9 is processed very fast, but leads to a very noise outcome, which resembles the mixed file. When values from 0.3-0.7 are used, the enhanced file is not only noisy but also very distorted. This phenomenon reaches its peak for 0.7. Therefore, although values larger than 0.1 are computed faster, the performance is deteriorated so much, that computational time should no longer be taken into account. For values of the parameter smaller than 0.1, there is no considerable performance improvement if not source confusion, while the processing time is dramatically increased.

Finally, the introduction of the CI Simulator does not affect the selection of the optimal value for this parameter.

Beta:

Beta is an exponent inside the instantaneous GA estimator, which is used for filtering after the separation of the speech and interferer estimated components. This parameter is only involved in the enhancement phase. Moreover, it does not affect the algorithm's computational time.

The optimal value for Beta was investigated within the range of [0,1]. For the lower extreme, the enhanced signal is degenerated to the mixed signal, while for the upper extreme, the functionality of the instantaneous GA estimator is cancelled.

Regardless of the noise type and the SNR of the mixed signal, it was shown that a larger value of beta leads to better performance in terms of the objective measure. From this it can be inferred that 1 is the optimal value. However, although beta equal to 1 leads to the largest degree of speech enhancement, sometimes it produces an enhanced sound that is artificial. For this reason, in some cases a smaller value, around 0.8, is preferred, as it produces a more natural outcome. In general, a selection between 0.8-1 for the value of Beta is always safe. Moreover, values larger than 1 were not

thoroughly investigated. Nevertheless, by trying them it was shown that up to a point (e.g. Beta=2) they lead to stronger enhancement and at the same time to an even more artificial outcome.

Finally, when the CI Simulator is introduced, the algorithm can tolerate slightly larger values for Beta. For example, if for a certain file the optimal value without the CI Simulator is 0.8, when the Simulator is used, 1 will probably be the optimal choice.

Geometric Index:

The Geometric Index determines the morphology of the overlapping blocks (patches) that are extracted from the STFT feature space for the formation of the matrix that is sparsely coded. This parameter must have a common value between dictionary training and enhancement, as it specifies the size of the dictionaries. Furthermore, a tall and narrow patch captures better the harmonic content of a signal, while a short and wide one, captures the temporal dynamics of the signal. Finally, the Geometric Index is not only responsible for the degree of speech enhancement, but also for the quality of the outcome. For this reason, subjective evaluation is required as well for the selection of its value.

The optimal Geometric Index is generally independent of the SNR of the mixed file, but highly dependent on the noise type. For babble noise, a very tall and narrow patch, such as 1 (1024x2), produces speech distortion, as the time window is very short to capture the information contained in one word. On the other hand, a too short and wide patch leads to an artificial enhanced file with inadequate representation of its spectral information. Therefore, a patch of average height and width (16-32) is optimal for babble noise. Furthermore, a patch of small area is unfavorable for babble noise. The Geometric Index 9 (64x16=1024) could be a suggested compromise of all the above.

Regarding piano noise, it has been shown that a tall and narrow patch is ideal in this case. For piano noise, the spectral information is more important than the temporal one. Moreover, piano noise requires a patch with large area. Therefore, patch 2 (512x4=2048) could be a good suggestion, or even patch 1 (1024x2=2048) for the CI case, which is a little bit artificial, but also sharp and clear after the CI Simulator. For a shorter and wider patch, such as 6 (128x8=1024), there is less piano noise suppression, but the speech is very natural as this patch represents better the characteristics of the speech signal class.

As far as white noise is concerned, the outputs produced can be distinguished in two categories. In the first category, the speech is sharp and clear, but there is a high frequency artifact introduced. It has been observed that this happens for patches with small area (512), such as 4, 7, 10, 13 and 16. As an example, Audio_15, which has been processed with patch 7, is provided. In the second category, the speech is more dull, but the low frequency artifact that is introduced is less disturbing. This category includes patches with large area (2048), such as 5,8 and 11. As an example, Audio_16, which has been processed with patch 8, is provided. The second category is subjectively preferred both without and with the CI Simulator and objectively without the Simulator. The first category appears as optimal only objectively with the CI Simulator.

The performance of the algorithm in relation to the Geometric Index is often consistent between the case without the CI Simulator and with the CI Simulator. However, the Simulator has the property to diffuse artifacts, thus eliminating especially the low frequency ones.

Finally, a larger patch area is associated with a shorter computational time, as it decreases the width of the matrix that is sparsely coded.

FFT Size

The FFT Size of the STFT transform is critical for the delay of a real time system where this algorithm would be implemented, as it determines the size of the processing window of the input signals. This parameter should have the same value between the dictionary training phase and the enhancement phase, otherwise there will be no correspondence between the input data during enhancement and the dictionaries.

One would expect that a large FFT would lead to better performance. However, this is not the case, with the exception of piano noise. For piano noise, which contains a lot of spectral information, a large FFT is necessary. For babble and white noise, a smaller FFT is adequate and often optimal.

The need for a large FFT Size becomes less important for babble and white noise, when the CI Simulator is introduced. The CI Simulator uses an FFT of 256 points. Therefore, any benefit acquired by applying a large FFT during enhancement, is possibly overlooked in CIs.

As far as the SNR of the mixed signal is concerned, in individual files degraded with either babble or white noise, it has been observed that the selection of the FFT Size is more SNR dependent in comparison to other parameters. However, a generalized rule cannot be derived, as the individual SNR dependencies are dissolved over multiple files.

Finally, the FFT Size influences the computational time of LARC. A larger FFT Size leads to longer time, as it increases the area of the STFT feature space. For piano noise, without doubt a large FFT Size is optimal. However, for babble and white noise, for which both the objective and subjective differences are small when the FFT Size varies, the computational time should be taken into account and thus a smaller FFT Size should often be preferred.

Additional Conclusions:

Within the investigated ranges of variation of the 4 parameters, it has been shown that the performance of the algorithm is more sensitive towards the Residual Coherence Threshold and Beta. Small changes of the value of these parameters induce large differences in the algorithm's performance. The remaining 2 parameters, the Geometric Index and, especially, the FFT Size, have a smaller impact.

The algorithm was evaluated with two objective measures, the fwSegSNR and PESQ. The fwSegSNR is a better indication of the degree of speech enhancement. In this application, where the aim is the improvement of speech intelligibility, this measure correlates better with subjective evaluation.

Finally, the fwSegSNR gain achieved by the algorithm is approximately 2 dB for babble noise, 4 dB for piano noise and 6 dB for white noise. The performance of piano and white noise is better than for

babble noise. On one hand, piano noise is very structured and dissimilar from speech. Therefore, for this noise type, an effective representative dictionary, incoherent to speech, can be trained. On the other hand, white noise is unstructured and incoherent to speech and thus rejected from the speech dictionary, facilitating the separation into the speech and the interferer components during enhancement. When the CI Simulator is introduced, the algorithm's performance is deteriorated in terms of the objective measure, by approximately 0.5 dB, 3 dB and 3.5 dB for babble, piano and white noise, respectively.

H. Further Investigation

It would be interesting to investigate the algorithm's output when clean speech or pure noise is given as an input. When clean speech is given to the algorithm for "enhancement", it performs extremely well. The "enhanced" signal is almost the same as the input signal. On the other hand, when pure noise is given as an input, it is not simply suppressed. Babble and piano noise are reconstructed, as they are structured and explainable by the speech dictionary. However, they sound distorted. Regarding white noise, it is modified by the algorithm. White noise is unstructured and, therefore, it cannot be reconstructed using a speech dictionary. The outcome of enhancement resembles a babble-like artifact.

In a real-time system, prior to applying this algorithm on the input, the noise type would need to be detected, in order to use the corresponding interferer dictionary. For this reason, it is worth to investigate the effect of detecting the wrong noise type. When a dictionary corresponding to a similar interferer signal class is used (e.g. wind instead of car), there is better speech enhancement than when the dictionary of a dissimilar interferer signal class is used (e.g. babble instead of car). However, the performance is much better when the noise type is correctly detected. An interesting observation is that when the correct dictionary is used, there is a babble-like artifact in the enhanced signal, which is much weaker when a wrong dictionary is used. A possible explanation for this could be that for the wrong dictionary, the interferer component of the mixed signal is rejected by the wrong interferer dictionary. Therefore, a larger amount of interferer component is encoded by the speech dictionary and is better represented inside the enhanced signal where it is inevitably included.

IV. CLINICAL TESTS

A. Aim and Description

The Speech Enhancement algorithm was evaluated through adaptive SRT tests, both to CI patients and to NH people using the CI Simulator, conducted in the University Hospital of Zurich.

The Speech Reception Threshold (SRT) is the Signal to Noise Ratio (SNR) that yields 50% intelligibility of speech in noise. In the context of an adaptive SRT test, successive Oldenburg [15] speech sentences of 5 words each, are mixed with noise and the task of the subjects is to detect the words that comprise the sentences. The SNR of the sentences presented decreases, as the intelligibility of the subjects increases (adaptive procedure), until the SRT is calculated. The adaptive SRT tests were conducted using the MACarena software, developed in the USZ.

The goal of the study was to measure the SRT improvement, when the noisy sentences are enhanced with the algorithm, prior to being presented to the subjects. Furthermore, two different parameterization sets of the algorithm were compared with each other. Therefore, in total three enhancement conditions were tested:

1) Parameterization Set 1 (“SpEnh1”): Strong parameterization leading to better enhancement, but also to the generation of artifacts.
2) Parameterization Set 2 (“SpEnh2”): Soft parameterization leading to worse enhancement, but also to a more smooth result.
3) No enhancement at all (“Unprocessed”): Reference condition.

The purpose of including NH subjects besides CI patients in the study was twofold. On one hand, the effectiveness of the CI Simulator could be evaluated by comparing the results between the two groups of subjects. On the other hand, tests with NH people contributed to the familiarization with the procedure and to the realization of possible deficiencies of the test setup. Therefore, they served as a necessary preparation step before starting the tests with CI patients.

More specifically, the Oldenburg sentences were mixed with 3 different noise types: babble, piano and white noise. More details about the noises can be found in the next paragraph, IV.B. The SNR levels supported for the mixing lay within the range of [-10,10] dB with 1dB step. In order to achieve the desired SNR, the speech level was adjusted while the noise level remained constant at 65 dB. All the mixed files were preprocessed by the enhancement algorithm for both parameterization sets. In addition, the files of all 3 enhancement conditions were processed by the CI Simulator in order to be presented to NH subjects.

A test session, corresponding to one subject, consisted of 9 adaptive SRT tests (combination of 3 noise types with 3 enhancement conditions). For each test, 30 noisy sentences were presented to the subject. Nine lists, each one containing 30 sentences, were available for the entire study. The audio files were presented to the subjects via loudspeaker.

The lists of sentences corresponding to specific tests were randomized among the subjects, in order to avoid dependencies on the testing material. Furthermore, the presentation order of the tests was randomized among the subjects as well, in order to eliminate training effects. The restriction of successively presenting the 3 conditions belonging to the same noise type was maintained.

In total, 5 CI patients and 6 NH people contributed to the study. The CI patients used their own CI speech processor, set to their preferred everyday program. The "patient information sheet" can be found in Appendix C.

The mixed files were also enhanced by the Noise Canceller of Phonak, forming a 4th enhancement condition. However, it was decided not to investigate this condition, in order to keep the extent of the study relatively limited.

B. Noises Characteristics

A standard adaptive SRT test is conducted with the Oldenburg noise (olnoise). This noise file is generated by randomly superimposing the speech sentences. For this reason, the long-term spectrum of the sentence material, is very similar to the spectrum of the olnoise.

However, in this study, instead of this noise type, babble, piano and white noise was used. Babble noise originated from the NOISEX-92 corpus [16], piano noise was obtained from a proprietary corpus with location recordings and, finally, white noise was Gaussian. The same noises were also used for the parameter optimization and were provided by the developers of the Speech Enhancement algorithm.

Three noise files were, therefore, randomly segmented in order to degrade the speech files: a 19 seconds file of babble noise, a 6 seconds file of piano noise and a 6 seconds file of white noise. These files were first resampled from 44100 Hz to 22050 Hz, in order to match the sampling frequency of the speech files, and then calibrated, by adjusting their RMS power level to -22 dB.

The long-term and modulation spectra of the three aforementioned noises are presented in Figures 54 and 55, in comparison to the olnoise. For the calculation of the modulation spectrum, the signals were first half-wave rectified. Then their envelope was computed. Finally, the amplitude spectrum of the envelope was calculated after applying a Hanning window. The time window of the noises illustrated in both figures is of 6 seconds. The long-term spectrum is depicted in Figure 54 also in comparison to a speech sentence of 2 seconds. Figure 54 was created with Adobe Audition. Finally, the modulation spectrum is presented (Figure 55) from 1 to 100 Hz.

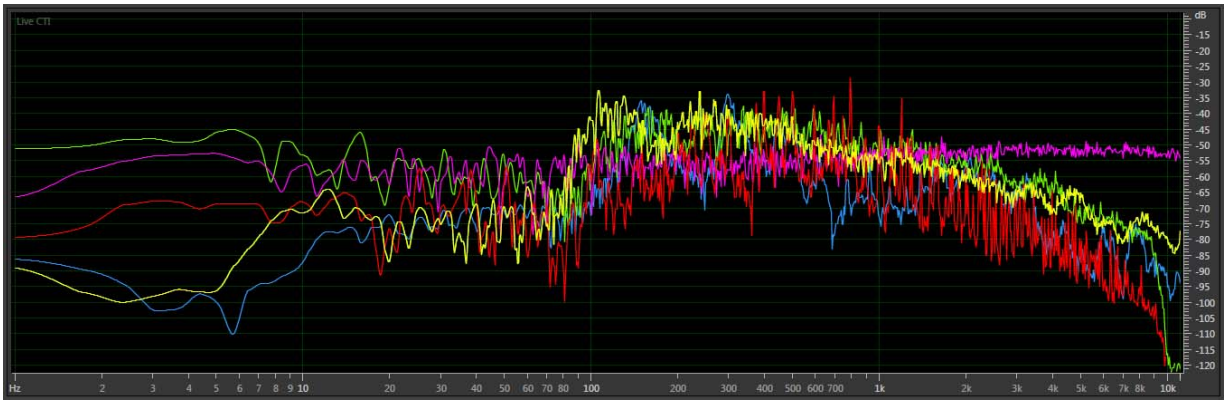


Figure 54: Long-term spectra of 6 seconds of babble noise (green), piano noise (red), white noise (purple), olnoise (yellow) and 2 seconds of speech (blue).

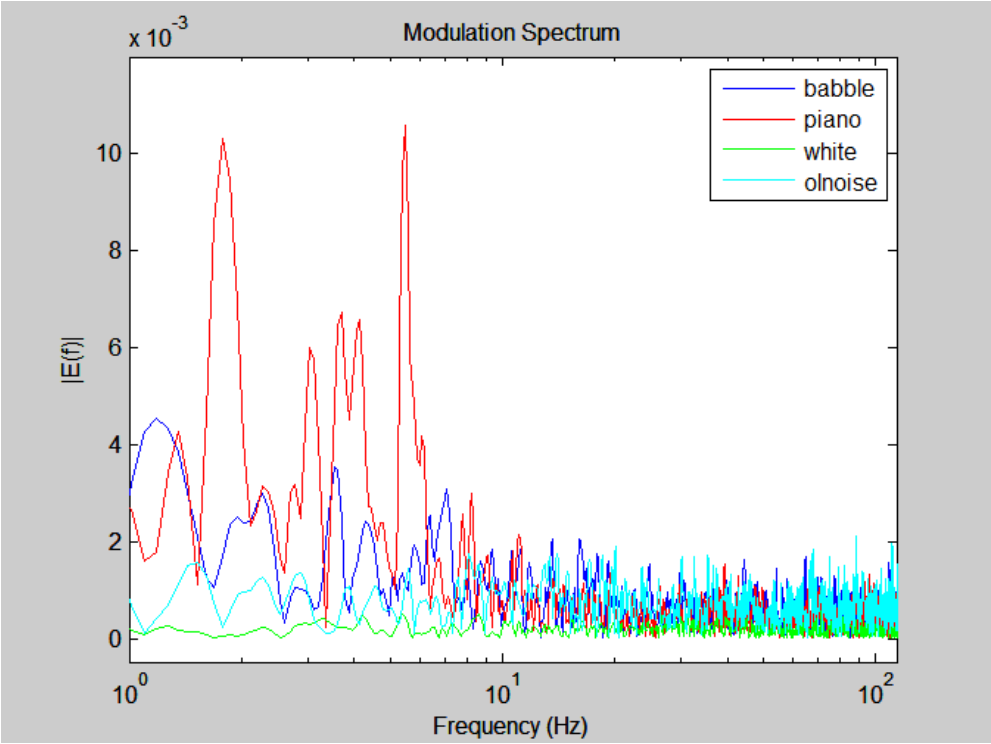


Figure 55: Modulation spectrum of 6 seconds of babble noise (blue), piano noise (red), white noise (green) and olnoise (cyan).

The SRT that was reported [15] for the olnoise without any enhancement, with tests to NH people, was -6.1 dB. The same measure was calculated for the 3 noise types used in the study, with 3 MACarena adaptive SRT tests to each of the 2 NH subjects. Again, the presentation order of the tests was randomized. The results are illustrated in Figure 56. It is obvious that the order of intelligibility from the easiest to the most difficult is: piano, white, Oldenburg and babble noise. Speech is more intelligible when it is mixed with noise that is dissimilar to and distinguishable from it.

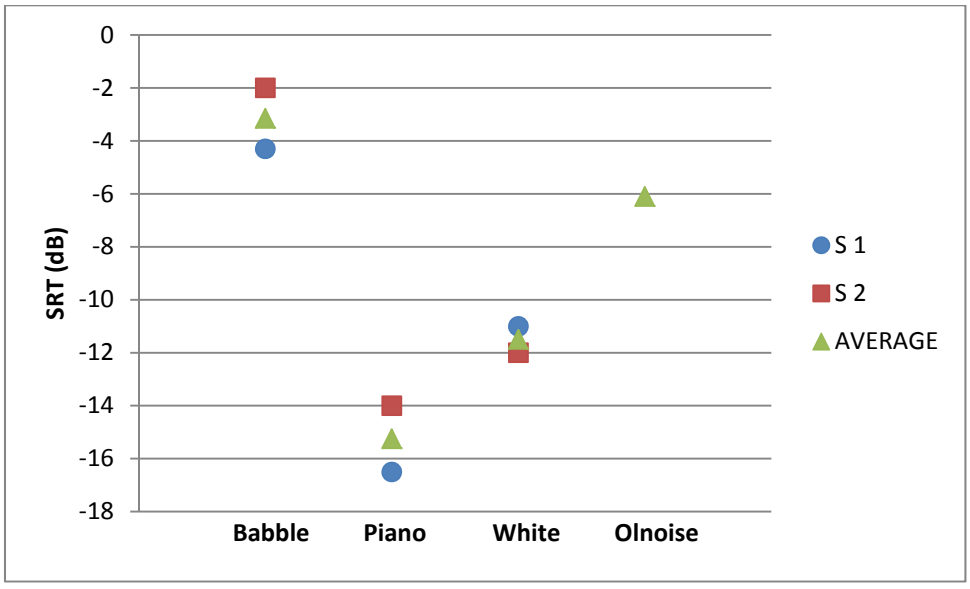


Figure 56: Results of adaptive Oldenburg SRT tests to NH people for babble, piano, white and Oldenburg noise.

C. Selection of Parameter Sets

The selection of the parameter sets 1 and 2 (enhancement conditions 1 and 2), was based on the test material. A list of 30 Oldenburg sentences was mixed with the 3 noise types under investigation (babble, piano and white noise) at 0 dB SNR. The 4 parameters (Residual Coherence Threshold, Beta, Geometric Index and FFT Size), varied around the default values of Table 2, within the ranges of Table 3 and the fwSegSNR gain after enhancement was averaged among the 30 files (Figures 57-68).

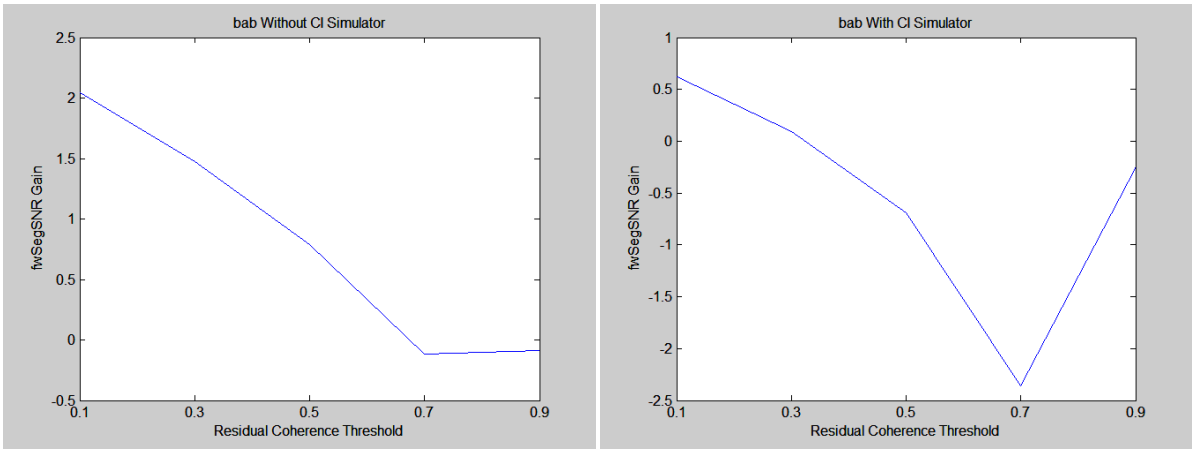


Figure 57: Optimization of Res. Coh. Thr. for babble noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

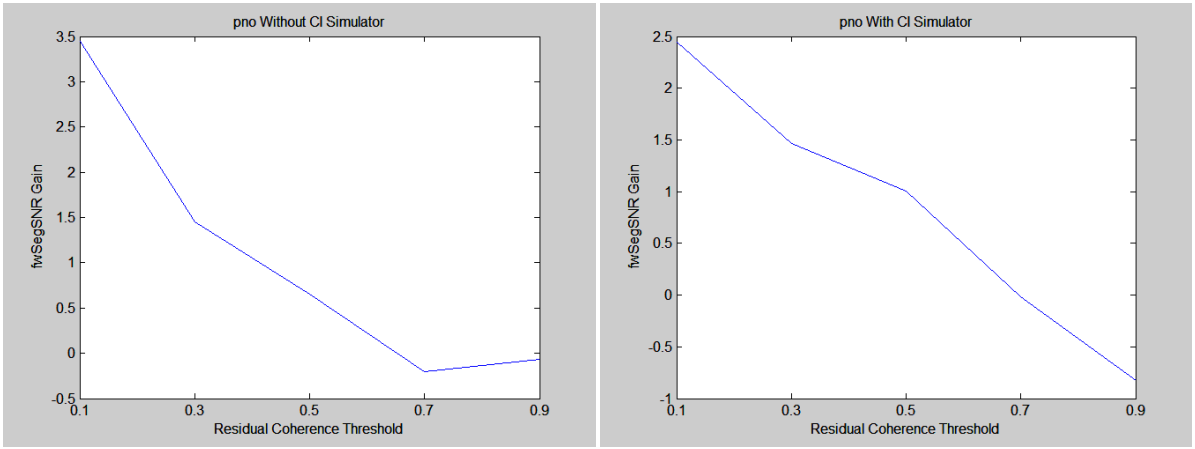


Figure 58: Optimization of Res. Coh. Thr. for piano noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

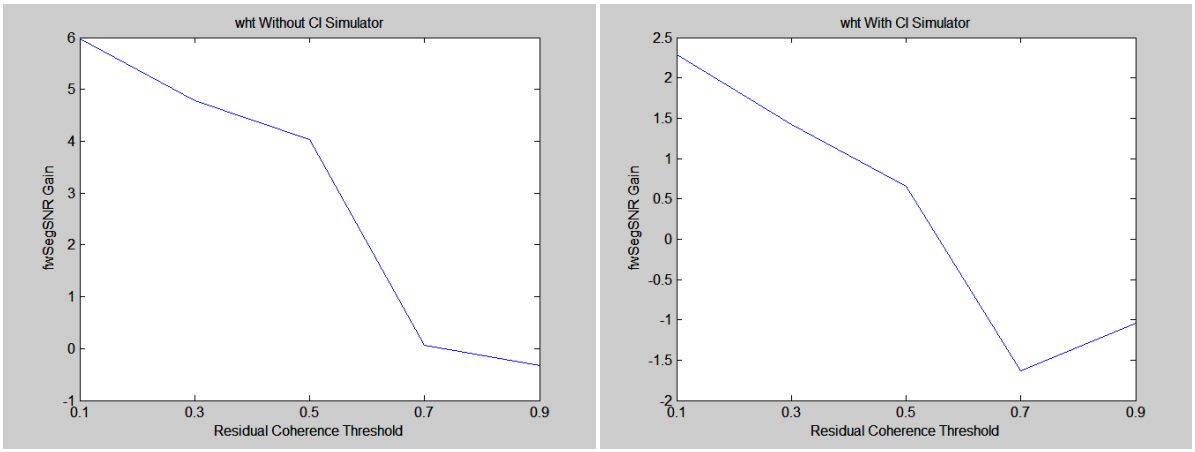


Figure 59: Optimization of Res. Coh. Thr. for white noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

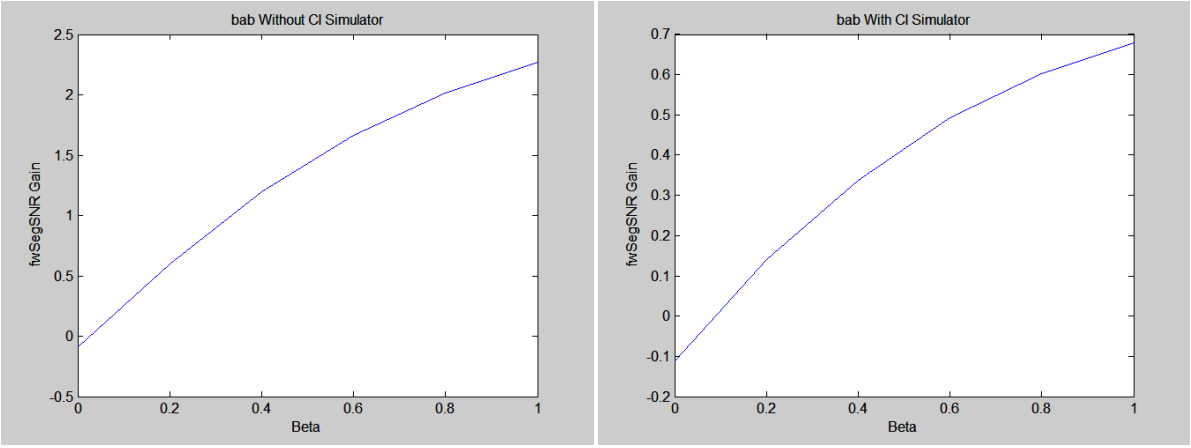


Figure 60: Optimization of Beta for babble noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

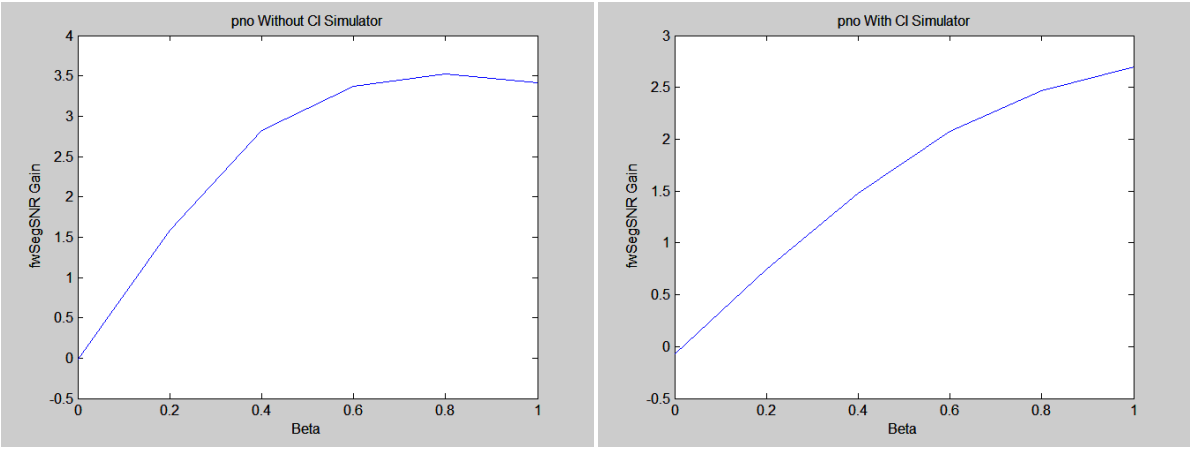


Figure 61: Optimization of Beta for piano noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

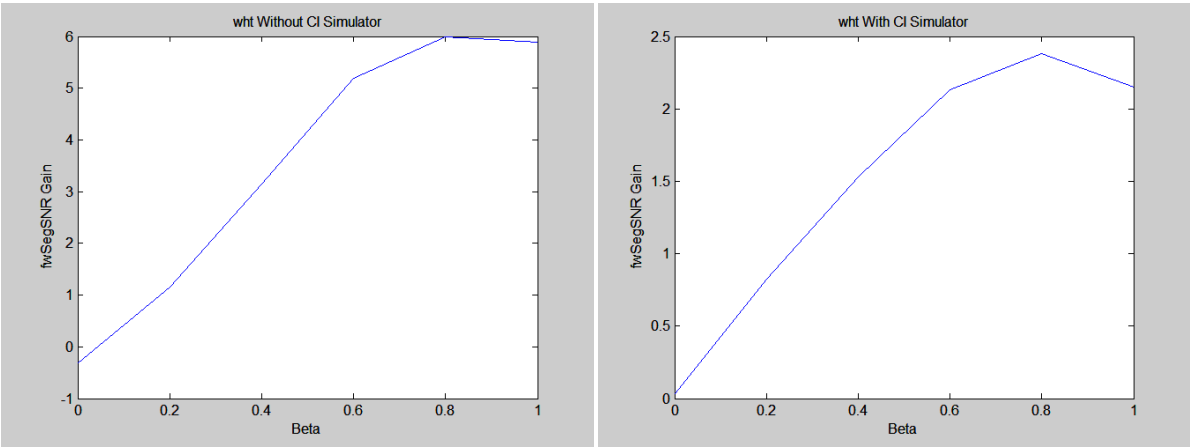


Figure 62: Optimization of Beta for white noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

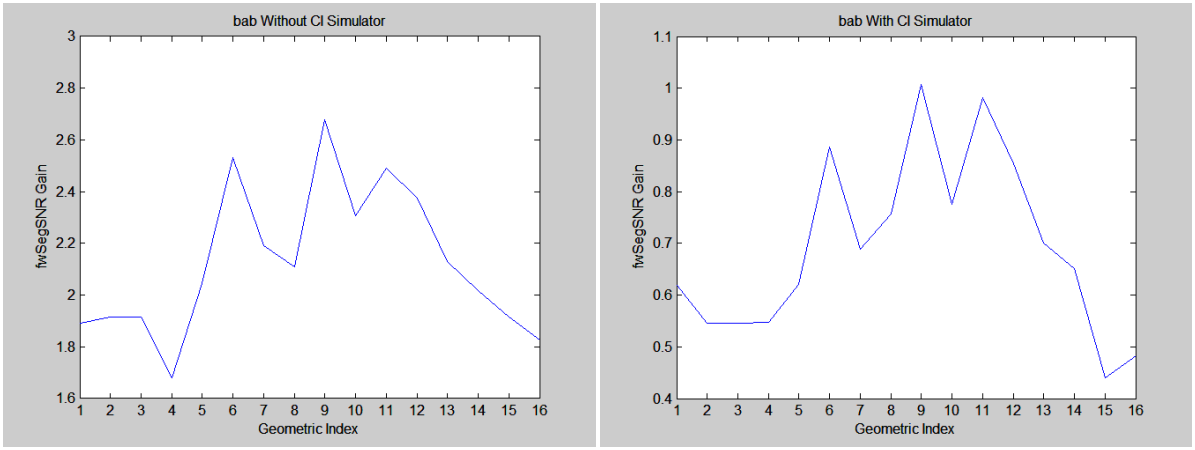


Figure 63: Optimization of Geom. Index for babble noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

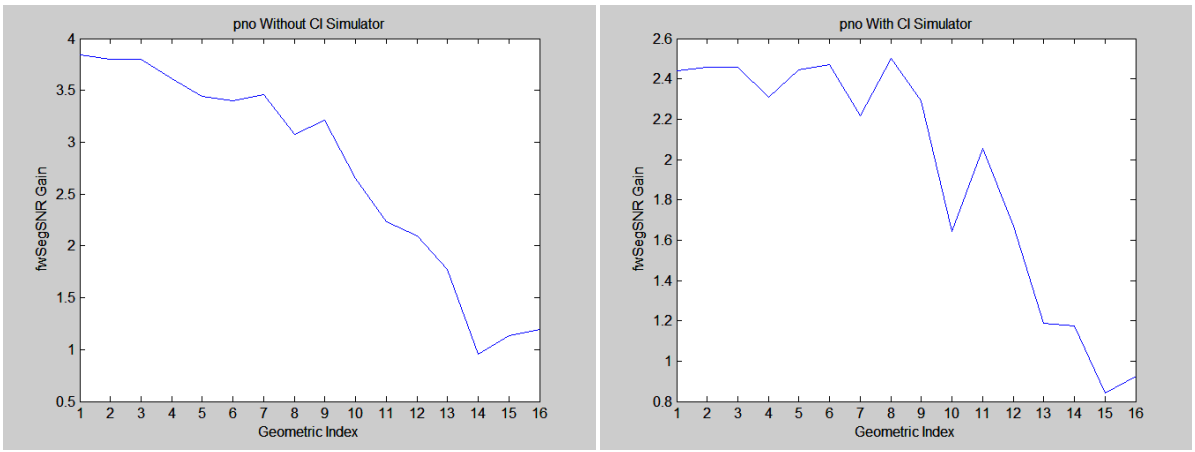


Figure 64: Optimization of Geom. Index for piano noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

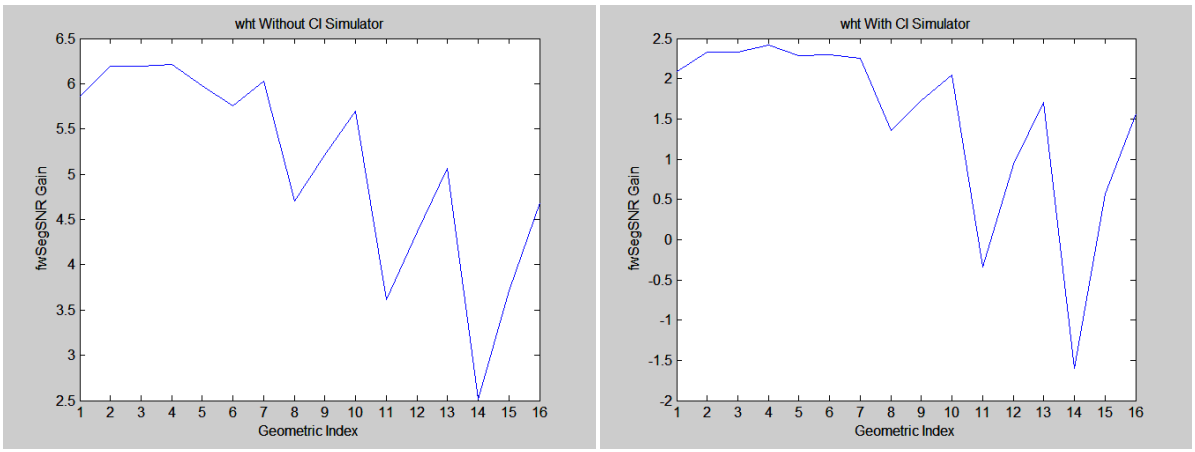


Figure 65: Optimization of Geom. Index for white noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

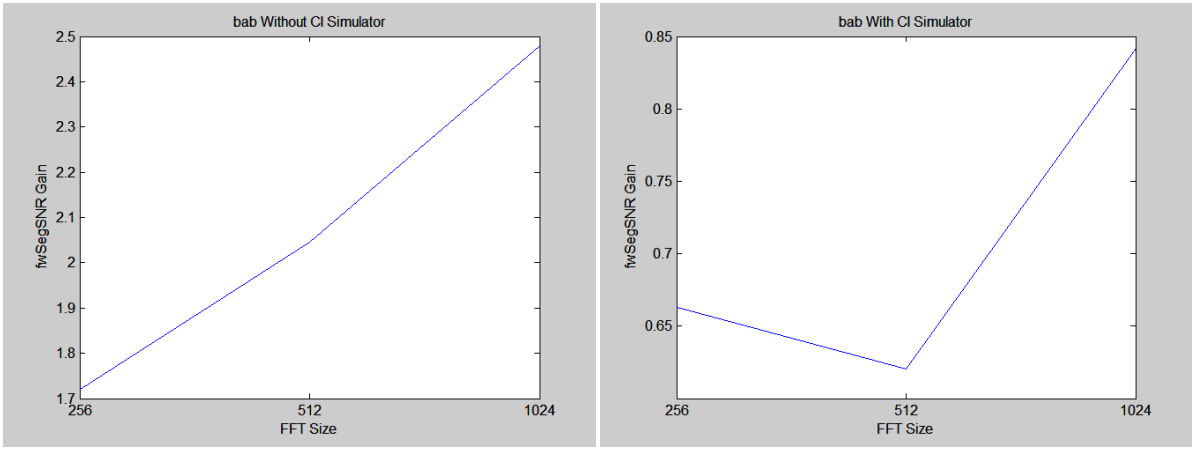


Figure 66: Optimization of FFT Size for babble noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

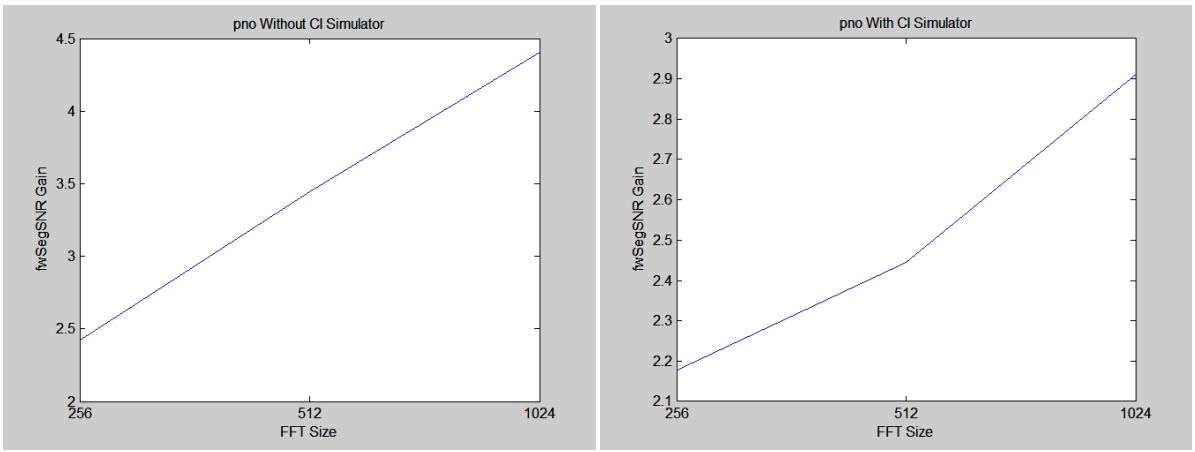


Figure 67: Optimization of FFT Size for piano noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

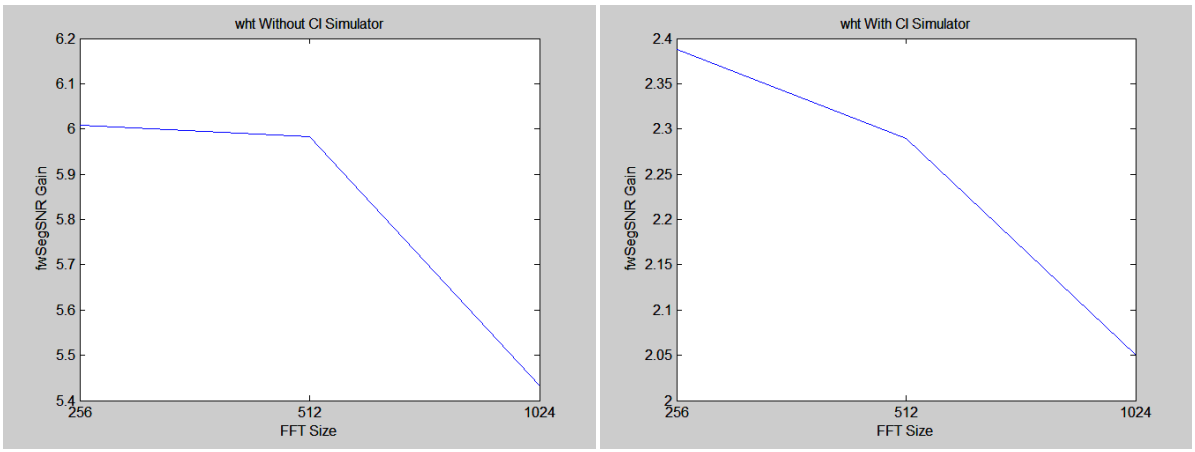


Figure 68: Optimization of FFT Size for white noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

The optimal parameters for sets 1 and 2, were selected based both on the objective measure and subjective listening. Both enhancement without and with the CI Simulator was taken into account and a compromise was made between both situations. The optimal parameters are listed in Table 10.

		SET 1	SET 2
BABBLE	Res. Coh. Thr.	0.1	0.1
	Beta	1	0.8
	Geom. Index	11	6
	FFT Size	1024	1024
PIANO	Res. Coh. Thr.	0.1	0.1
	Beta	0.8	0.6
	Geom. Index	1	6
	FFT Size	1024	1024
WHITE	Res. Coh. Thr.	0.1	0.1
	Beta	1	0.8
	Geom. Index	5	8
	FFT Size	256	1024

Table 10: Parameterization sets 1&2 for babble, piano and white noise.

A comparison between the 3 enhancement conditions, in terms of the objective measure, is presented in Figures 69-71 for the 3 noise types under investigation. The fwSegSNR gain was averaged among 30 degraded Oldenburg sentences after 3 discrete enhancement conditions (SpEnh1 with set 1, SpEnh2 with set 2 and Unprocessed with no enhancement). The validation list of 30 files was different from the list of 30 files that was used during optimization.

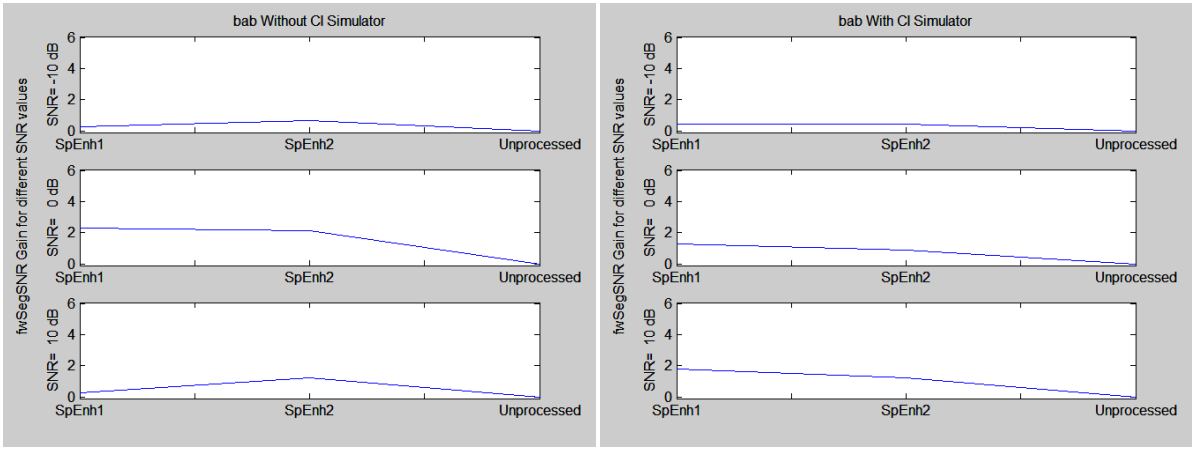


Figure 69: Comparison between the 3 enhancement conditions for babble noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

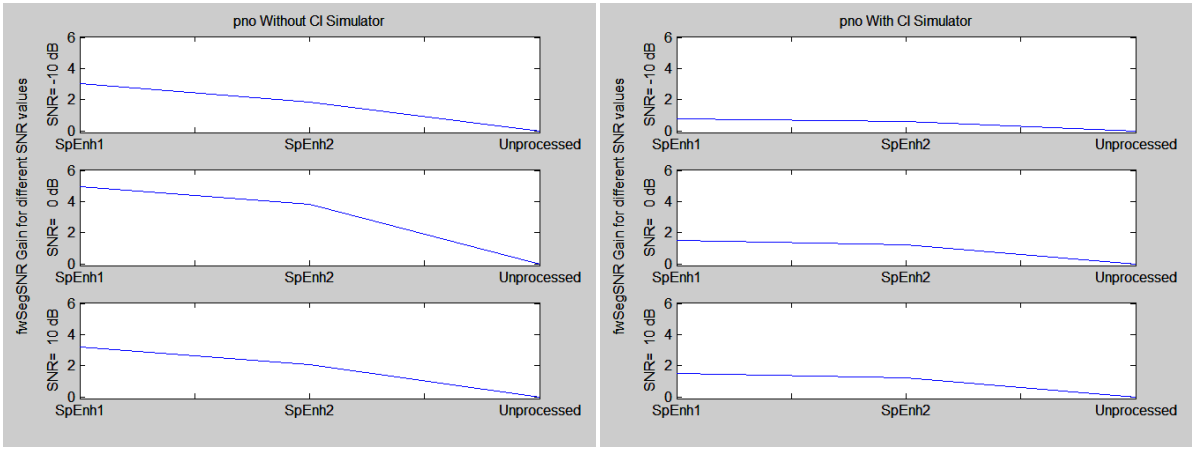


Figure 70: Comparison between the 3 enhancement conditions for piano noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

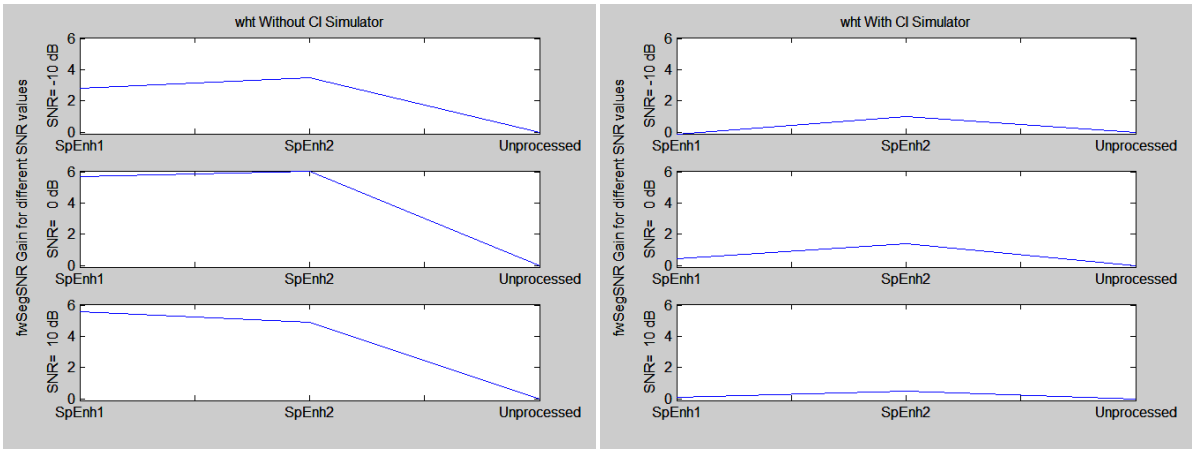


Figure 71: Comparison between the 3 enhancement conditions for white noise. (Left): without the CI Simulator. (Right): with the CI Simulator.

It was aimed to select the parameters, such that set 1 offers better enhancement at the cost of artifact generation and set 2 has smooth sound quality while being more noisy. As it can be seen from the figures above, better enhancement was, in general, achieved with set 1 for babble and piano noise. Regarding white noise, set 2 is more preferable to set 1 according to the objective measure. However, by subjective listening, the files enhanced with set 1 have more sharp speech, while the files enhanced with set 2 sound more natural and at the same time more noisy. Furthermore, it can be observed that also for this speech material, the algorithm performs better for piano and white noise than for babble noise. The fwSegSNR gain drops when the CI Simulator is included.

In total, 102060 files were generated for the study (3 noise types, 3 enhancement conditions, without and with the CI Simulator, 9 lists, 21 SNRs and 30 files for each). The degraded speech files of the “Unprocessed” enhancement condition (3), were downsampled to 16kHz and upsampled back, as the algorithm resamples its inputs to 16kHz.

D. Results from Tests with NH Subjects using the CI Simulator

The following figures present the measured value of SRT, for 6 NH subjects , for 3 different types of noise (babble, piano and white) and for 3 conditions (2 enhancement conditions and 1 without any enhancement) using the CI Simulator.

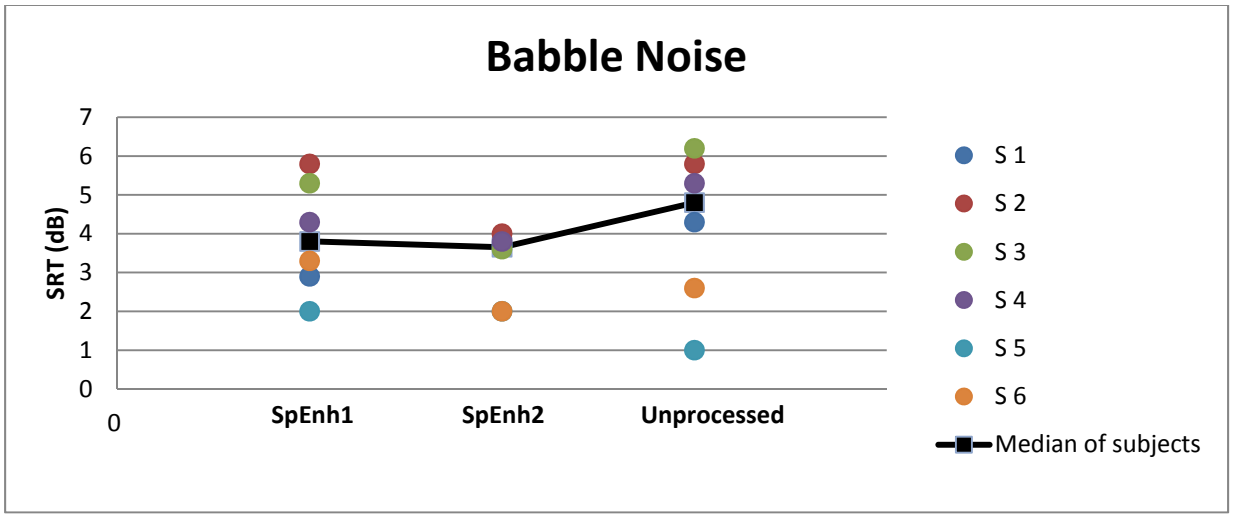


Figure 72: SRT among 6 NH subjects with the CI Simulator for 3 enhancement conditions. Babble noise.

In Figure 72, it can be seen that the median SRT which corresponds to the enhancement conditions (SpEnh1 and SpEnh2) is lower than the median SRT of the Unprocessed condition. A lower value represents a lower SNR required for 50% intelligibility, indicating thus a performance improvement when enhancement is applied. In an effort to compare the 2 enhancement conditions, it could be said that they exhibit similar performance. However, the data for SpEnh2 are more concentrated around the median with the exception of one point. Moreover, there is a big variance (1dB to 6.2dB) between the data of the Unprocessed condition. Finally the medians of SpEnh1, SpEnh2 and Unprocessed are 3.8 dB, 3.65 dB and 4.8 dB, respectively.

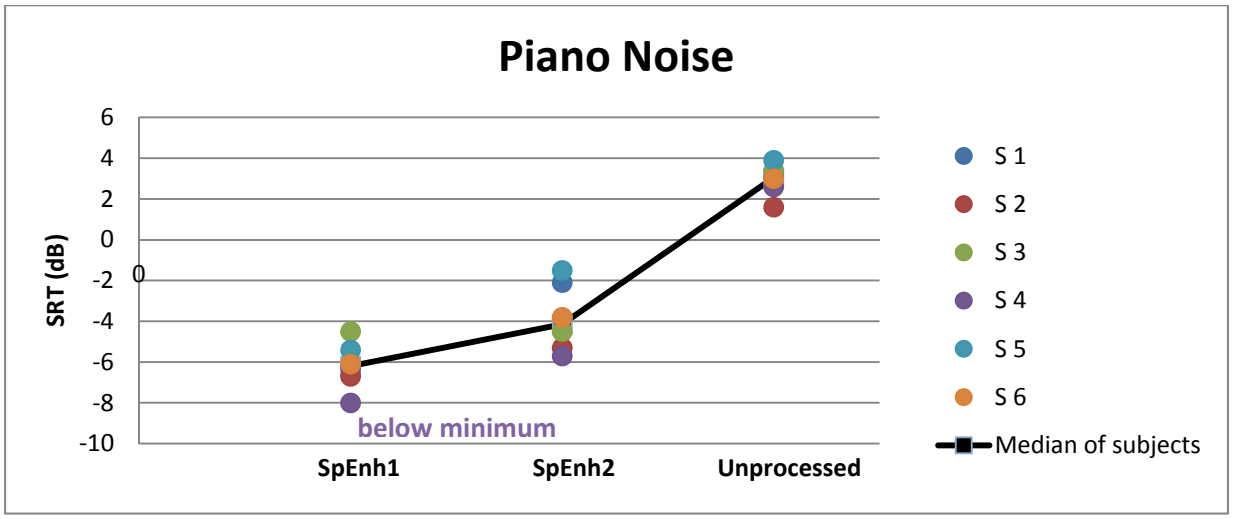


Figure 73: SRT among 6 NH subjects with the CI Simulator for 3 enhancement conditions. Piano noise.

In the case of piano noise (Figure 73), the data among different subjects are more concentrated around their medians than for babble noise. Furthermore, it is clear that there is a performance improvement when enhancement is applied, which becomes more prominent for SpEnh1. The medians of SpEnh1, SpEnh2 and Unprocessed are -6.2 dB, -4.15 dB and 3.05 dB, respectively. The performance of the subjects with SpEnh1, was so good that in one case (subject 4-purple) it exceeded the prepared SNR range of the test. More specifically, a file with lower SNR than the minimum provided was required in order to continue the adaptive test procedure.

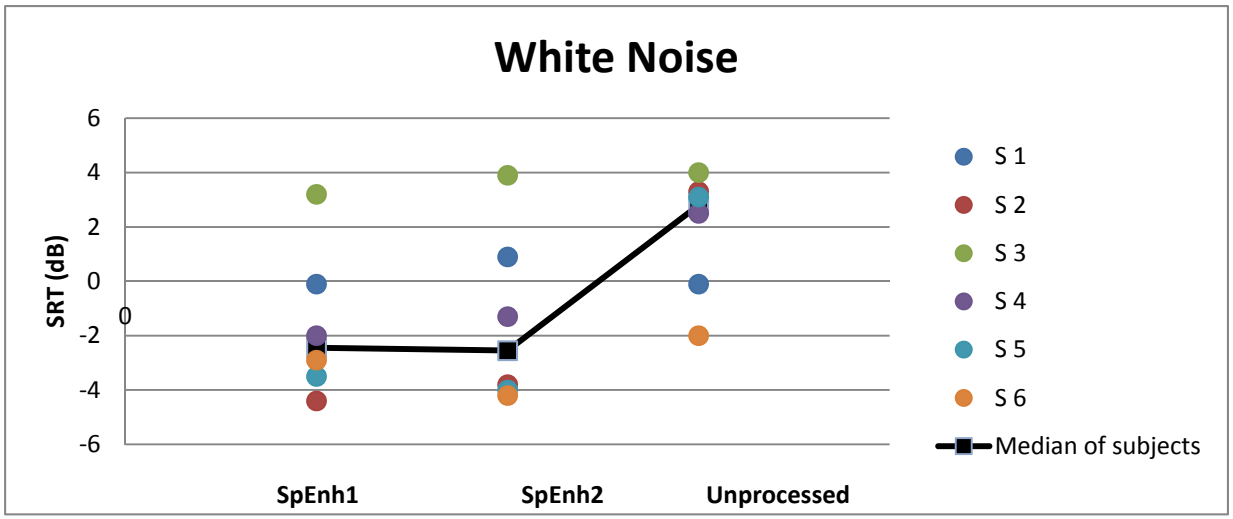


Figure 74: SRT among 6 NH subjects with the CI Simulator for 3 enhancement conditions. White noise.

For white noise (Figure 74), the variance of the data around their medians is large in all 3 conditions. In SpEnh2 it reaches 8.1dB. However, by comparing the medians, it is clear that enhancement leads to a better performance. The medians of SpEnh1, SpEnh2 and Unprocessed are -2.45 dB, -2.55 dB and 2.8 dB, respectively. Furthermore, the 2 enhancement conditions, SpEnh1 and SpEnh2, are comparable with each other.

In conclusion, regardless of the noise type, the median SRT of the Unprocessed condition is always larger than the median of the 2 Enhancement conditions, indicating a performance improvement when the SE algorithm is used.

In order to get a feeling of the effectiveness of the SE algorithm in improving the intelligibility of speech in noise, it is not sufficient to be aware of the absolute SRT values that were presented above. It is preferable to measure the relative SRT value of the first 2 enhancement conditions in comparison with the SRT value of the Unprocessed condition. For this reason, the SRT values for SpEnh1 and SpEnh2 were subtracted from the SRT value of the Unprocessed condition. The results are presented in the Figures 75-77.

In Figure 75 for babble noise, it can be seen that the median relative SRT for SpEnh2 is 0.45 dB, while for SpEhn2 it is 1.05 dB. These values are not very large, but they are positive thus indicating improvement. Moreover, the median performance improvement for SpEnh2 is larger than the one of SpEnh1 by 0.6 dB. Besides this, in SpEnh2 the relative SRT value of individually 5 out of 6 subjects is positive.

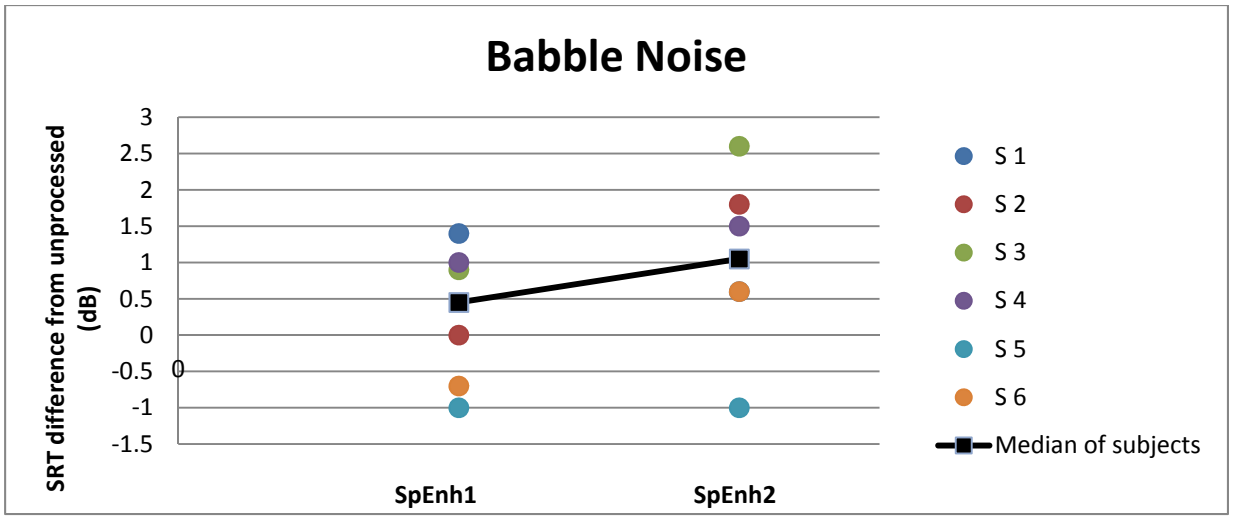


Figure 75: SRT improvement among 6 NH subjects with the CI Simulator for 2 enhancement conditions. Babble noise.

In the case of piano noise (Figure 76), there is a significant median SRT improvement. For SpEnh1 it is 9.2 dB and for SpEnh2 6.85 dB. The exceptionally good performance of subject 4 in SpEnh1, is indicated as “above maximum” improvement. In all individual subjects there is a positive relative SRT value.

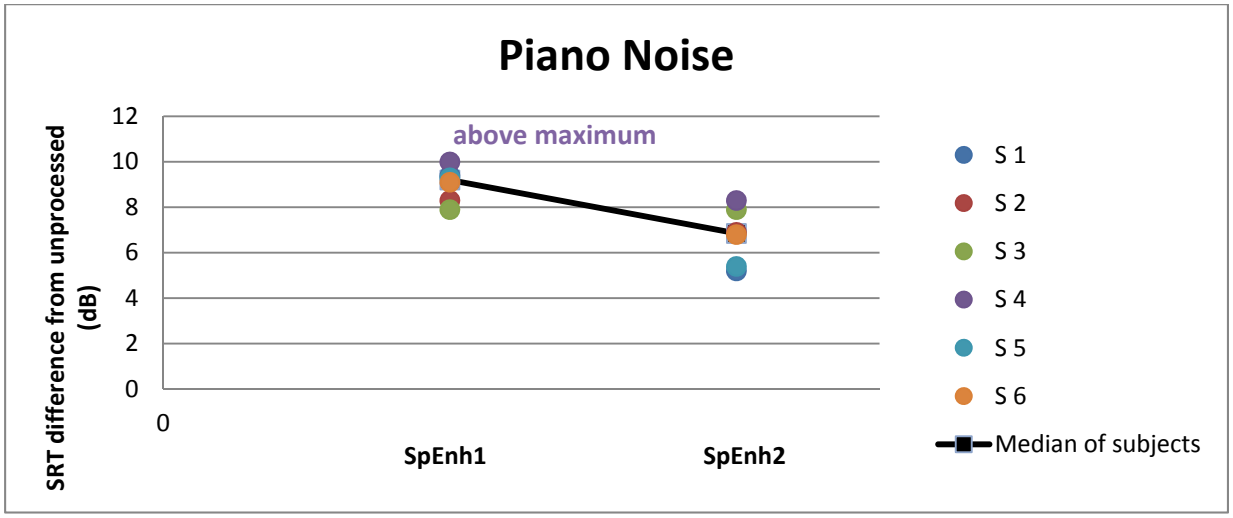


Figure 76: SRT improvement among 6 NH subjects with the CI Simulator for 2 enhancement conditions. Piano noise.

For white noise (Figure 77), there is a median relative SRT of 2.7 dB and 3 dB for SpEnh1 and SpEnh2, respectively, showing a small preference of the subjects to SpEnh2. Furthermore, in all individual subjects except for one, there is a positive relative SRT indicating intelligibility improvement with the use of the SE algorithm.

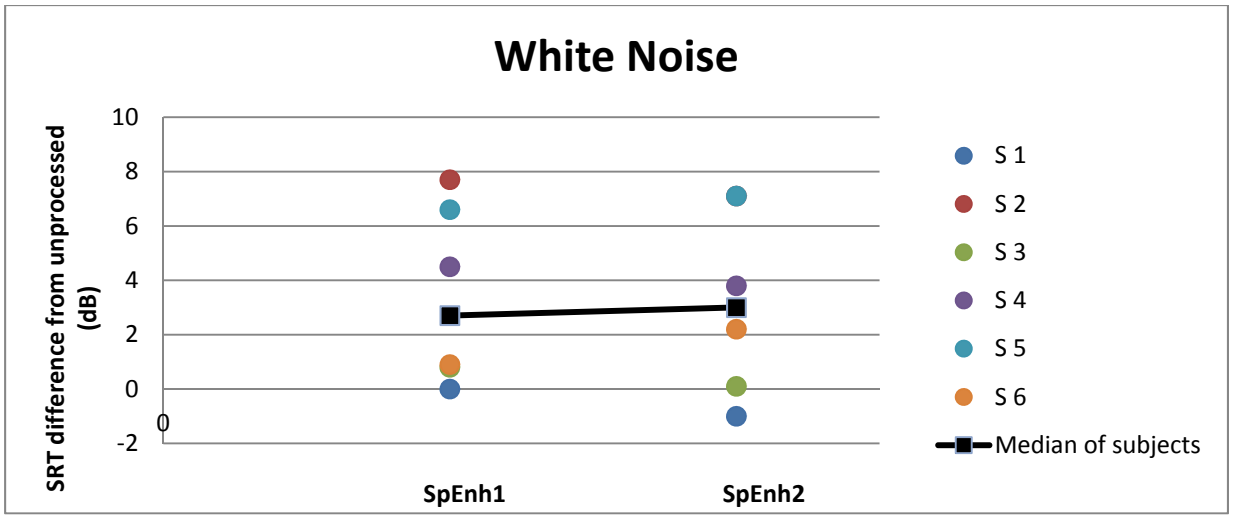


Figure 77: SRT improvement among 6 NH subjects with the CI Simulator for 2 enhancement conditions. White noise.

At this point, it would be interesting to examine the absolute performance and the performance improvement in relation to the noise type. For this reason, the following figures illustrate the median values of SRT and relative SRT with respect to the 3 different noise types.

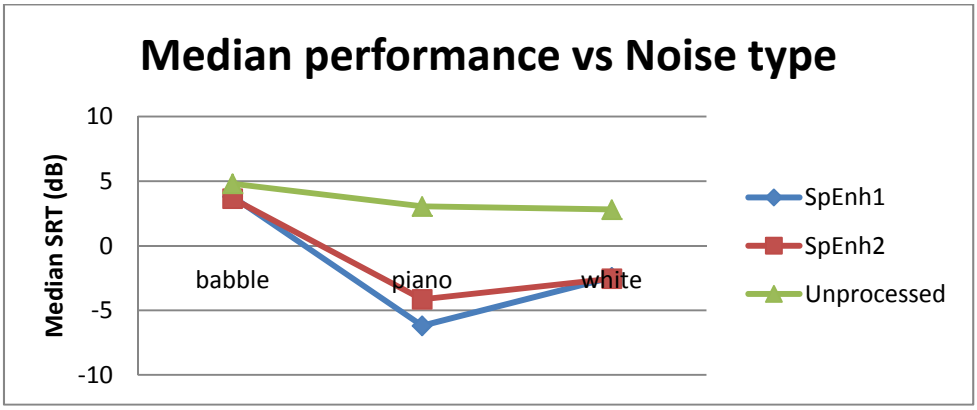


Figure 78: Median SRT with respect to noise type.

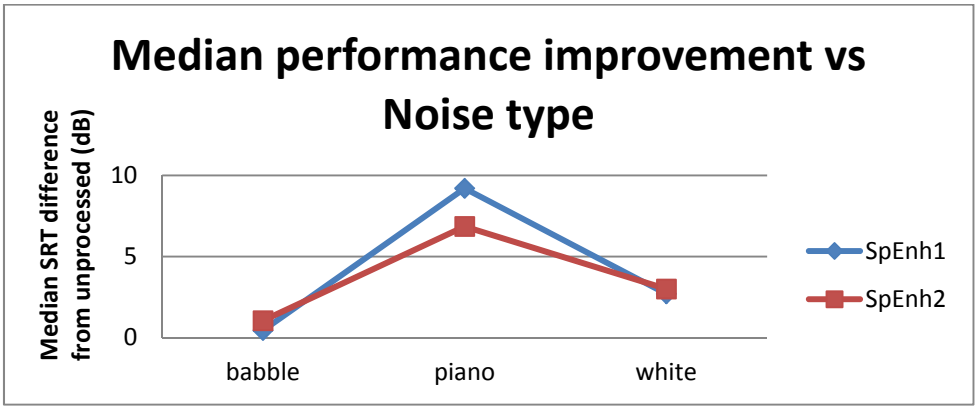


Figure 79: Median SRT improvement with respect to noise type.

In the Unprocessed condition, the median performance doesn't vary a lot among different noise types. The largest difference observed is 2dB between babble and white noise. However, when enhancement is introduced, the variance of the performance among different noise types is highlighted. In both enhancement conditions, the absolute as well as the relative performance is much more significant for piano noise than for babble noise and medium for white noise.

A factor which affects the performance and is interesting to be investigated is the training effect of the subjects. Therefore, the SRT value measured is presented in the following figures in relation to the order of appearance of one test within the session of the 9 tests. Three different figures (80-82) exist, each one corresponding to a separate noise type. In addition, every figure contains 3 curves corresponding to the 3 different conditions. Therefore, there are in total 9 training curves, one for each of the 9 tests that comprise a session. When a specific test appears at the same order for more than one subject, the values are averaged.

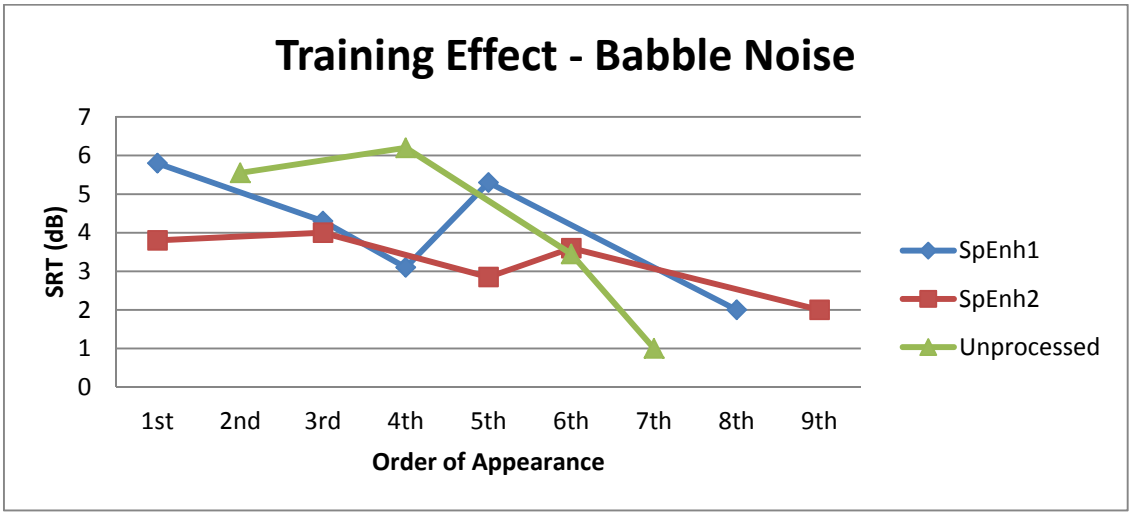


Figure 80: SRT of individual tests with respect to order of appearance. Babble noise.

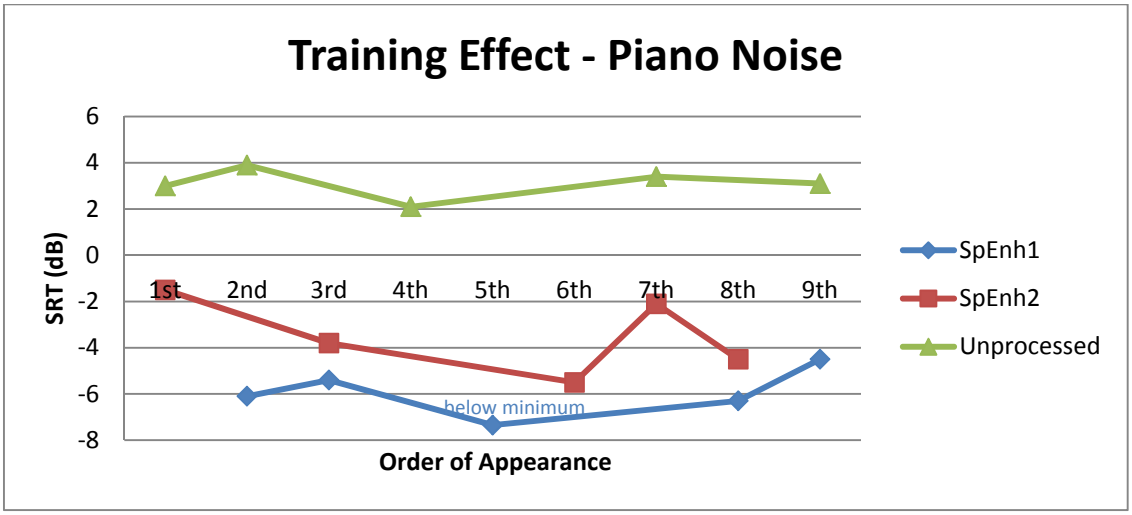


Figure 81: SRT of individual tests with respect to order of appearance. Piano noise.

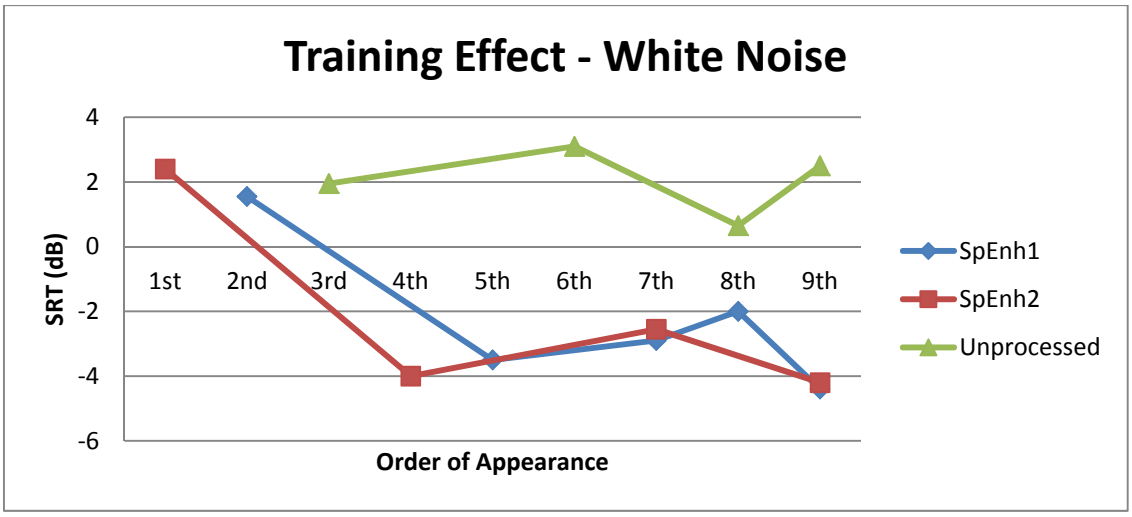


Figure 82: SRT of individual tests with respect to order of appearance. White noise.

In general, most of the curves exhibit a declining tendency, which proves that the training effect exists. Regarding the Unprocessed condition, the training effect is only evident in the case of babble noise. As far as the Enhancement conditions are concerned, there is almost no training effect in piano noise, a medium one in babble noise and a very big one in white noise. There, the maximum SRT difference that is observed between the 1st and the 9th order is 6.6 dB for SpEnh2. The exceptionally good performance of subject 4 in piano SpEnh1 is indicated by "below minimum".

To simply illustrate the preference of the subjects to the 3 conditions in relation to their intelligibility, it was counted how many times (i.e. for how many subjects) a certain condition was the preferred one of the 3, for a certain type of noise. When a subject showed the same performance for 2 different conditions, the score was distributed as half and half. Moreover, the exceptionally good performance of subject 4 was of course counted in favour of the condition for which it appeared. The following histogram (Figure 83) illustrates the above.

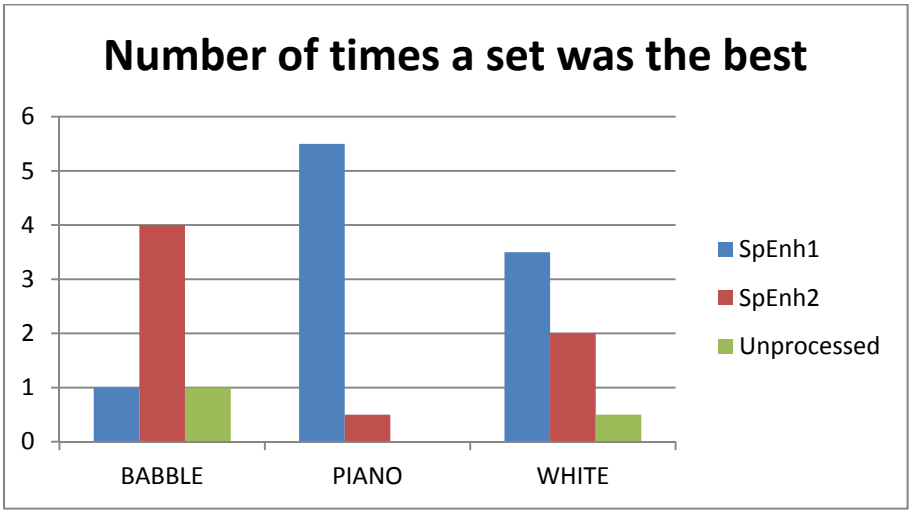


Figure 83: Set preference among 6 subjects for 3 noise types.

In the case of babble noise, it is clear that SpEnh2 is the preferred one. Regarding piano noise, SpEnh1 outperforms the rest of the conditions. Besides this, the Unprocessed condition was never

preferred among the subjects. Finally, for white noise, the two Enhancement conditions exhibit a comparable performance, with SpEnh1 overcoming SpEnh2. At his point, it is worth to remind that SpEnh1 is in general a condition with sharper parameterization which leads to a better enhancement but also to the generation of more artefacts. On the other hand, SpEnh2 involves softer parameterization as well as the generation of less prominent artefacts. The aforementioned provides information regarding the preference of a sharper or a softer parameterization of the SE algorithm in relation to the noise type.

Finally, it would be interesting to observe the deterioration in SRT when the CI Simulator is introduced. For this reason, the median SRT without the CI Simulator for the Unprocessed condition (Figure 56, Paragraph IV.B), is compared with the median SRT corresponding to the Unprocessed condition, that was reported in this paragraph. The comparison for the 3 noise types, is presented in Table 11. The SRT is increased by 7.95 dB for babble noise, by 18.3 dB for piano noise and by 14.3 dB for white noise, with the introduction of the CI Simulator. For example, for babble noise, in order for a degraded speech file to be intelligible by 50% after the CI Simulator, it needs have an initial SNR of 7.95 dB higher than when the intelligibility is measured without the CI Simulator. Bigger deterioration on performance is observed for piano and white noise, as objectively reported also in Chapter III.

	Without CI Simulator	With CI Simulator	Deterioration
BABBLE	-3.15 dB	4.8 dB	7.95 dB
PIANO	-15.25 dB	3.05 dB	18.3 dB
WHITE	-11.5 dB	2.8 dB	14.3 dB

Table 11: SRT deterioration in the "Unprocessed" condition, with the introduction of the CI Simulator, for 3 noise types.

E. Results from Tests with CI Patients

The SRT that was measured for 5 CI patients , for 3 different types of noise (babble, piano and white) and for 3 conditions (2 enhancement conditions and 1 without any enhancement) is illustrated in the following figures.

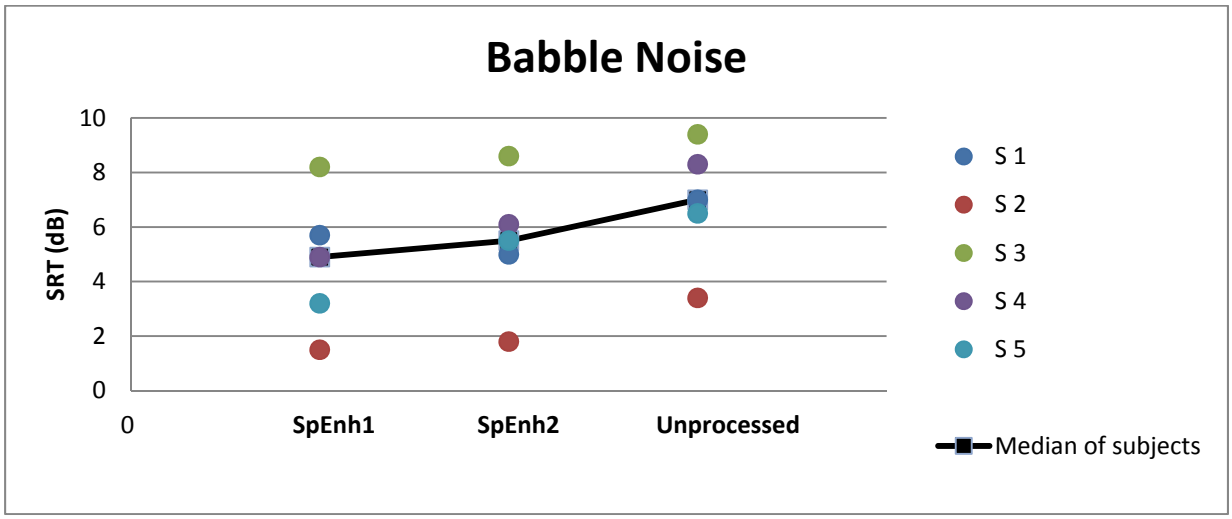


Figure 84: SRT among 5 CI patients for 3 enhancement conditions. Babble noise.

In Figure 84 for babble noise, it is immediately noticeable that there is a big variability of results among the 5 CI patients for all three enhancement conditions. The range of measured SRTs is 6-7 dB, exceeding the one of NH people with the Simulator by 2 dB. Moreover, in the case of CI patients, it is clear that SpEnh1 is slightly preferred, as it leads to a lower SRT, contrary to what observed for NH people with the Simulator. Finally, the medians of SRT are 4.9 dB, 5.5 dB and 7 dB for SpEnh1, SpEnh2 and Unprocessed, respectively. In general, the CI patients exhibit an SRT of 2 dB higher than the NH people with the Simulator. Therefore, their performance is worse than predicted. However, the simulated impairment in intelligibility can be increased by lowering the spectrum slope of the added noise.

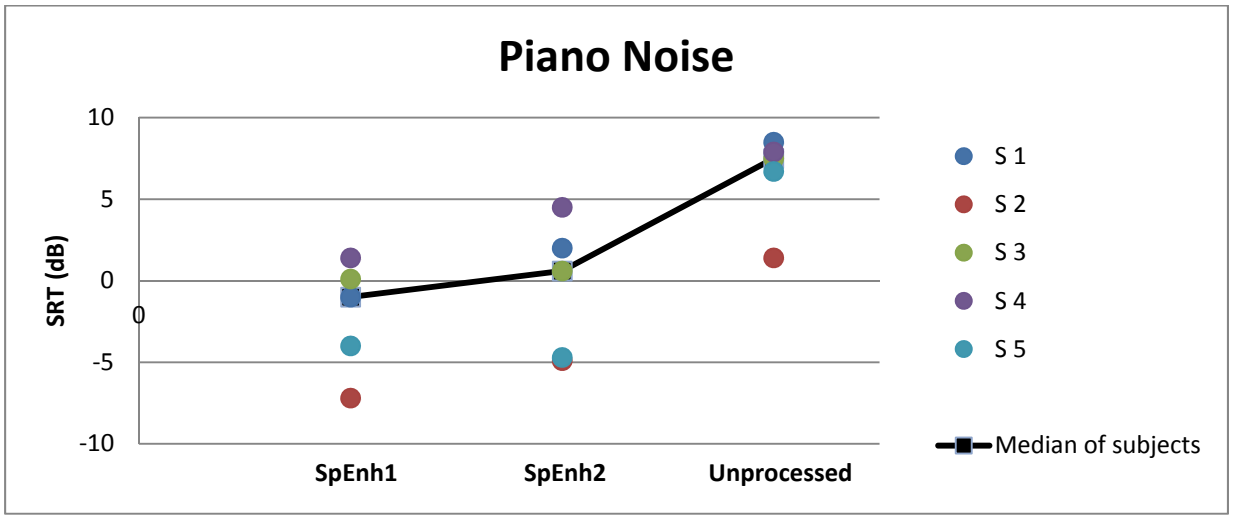


Figure 85: SRT among 5 CI patients for 3 enhancement conditions. Piano noise.

In Figure 85 for piano noise, a variability of approximately 8 dB range is observed among the 5 subjects. This is bigger than the variability of NH people with the Simulator, by 4 dB. However, the preference to SpEnh1 is consistent between the 2 cases. The median SRTs measured for SpEnh1, SpEnh2 and Unprocessed are -1 dB, 0.6 dB and 7.5 dB, respectively. Again, the performance of CI patients is worse than the simulated one, by 4 dB.

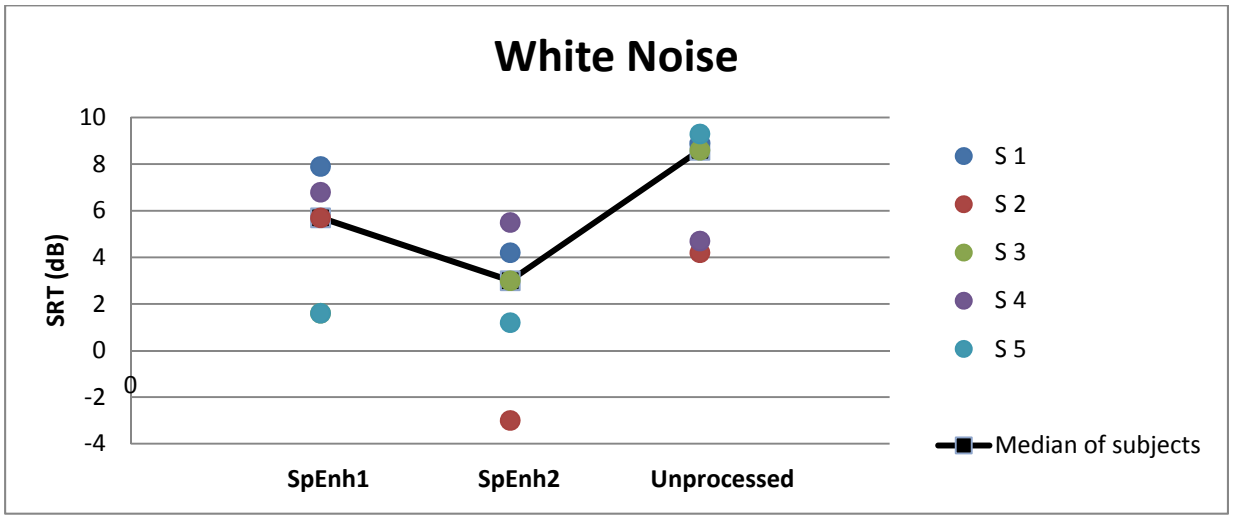


Figure 86: SRT among 5 CI patients for 3 enhancement conditions. White noise.

Regarding white noise, Figure 86, there is a variability of 6-7 dB, which is slightly smaller than the one for NH people with the CI Simulator. Unlike NH people, for which SpEnh1 and SpEnh2 exhibit similar performance, for CI patients, SpEnh2 is clearly preferable. The smooth outcome produced by parameterization set 2, leads to better intelligibility in CI patients. The median SRTs are 5.7 dB, 3 dB and 8.6 dB for the three enhancement conditions. This values are approximately by 6 dB larger here than for NH people with the Simulator.

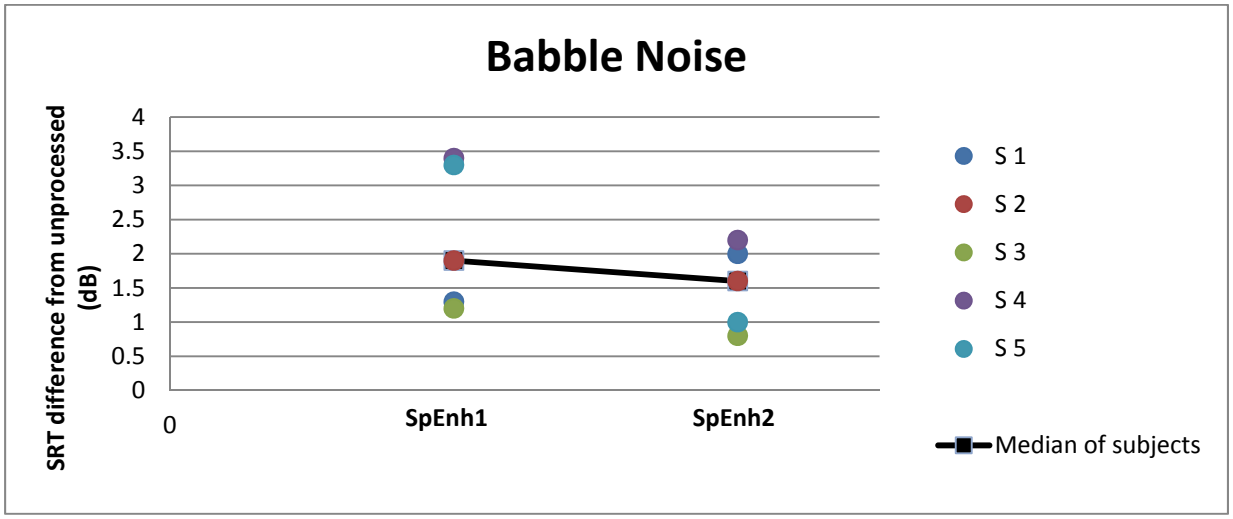


Figure 87: SRT improvement among 5 CI patients for 2 enhancement conditions. Babble noise.

The SRT improvement when the algorithm is used (SpEnh1&2) in relation to the Unprocessed condition is examined for babble noise through Figure 87. It is noticeable that all the individual values are positive, showing an improvement in intelligibility with the application of the algorithm. The median SRT improvement is 1.9 for SpEnh1 and 1.6 for SpEnh2. This indicates a slight preference for SpEnh1, contrary to what observed for NH people with the Simulator. In addition, there is a smaller variability in the relative than in the absolute results, as the differences in the general level of performance among the CI patients are eliminated.

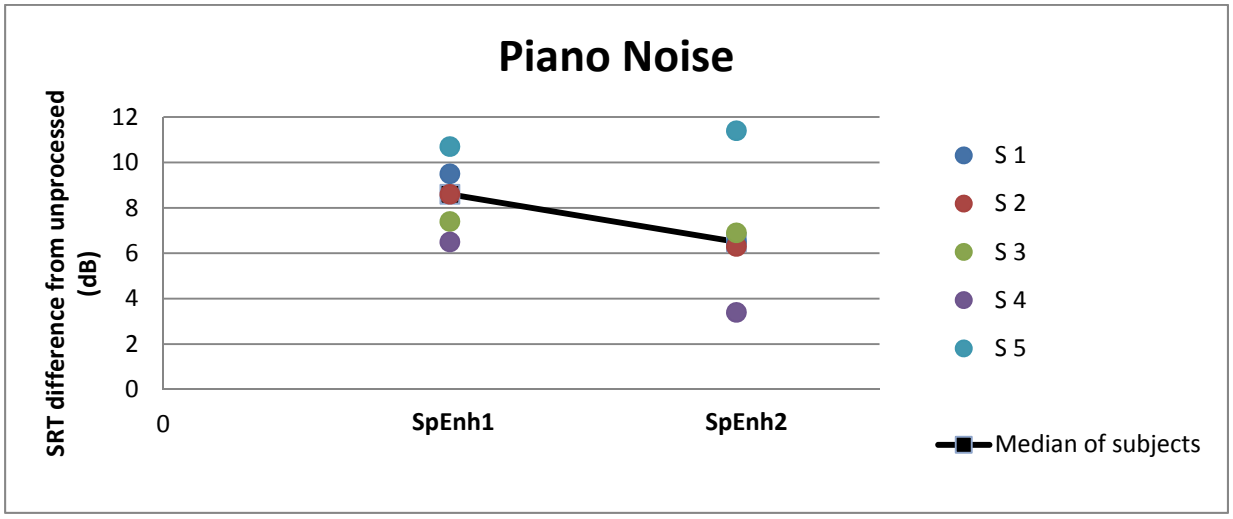


Figure 88: SRT improvement among 5 CI patients for 2 enhancement conditions. Piano noise.

An exceptional SRT improvement is exhibited for piano noise (Figure 88), especially for SpEnh1. The median improvement achieved is 8.6 and 6.5 for SpEnh1 and SpEnh2, respectively. The relative SRT, shown here, has a similar value to the case of NH people with the Simulator, contrary to the absolute SRT. This is because the same degree of impairment is imposed by the Simulator to all enhancement conditions. Therefore, the correction that needs to be made to this degree, in order for the CI Simulator to match the actual impairment that takes place in a CI, is not necessary when the relative SRT is measured instead of the absolute one.

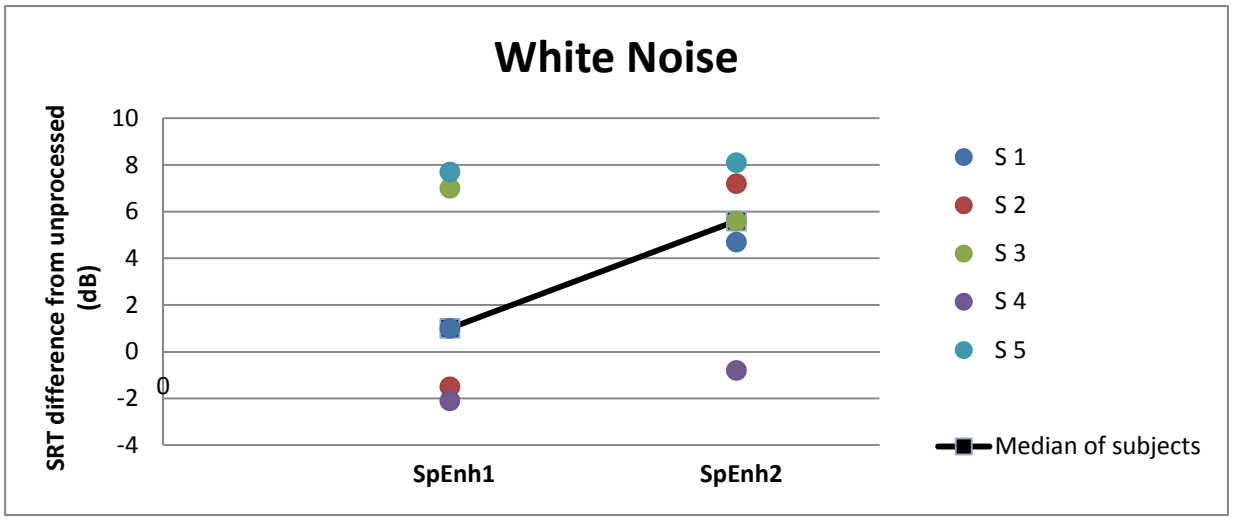


Figure 89: SRT improvement among 5 CI patients for 2 enhancement conditions. White noise.

Although for NH people with the CI Simulator the SRT improvement is comparable between SpEnh1 and SpEnh2 in the case of white noise, for CI patients, SpEnh2 is clearly preferred. The median SRT improvement measured, is (Figure 89) 1 dB for SpEnh1 and 5.6 dB for SpEnh2. Despite the fact that relative values are illustrated, the variability of results lies within the wide range of 9-10 dB.

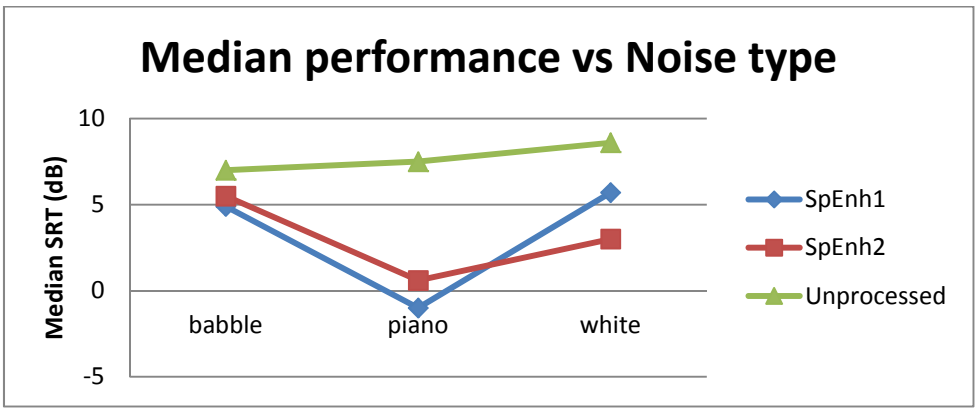


Figure 90: Median SRT with respect to noise type.

Figures 90 and 91, illustrate the median absolute and relative SRT, respectively, in relation to the noise type. As observed also in the case of NH people with the CI Simulator, for the Unprocessed condition, similar SRT is shown among the three noise types. The maximum difference is approximately 3 dB, between babble and white noise. However, when the algorithm is used, the performance is much better for piano noise than for the other two noise types for both enhancement conditions. For SpEnh2, for white noise the intelligibility is better than for babble

noise, similarly to NH people with the Simulator for both enhancement conditions. However, SpEnh1 leads to an intelligibility of white noise comparable to the one of babble noise.

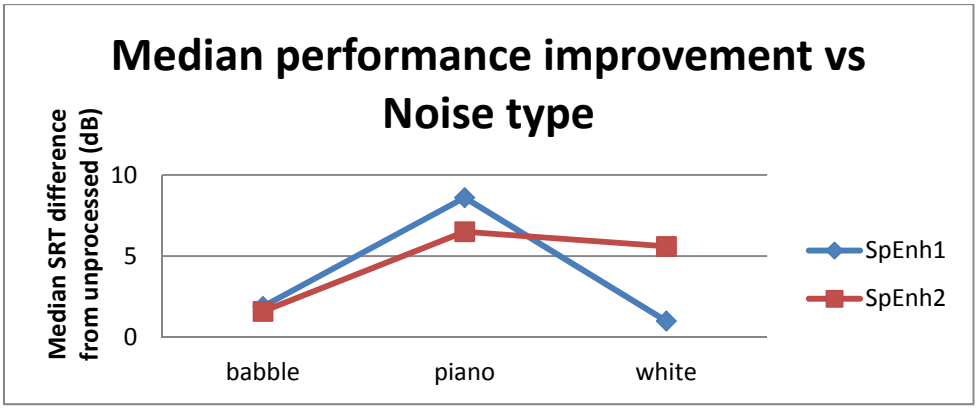


Figure 91: Median SRT improvement with respect to noise type.

Figures 92-94 depict the SRT of individuals tests in relation to the order of appearance in a session.

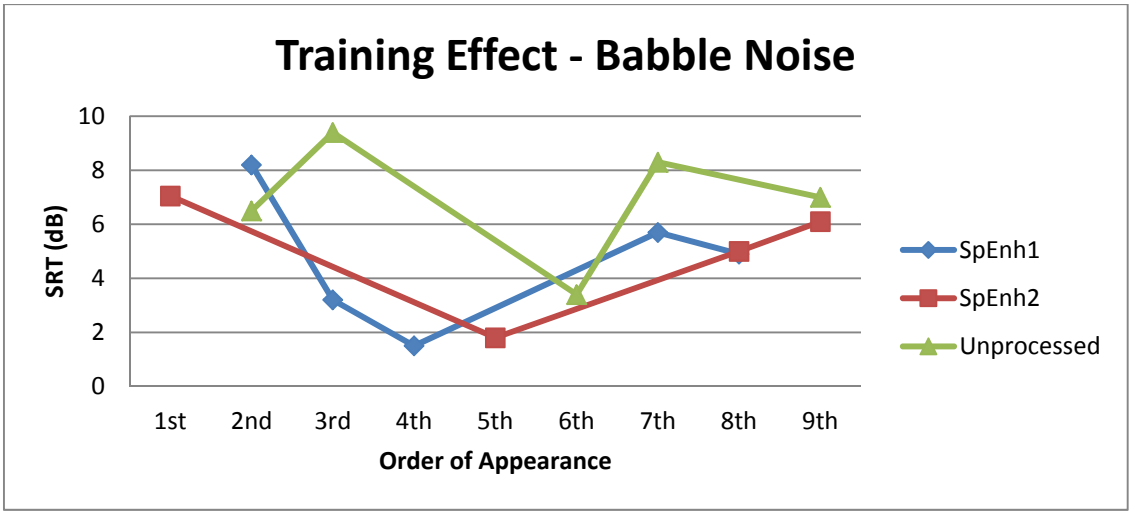


Figure 92: SRT of individual tests with respect to order of appearance. Babble noise.

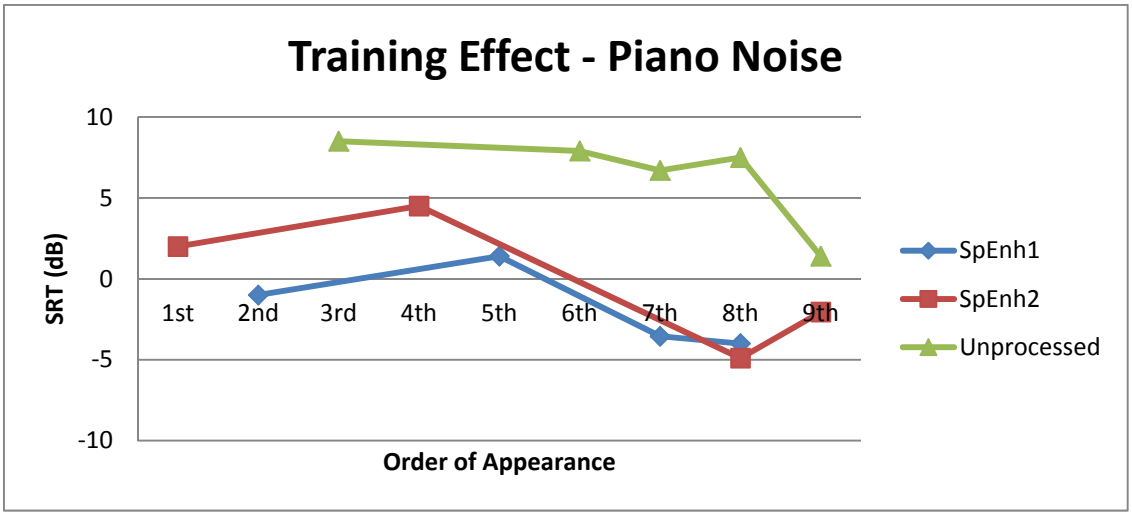


Figure 93: SRT of individual tests with respect to order of appearance. Piano noise.

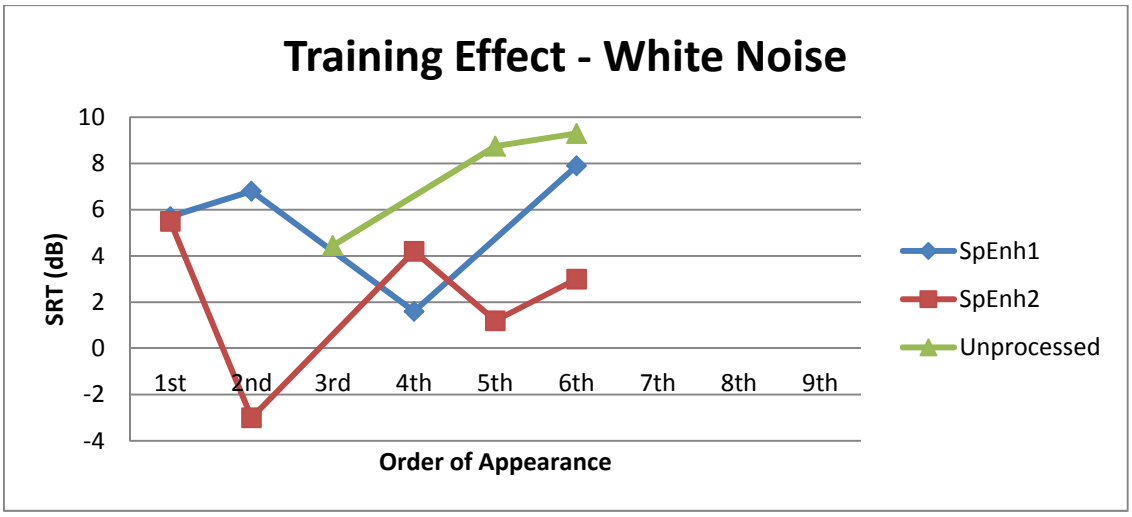


Figure 94: SRT of individual tests with respect to order of appearance. White noise.

The training effect, meaning improvement of performance as the order or appearance increases, can only be observed for piano noise. For the other two noise types, the test that takes place first usually exhibits a bad performance. However, it cannot be inferred that the performance improves as the order of appearance increases. In any case, the variability between different subjects should also be taken into account. Safe conclusions regarding the training effect cannot be made, unless the variability is eliminated.

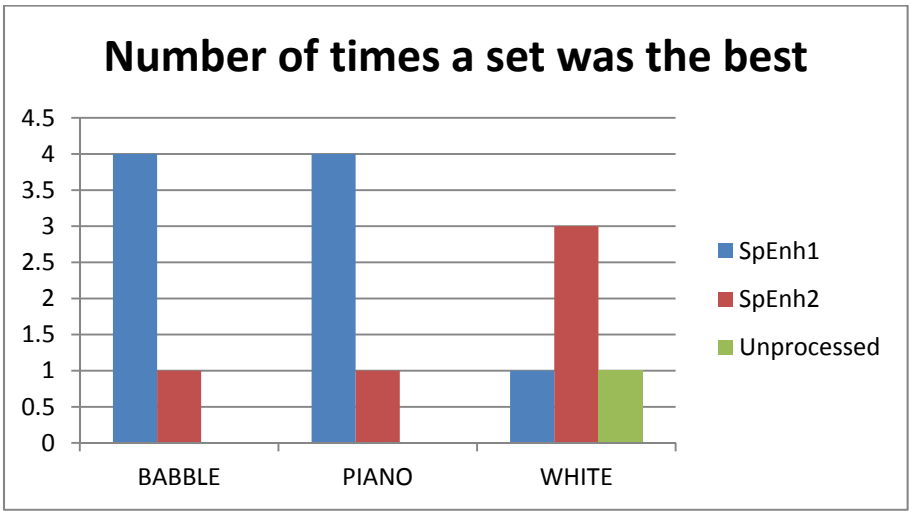


Figure 95: Set preference among 5 subjects for 3 noise types.

Finally, the number of times for which a certain enhancement condition was the most preferred one is illustrated in the histogram of Figure 95, for all the types of noise. Without doubt, SpEnh1 is the most preferred condition for babble and piano noise. This comes in agreement with NH people with the CI Simulator only for piano noise. Regarding white noise, SpEnh2 leads to better intelligibility in CI patients, as opposed to NH people with the Simulator, for which SpEnh1 is comparable to SpEnh2. SpEnh2 contains an artifact, which is reduced with the presence of the CI Simulator. Moreover, as previously observed, a CI has a similar but stronger effect than the CI Simulator. Therefore, a CI reduces the artifact of SpEnh2 even more, leading to optimal intelligibility. Finally, the Unprocessed condition was only once in 15 times (white noise) the most preferred.

F. Non Parametric Statistics

The median SRT measurements for both subject categories are summarized in Tables 12 and 13. An improvement in the median SRT is detectable when the enhancement algorithm is introduced, as analyzed in paragraphs IV.D and IV.E, and especially for piano noise. The CI patients, in general, exhibit larger median SRT values than the NH subjects. This indicates worse performance of CI patients in comparison to NH subjects. Finally, the preference to one of the first two enhancement conditions is consistent between the two subject categories in terms of the median, with the exception of babble noise.

	SpEnh1	SpEnh2	Unprocessed
BABBLE	3.80	3.65	4.80
PIANO	-6.20	-4.15	3.05
WHITE	-2.45	-2.55	2.80

Table 12: Median SRT of NH subjects

	SpEnh1	SpEnh2	Unprocessed
BABBLE	4.9	5.5	7.0
PIANO	-1.0	0.6	7.5
WHITE	5.7	3.0	8.6

Table 13: Median SRT of CI patients

The median SRT performances are, however, not adequate in order to indicate the differences between the three enhancement conditions. For this reason, non parametric Wilcoxon tests were conducted between are possible pairs that can be formed by the three conditions. The Wilcoxon method tests the null hypothesis that two related medians are the same. The result of the test is an asymptotic significance value. If this value exceeds a threshold (in this case 0.05), then the null hypothesis is retained. Otherwise, it is rejected. Therefore, the smaller the significance, the stronger the rejection of the null hypothesis and thus the bigger the difference between the related samples. The Wilcoxon tests were conducted using the SPSS software. Wilcoxon was selected as the appropriate testing method, because it doesn't assume a normal distribution of the data. The calculated significances are summarized in Tables 14 and 15 for both subject categories. The cases where the null hypothesis was retained are indicated with green, while the cases where the null hypothesis was rejected are indicated with red.

	SpEnh1-Unprocessed	SpEnh2-Unprocessed	SpEnh1-SpEnh2
BABBLE	0.416	0.116	0.138
PIANO	0.028	0.028	0.043
WHITE	0.043	0.074	0.463

Table 14: Asymptotic significances resulting from the Wilcoxon tests for NH subjects. (Red): Rejection of the null hypothesis. (Green): Retainment of the null hypothesis.

	SpEnh1-Unprocessed	SpEnh2-Unprocessed	SpEnh1-SpEnh2
BABBLE	0.043	0.043	0.225
PIANO	0.043	0.043	0.138
WHITE	0.500	0.080	0.225

Table 15: Asymptotic significances resulting from the Wilcoxon tests for CI patients. (Red): Rejection of the null hypothesis. (Green): Retainment of the null hypothesis.

For babble noise, there is no consistency between NH subjects and CI patients regarding the difference of the first two enhancement conditions from the Unprocessed condition. For NH subjects the null hypothesis is retained, meaning that there is no difference between the investigated samples, while for CI patients the null hypothesis is marginally rejected, indicating difference from the Unprocessed condition when the algorithm is introduced. Regarding piano noise, the Wilcoxon tests show that the SRT performance of both subject categories changes when the algorithm is introduced. This is more prominent for NH subjects, for which the corresponding asymptotic significances are further below the threshold than for CI patients. As far as white noise is concerned, a large inconsistency can be observed between NH subjects and CI patients regarding the pair SpEnh1-Unprocessed. While for NH people SpEnh1 differs from the Unprocessed condition, for CI patients the null hypothesis is retained with high significance. On the other hand, SpEnh2 is shown not to differ from the Unprocessed condition for both subject categories, but with a significance that lies very close to the threshold. Finally, the first two enhancement conditions do not differ from each other, with the exception of piano noise for NH people.

G. Conclusions & Comments

To begin with, as a general conclusion, it could be claimed that the application of the speech enhancement algorithm, improves the intelligibility of the subjects. This is indicated by a lower Speech Recognition Threshold (SRT). Regarding NH people with the CI Simulator, the SRT is reduced by 1.05 dB, 9.2 dB and 3 dB for babble, piano and white noise, respectively. As far as the CI patients are concerned, the corresponding values are 1.9 dB, 8.6 dB and 5.6 dB.

The largest intelligibility improvement can be observed for piano and white noise. On one hand, piano is a very structured noise and it is easy to train an efficient dictionary to represent it. On the other hand, white noise is unstructured and, therefore, rejected by the speech dictionary due to large incoherence to it. Babble noise is more challenging for a speech enhancement algorithm, as it not easily distinguishable from speech. The same phenomenon was also objectively observed in the previous chapter and is verified in the present chapter by subjective tests.

A lower SRT for the enhancement conditions where the algorithm is involved (SpEnh1&2), is not only reported in the average results, but also in the individual ones. In total 33 comparisons were made, in order to detect the preferred enhancement condition of individual subjects (11 subjects and 3 noise types for each). Only 2 of the 33 times, the Unprocessed condition was more preferable than SpEnh1&2 and 1 time equally preferable. This preference to the Unprocessed condition can be justified by the fact that the tests with SpEnh1&2, for the same noise type, were conducted in the very beginning of the test session, when there was not adequate training.

Regarding babble noise, there is no consistency between the preferred enhancement condition of NH people with the Simulator and the preferred enhancement condition of CI patients. NH people prefer SpEnh2, which is more smooth and contains less artifacts. CI patients exhibit better intelligibility improvement for SpEnh1, which provides a more artificial result with a larger degree of enhancement. For piano noise, without doubt, SpEnh1 is the optimal enhancement condition. The preferred condition of NH people for white noise is not clear. The median SRT indicates SpEnh2 as optimal. However, the individual preferences are in favor of SpEnh1. In general, the two conditions

are comparable for NH people. This does not apply for CI patients, for which SpEnh2 is obviously preferred. The optimal enhancement conditions of CI patients coincide with the ones that resulted from objective evaluation (paragraph IV.C).

By comparing the SRT values between CI patients and NH people with the CI Simulator, it was observed that the patients, in general, perform worse. This means that the CI Simulator does not cause adequate impairment to speech intelligibility. The degree of impairment can, however, be adjusted by decreasing the slope of the spectrum of the noise that is added by the Simulator. This difference in the degree of impairment can probably explain the stronger tendency of CI patients to prefer SpEnh2 in white noise. SpEnh2 generates an artifact that is very unpleasant before the CI Simulator. Nevertheless, this artifact becomes less apparent after the CI Simulator, leading to more intelligibility. It can be guessed that, as the CI effect is even more intense for CI patients than with the CI Simulator, this artifact is eliminated, leading to a strong preference for SpEnh2 by the CI patients.

An additional effect that takes place in a CI is the following. The SRT of the Unprocessed condition without the CI Simulator exhibits big differences among babble, piano and white noise (up to 12 dB). Speech in piano or white noise is much more intelligible (lower SRT) than speech in babble noise. However, when the CI Simulator is used with NH people or when the subjects are CI patients, the SRT differences become much smaller (up to 2 dB). This happens because piano and white noise are spread over the frequency spectrum because of the CI effect and mask speech, from which are no longer distinguishable as they used to be.

A phenomenon that is worth of being investigated in the variability of values that exists for the same test among the subjects. A possible explanation for this could be that the SNR in the output of the enhancement algorithm is not proportional to the SNR of the input, but varies around a value, leading to variability among the subjects. However, this phenomenon takes place also in the Unprocessed condition, where the algorithm is not involved. Furthermore, similarly, the output of the CI Simulator or of the CI does not have an SNR proportional to the input. This covers also the Unprocessed condition. Moreover, the CI patients exhibit a slightly larger variability than the NH people with the CI Simulator. This happens due to the fact that the degree of impairment varies a lot among CI patients. However, the variability for NH people is big and it cannot be justified by difference in hearing ability among them. The variability is not only big in the absolute SRT values, but also in the relative to the Unprocessed ones, although slightly smaller for the latter (babble and piano noise). Therefore, even if the individual performance abilities are equalized, the variability is not eliminated.

The cause of the variability phenomenon, mainly lies in the training effect. The performance of a subject usually improves as the subject becomes more familiar with the testing procedure and with speech in a certain noise type. Therefore, tests that appear early in the session come along with bad results. The training effect is more evident for NH people with the CI Simulator, for which the individual hearing abilities do not introduce additional variability. In addition, the training effect does not only involve training during the test session but also before. In order to receive results deprived of training effects, many training rounds should precede the test session for all noise types and with all enhancement conditions. However, this was not possible within the scope of this study, where fewer training rounds were conducted before the test session.

A minor observation that was made during the tests, was that a larger SNR does not always lead to better intelligibility. A very loud speech can cause aversion to the subjects, who unconsciously "refuse" to listen to the sentence. Furthermore, for CI patients there is sound clipping above a certain level. Therefore, exceeding this level does not contribute to intelligibility and sometimes it can be annoying as it reaches the comfort limit of the patients.

Finally, it is interesting to observe the mental procedure of sentence recognition. When the sentence is not immediately recognized, some metal effort is required. This effort is made at the expense of remembering the previously recognized words. While trying to understand the last words, the subject might have difficulty in remembering the first ones. Then confusions are made. For example, the subject might have heard "rote" in the 4th position, which is the correct one, but thinks he/she heard "teure". This is a phonetic reversion. Or the sentence might start with "Doris" and the subject will claim that "drei" was in the 3rd position. These two words sound similar, but the subject forgets in which position he/she heard that sound. Therefore, instead of choosing a word from the available ones for position 1 that sounds like this, he/she chooses a word from the available ones for position 3. Furthermore, "Dosen" was frequently confused with "Bilder. These two words appear dissimilar. Hence, either the phonetic distance between them decreases after the CI or the combination "malt Dosen" that often appeared, was replaced by "malt Bilder" in order to infer some meaningful content from the sentence.

V. WAVELET BASED DICTIONARY LEARNING METHOD

A. Introduction

The SE algorithm under investigation is accompanied with a high computational cost. The learning-based approach of an explicit dictionary from the training data, leads to a representation form that lacks any structure and is, thus, costly to apply. On the other hand, an analytic approach would lead to an implicit structured dictionary, with a fast implementation. However, a purely structured dictionary would not be adaptable to the data [17].

Several attempts have been made in the past to combine the advantages of both the learning and the analytic approach with the objective of creating fast and adaptable dictionaries. One suggestion is the design of dictionaries, which are specified by a set of trained parameters. The implementation suggested in [18], imposes a structure on these parametric dictionaries by promoting a correlation between the atoms. Accordingly, in [19] a double sparsity approach is proposed. There, the dictionary is the product of an implicit base dictionary with a trained sparse matrix. In this way, the dictionary atoms have some underlying structure over a fundamental dictionary. A modification of the K-SVD algorithm is suggested to train such a dictionary.

The work in [20] is the evolution of [19], where a wavelet dictionary is proposed as the base dictionary. In this way, a multi-scale approach is accomplished. The advantage of multiple scales is that sub-dictionaries corresponding to different data scales can be separately trained. The fact that these dictionaries consist of smaller atoms, operates in favor of the computational cost. Furthermore, an assertion made in [20], is that expressing the dictionary as the product of a fundamental wavelet dictionary with an explicit matrix is equivalent in the sparse coding problem to transferring the data in the wavelet domain and using an explicit dictionary. Although proposed for image applications, [20] was the main source of inspiration of the modification of the SE algorithm described in this chapter. In this modification, the data are transformed in the Wavelet domain instead of the STFT domain, by applying a discrete wavelet transform on them. The aforementioned equivalence principle is used also in [21] for image denoising, where a zero-tree structure of the wavelet coefficients is additionally applied.

In the audio processing field, wavelets have been used for several applications such as audio compression [22] and classification [23]. More related to speech in noise applications, in [24] a noise suppression algorithm for hearing aids based on wavelets is proposed. There, a Wiener filter for noise suppression is implemented in the wavelet domain. Similarly, in [25] an audio denoising scheme is proposed, which combines wavelets with a block attenuation method that eliminates residual noise. There, it is claimed that wavelets are more efficient than STFT in audio denoising. STFT is suitable for analyzing stationary parts of a signal, while wavelets capture transient features as well.

In conclusion, the modification of the SE algorithm that is described in this chapter is probably the first scheme that proposes the use of wavelets for dictionary learning aiming at speech enhancement. Here, both the training and enhancement data matrices are the wavelet coefficients of the corresponding signals. The objective is the reduction of the computational cost and the investigation of the potential of a multi-scale approach. A faster implementation can be made by reducing the size and, especially, the width of the data matrices that are sparsely coded. The modification proposed does not involve a patch-based approach and is thus expected to result in

more narrow data matrices. The wavelets, by representing the data at various scales, capture both the general and the detailed characteristics of the signals. Therefore, the role of tiling of the feature space by properly selected patches, which is omitted in this modification, is implied in the working scheme of wavelets. Finally, the training of sub-dictionaries corresponding to different scales or levels of decomposition is examined as well.

B. Frame Based Wavelet Reconstruction

A framework for real-time decomposition and reconstruction of a signal using wavelets was built prior to incorporating any speech enhancement. The processing steps comprising this framework are presented in Figure 96.

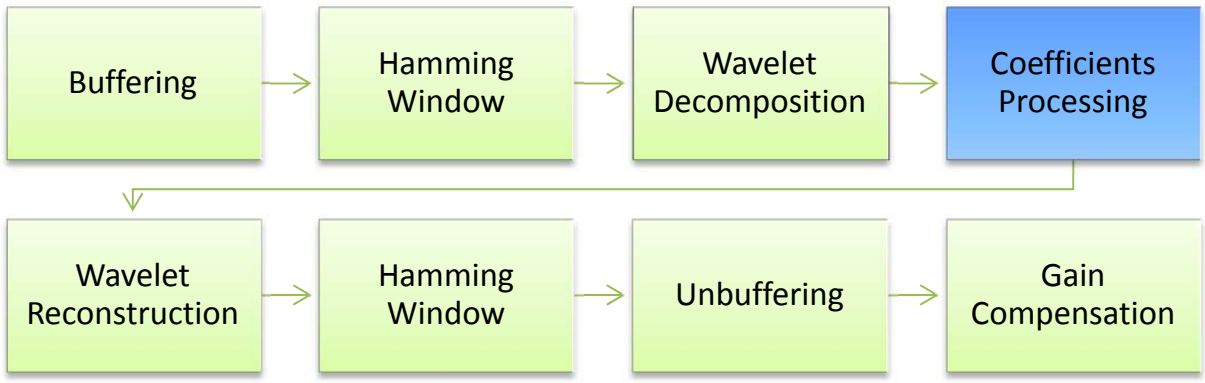


Figure 96: Processing steps of the real-time wavelet reconstruction framework.

In this system, the input signal is buffered with a specified window length and a desired overlap between the window segments. A hamming window is applied on the segments to eliminate the border effects. A discrete wavelet transform is, then, imposed on the segments leading to the coefficients of the wavelet decomposition. Any possible processing of the coefficients could take place at the next point. Following to this, reconstruction from the wavelet to the time domain takes place. A hamming window is applied on the segments before they are unbuffered by being overlapped-added. Finally, gain compensation is made on the final signal. This involves division by a unitary signal of the same length as the input, which has passed through the system, until after unbuffering. Speech enhancement will later be incorporated at the "coefficients processing" step, since the wavelet coefficients will serve as the data matrix, replacing the STFT coefficients.

The mean square error between the reconstructed and the input signal of this system was investigated with respect to the main parameters that affect its performance. These are the length and the overlap of the buffering window, the wavelet type used and the levels of the discrete wavelet transform. Moreover, the need of applying a hamming window was examined. Finally, the effect of the parameters was searched in the presence of noise in the coefficients. Uniformly random noise within a specified range was added to the coefficients. This range was

$$[-\text{coefficient value} \times \text{noise amount} \quad + \text{coefficient value} \times \text{noise amount}]. \qquad (20)$$

To begin with, the reconstruction mean square error was measured in relation to the noise amount (Figure 97) for a clean speech sentence. The noise amount varied within various levels of magnitude. Two of them are illustrated. The window size was 50 msec with 40 msec overlap and the wavelet 'db8' was used with 7 levels. By subjective listening, a noise amount greater than 1 is disturbing, while for more than 90, the sentence becomes unrecognizable.

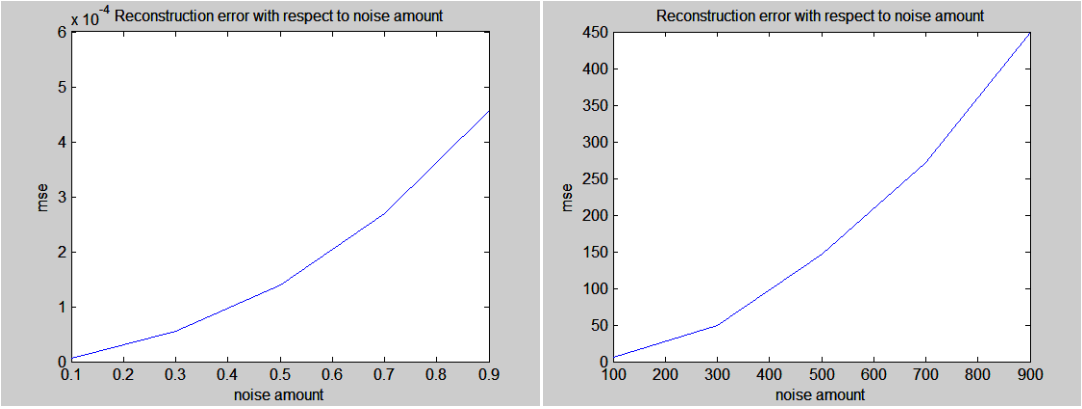


Figure 97: Reconstruction error with respect to noise amount.

The effect of the Hamming window on the reconstruction error was investigated with respect to the noise amount. Four cases were compared: application of Hamming window both before wavelet decomposition and after wavelet reconstruction, application of Hamming window only before wavelet decomposition, application of Hamming window only after wavelet reconstruction and application of no Hamming window. It is shown in Figure 98, that a smaller reconstruction error was achieved without a Hamming window. However, by subjective listening a crackling noise artifact was detected without the Hamming window and for large noise amount. For this reason, it was decided to use a Hamming window. The noise amount examined was 0.1, 5 and 10. The window size was 50 msec with 40 msec overlap and the wavelet 'db8' was used with 7 levels. Also 'haar' wavelet was tried. It produced the error of the same order of magnitude, but no subjective difference was detected.

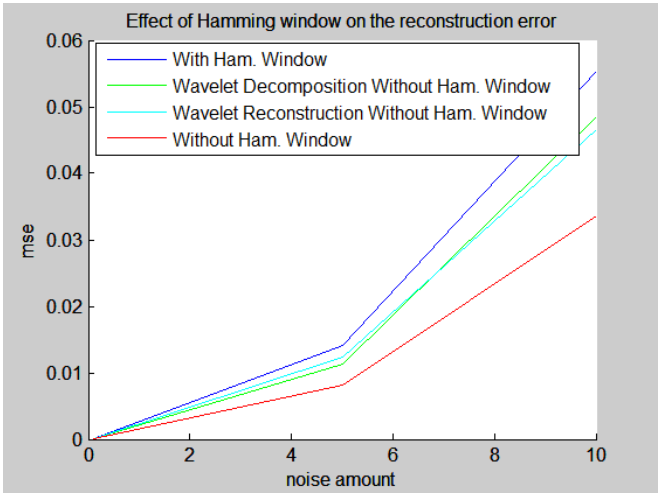


Figure 98: Effect of Hamming window on the reconstruction error with respect to noise amount.

The error with respect to the number of decomposition levels is illustrated in Figure 99 for 'db3', 'db8' and 'haar' wavelets. Three amounts of noise were examined, 0, 0.3 and 10. It can be observed that for no noise, 'haar' minimizes the error. However, the order of magnitude of the error is anyway too small. At the presence of noise, 'db8' seems to lead to a smaller error. The number of levels doesn't seem to play any role when noise is present. The same experiment was repeated many times, as the noise is randomly added. The results resembled Figure 99 only for no noise. By subjective listening, the reconstructed files sound the same when no noise is added regardless of wavelet or number of levels. At the presence of noise, the reconstructed sound is more clear for more levels and for 'db' in comparison to 'haar'. At the experiments, the window size was 50 msec with 40 msec overlap. Finally, it should be mentioned that a larger number of levels would increase the processing time of a real-time system, as additional delay would be introduced at each level of the application of the filters that perform wavelet decomposition.

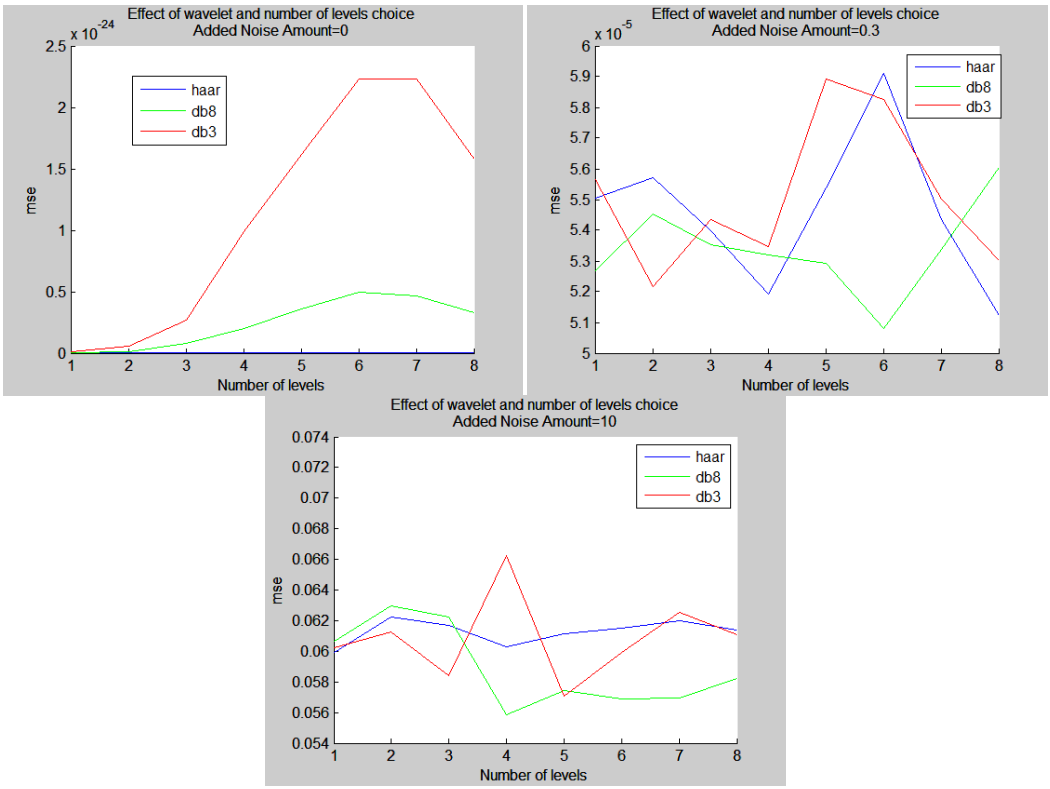


Figure 99: Effect of wavelet choice and number of levels on the error. (Up Left): noise amount 0. (Up Right): noise amount 0.3. (Down): noise amount 10.

The reconstruction error with respect to the buffering window size is illustrated in Figure 100 for 3 different noise amounts. The wavelet 'db8' with 6 levels was used. The overlap of the window was 4/5 of the window size. The window sizes examined were 5, 10, 20, 50 and 100 msec. It can be seen that the error increases with the increase of the window size. Therefore, a small window is preferable. By subjective listening, the window size doesn't play any considerable role.

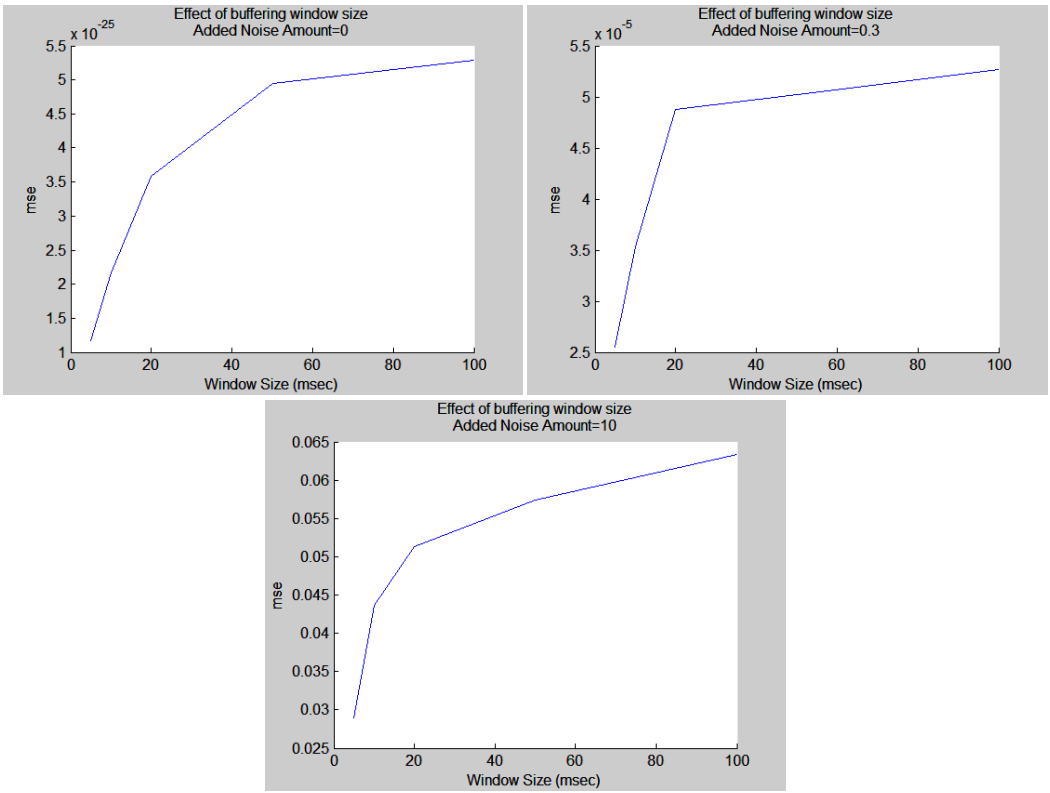


Figure 100: Effect of window size on the error. (Up Left): noise amount 0. (Up Right): noise amount 0.3. (Down): noise amount 10.

The window size, nevertheless, determines the size of the coefficients matrix that would be sparsely coded if speech enhancement was incorporated. As it can be observed in Figure 101, a larger window leads to a higher and more narrow matrix and vice versa. As explained in Chapter III, a faster sparse coding calls for a narrow matrix. For this reason, a large window would be preferred. However, a large window increases the delay of a real-time system. Therefore, the choice of the window size is a tradeoff between the sparse coding time and the delay of the system.

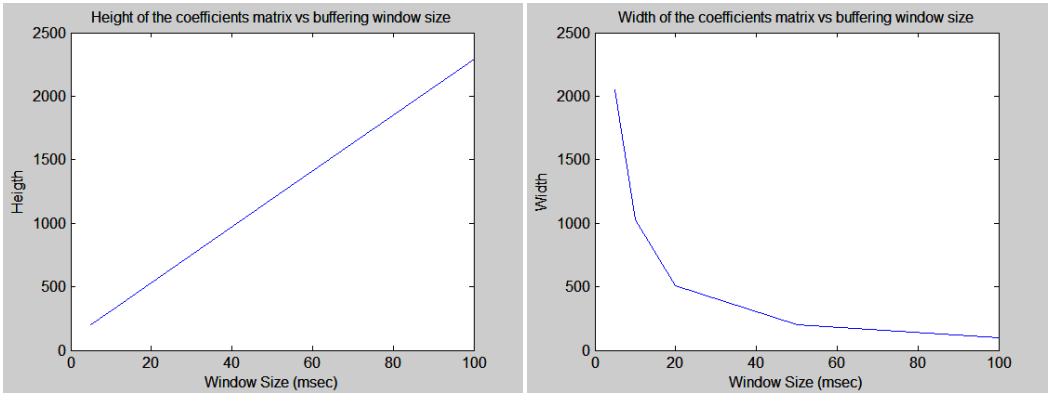


Figure 101: Effect of window length on the size of the coefficients matrix . (Left): matrix height. (Right): matrix width.

Finally, the optimal overlap of the buffering windows, expressed as a fraction of the window length, was investigated. In Figure 102, it can be observed that a large overlap leads to a smaller error. This can be slightly audible only for a big noise amount. The 'db8' wavelet with 6 levels and a 50 msec window were used.

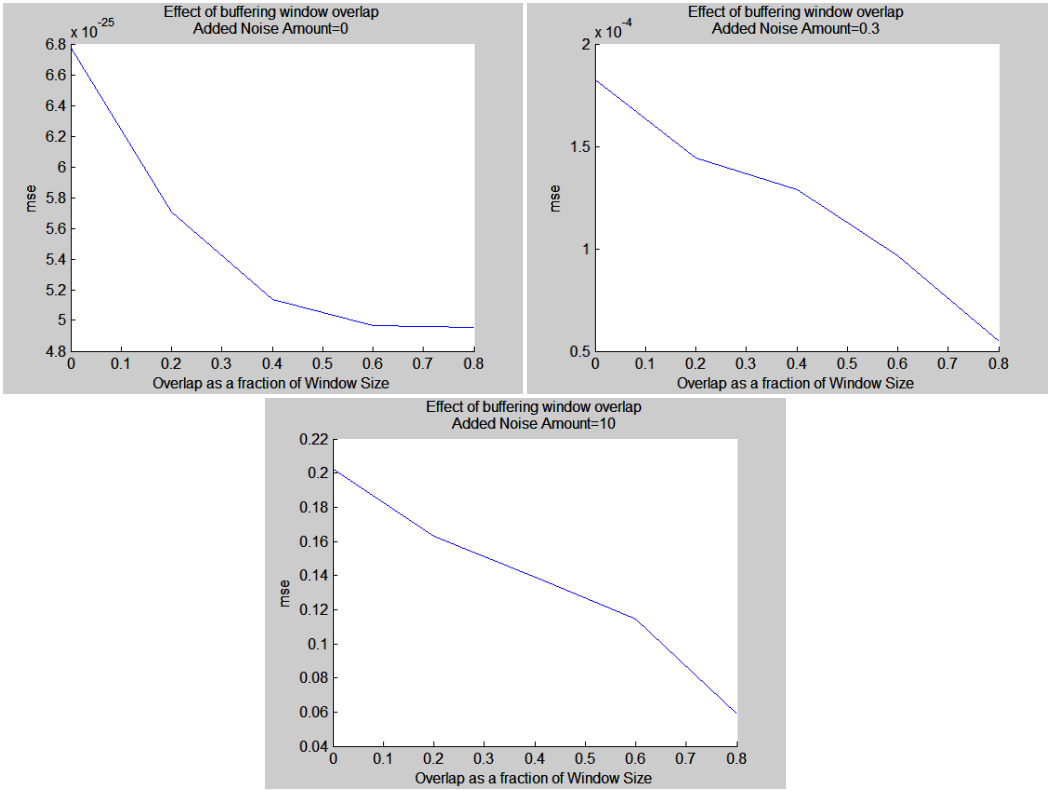


Figure 102: Effect of window overlap on the error. (Up Left): noise amount 0. (Up Right): noise amount 0.3. (Down): noise amount 10.

Regarding the size of the coefficients matrix, its height depends only on the window length and not on the overlap (Figure 103). However, a smaller overlap leads to a more narrow matrix. Therefore, it is preferable in terms of the sparse coding computational cost .

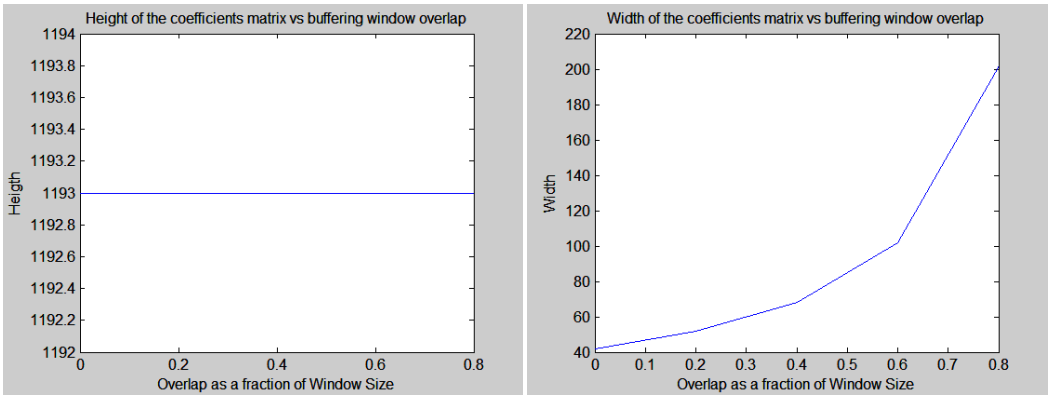


Figure 103: Effect of overlap on the size of the coefficients matrix . (Left): matrix height. (Right): matrix width.

In conclusion, a hamming window was decided to be used both before the wavelet decomposition and after the wavelet reconstruction in order to eliminate a crackling noise artifact generated without the Hamming window and for a large noise amount. The window's computational cost is, anyway, negligible. Furthermore, the 'db' wavelet is a safe choice. The 'haar' wavelet is only used for educational purposes and not for real applications. In this example, 'haar' increases the reconstruction error at the presence of noise. As far as the number of levels is concerned, a larger number is preferable. However, it would also lead to a longer delay in a real-time system. Regarding the buffering window length, it does not affect the audible outcome. However, it is the tradeoff between the sparse coding computational cost, which is decreased with a large window, and the delay of a real-time system, which would be longer with a large window. Finally, the error is slightly decreased with a large overlap of the buffering windows, but at the same time the sparse coding computational cost increases.

C. Wavelet Based Speech Enhancement

For the implementation of the wavelet based Speech Enhancement method, additional processing was incorporated in the "Coefficients Processing" step of the real-time wavelet reconstruction framework (Figure 96). It consists of two parts. In the first part, the wavelet coefficients matrix of the mixed signal is coded on the composite dictionary of speech and interferer using the LARC algorithm. In the second part, the speech component is isolated as in the standard Speech Enhancement algorithm. The wavelet coefficients of the estimated speech component are transformed in the time-domain resulting into the enhanced signal (Figure 104). Training of the speech and interferer dictionaries is made accordingly to the standard SE method, using the K-SVD algorithm. However, in the wavelet based method, the training data matrix consists of wavelet coefficients instead STFT coefficients. Furthermore, tiling of the feature space with overlapping blocks is omitted.

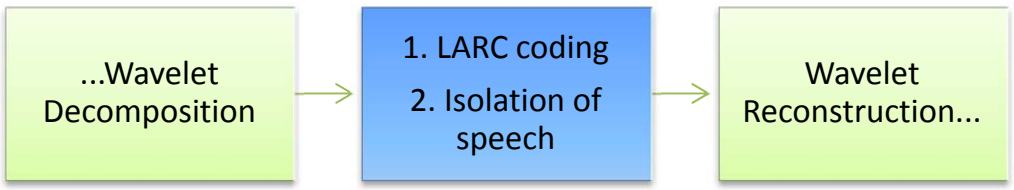


Figure 104: Incorporation of Speech Enhancement in the real-time wavelet reconstruction framework.

The wavelet parameters used were: buffering window of 50 msec, window overlap of 40 msec, Daubechies 8 wavelet and 2 levels of decomposition. Two levels of decomposition result in three scales in the coefficient vector. The A2, D2 and D1, written in ascending order with respect to the frequencies to which they correspond. "A" stands for "approximate", while "D" for "detailed". For the training, the residual coherence threshold was equal to 0.2, the dictionaries consisted of 1000 atoms each and 20 iterations were used for the K-SVD.

A processing detail with a large impact needs to be mentioned at this point. The wavelet coefficients are normalized after wavelet decomposition and back-normalized before wavelet reconstruction, both in training and in enhancement. In fact, a gain is applied to the coefficients of the D2 and D1

scales, so that their maximum absolute value, reaches the maximum absolute value of the A2 scale. The maximum absolute values are measured jointly for the speech and the interferer from the training data matrices. The coefficients of the A2 scale, have larger values that the ones of the detailed scales. The objective of the normalization is that they are treated with equal importance by the algorithm. Figure 105 illustrates the wavelet coefficients matrix of a degraded speech signal (with white noise at 0 dB SNR) without and with normalization. The rows corresponding to A2, D2 and D1 are 1-211, 212-422 and 423-829, respectively. The audible contribution of normalization is a more bright enhanced speech signal.

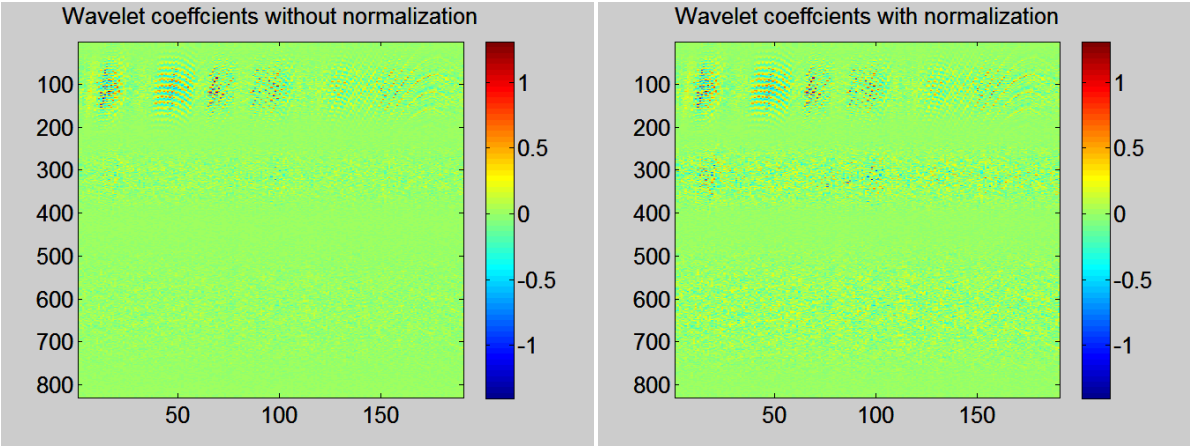


Figure 105: Wavelet coefficients of a degraded speech signal. (Left): without normalization. (Right): with normalization.

Figure 106 shows how the mixed file of Figure 105 is separated into the speech and interferer components. The corresponding normalized wavelet coefficients are illustrated. The residual coherence threshold used was equal to 0.12. This value provides the optimal speech quality with an adequate speech enhancement. However, it contains a musical noise artifact. A larger value such as 0.18, contains almost no noise (white or musical), but the speech is not so clear as for smaller values.

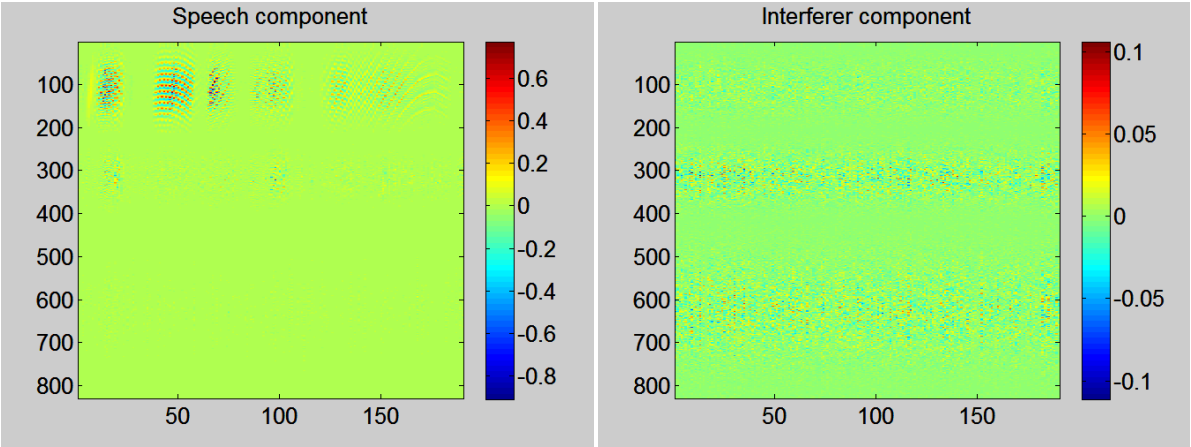


Figure 106: Normalized wavelet coefficients after SE. (Left): speech component. (Right): interferer component.

Several approaches were followed in order to improve the performance of the wavelet based SE method without any particular success. Nevertheless, they are mentioned below as they provide additional insight into the algorithm.

To begin with, an attempt was made to achieve better speech representation with the same residual coherence threshold during enhancement (μ). For this reason a small residual coherence threshold, equal to 0.05 instead of 0.2, was tried during training (μ train). It was observed that a speech dictionary resulting from a small μ train, works better for speech representation only with a small μ during enhancement. However, a small μ except for requiring a huge computational time, also leads to extreme source confusion. Enhancement with the standard μ (0.12) and a speech dictionary trained with μ train equal to 0.05, leads to a more intense and rough speech, which could possibly be preferable only without the CI Simulator. Moreover, the computational time only depends on μ and is the same regardless of μ train. Therefore, the selection of the value of μ for the training of the speech dictionary is a choice between an intense and rough speech quality (μ train=0.05) or a more natural one (μ train=0.2). On the other hand, a larger μ train for the speech dictionary equal to 0.35 was tried as well. For this value, with the standard μ during enhancement, the noise amount is the same in the enhanced signal, but the speech is badly represented. In the end, it was decided to use μ train equal to 0.2. Similarly the μ train of the interferer dictionary was investigated as well, by using the values 0.1 and 0.05. The computational time during enhancement was not influenced by the application of a dictionary trained with a different μ train, but also no improvement in the performance was noticeable. Therefore, the standard μ train equal to 0.2 was used for the interferer dictionary as well.

In general, the challenge in this dictionary learning method, is to create such dictionaries that will offer good representation of the speech and the interferer at the same μ during enhancement, while maintaining a "distance" between them. On one hand, a good speech representation would have a benefit on the intelligibility and quality of speech in the enhanced file. On the other hand, a good interferer dictionary would "attract" the interferer component of the mixed signal, leading to less source confusion. However, for the same μ, the speech and the interferer signal classes do not have an equally good representation by their corresponding dictionaries. As the right proportion between the representation of the speech and the representation of the interferer for the same μ was not achieved by changing the μ during training, it was tried to weigh the interferer dictionary accordingly during enhancement. For this reason, the composite dictionary was the

$$D = [Ds \quad w_i Di]. \tag{21}$$

It was observed that changing the weight of the interferer dictionary has a similar effect to changing the residual coherence threshold. A large weight resembles the case of a large μ. This could possibly be explained by the fact that the desired residual coherence of the coding on the composite dictionary is achieved faster when the interferer dictionary has large values imposed by weighting. At the same time, the part that is coded on the speech dictionary is less detailed than before, leading to an outcome resembling that of a larger residual coherence threshold.

In addition, the use of more levels of decomposition was investigated. For this reason, 5 levels were compared with 2 levels. No contribution was made to the performance by more levels. On the contrary, the enhanced signal was less bright. Regarding LARC computational time, it is not affected by the number of levels, as the coefficient vector maintains the same length. However, in a real-time system, more levels would lead to longer delay in the application of filters performing the wavelet decomposition.

Moreover, another two factors that influence dictionary training are the initialization of the dictionary and the number of iterations of the K-SVD. The dictionary training can be initialized with randomly selected columns from the training data matrix or by unitary atoms. None of the aforementioned approaches influenced the outcome.

Furthermore, it was suspected that the use of different wavelets would probably increase the distance between the speech and the interferer dictionary leading to better separation. Therefore, the choice of wavelet could be noise dependent. For this reason, the 'bior2.8' and the 'coif3' were tried, without making any difference to the outcome. Anyway, the wavelet families supported by a discrete wavelet analysis are limited.

Based on the speculation that certain scales might contain more information required to increase the distance between the speech and the interferer dictionary, it was decided to amplify the corresponding coefficients in order to increase their impact. The coefficients of the scales A2, D2 and D1 were separately amplified both during training and enhancement, but the outcome was even worse than without any amplification.

A last approach that was investigated involved the normalization of the coefficients of all scales between 0 and 1. This can be achieved by finding the maximum and minimum coefficient value from the training matrices, jointly for the speech and the interferer and separately for every scale and by applying the following normalization rule to the coefficient values:

$$value = \frac{value - \min_{levelx}}{\max_{levelx} - \min_{levelx}}. \tag{22}$$

This normalization provided extremely good representation of the speech and the interferer when coded individually with the corresponding dictionaries, for the same residual coherence threshold. Although very slow, this normalization seemed promising, as it highlighted the coefficients of all scales. However, when applied for speech enhancement, it failed to separate the speech from the interferer component. Despite the fact that it provided a very good representation of the mixed file coded on the composite dictionary, due to the non-linearity of the normalization transformation, this representation was not directly separable in order to isolate the speech component. This problem was partly solved by a suggestion involving the pseudo inverse calculation described in Appendix D. The separation was achieved, but loud 'beep' tones were generated at the positions of the ¼ and ½ of the sampling frequency. These tones were removed by filtering. In the end, no improvement was reported in the performance.

Finally, the performance was investigated in relation to the overlap of the buffering windows. It was shown that when the overlap is reduced by half, there is no noticeable deterioration in the enhanced signal. However, the computational time of LARC coding is downsized to approximately half of what it used to be. This can be justified by the fact that a smaller overlap leads to a smaller width of the coefficients matrix that is sparsely coded.

D. Wavelet Based Speech Enhancement with Scale Sub Dictionaries

A modification of the wavelet based SE method is presented in this paragraph. Here, LARC coding is performed separately for each scale of decomposition. This entails the training of separate dictionaries from the training coefficients of each scale.

More specifically, given that 2 levels of decomposition are used, three consecutive calls of the LARC function take place during enhancement, after wavelet decomposition. These perform coding of the coefficients of the scales A2, D2 and D1 on their corresponding concatenated (speech together with interferer) dictionaries. The speech component is isolated for each scale, by keeping only the part that is coded on the speech dictionary. Finally, the speech components of all scales are joined together, forming the coefficients matrix of the estimated speech signal. Wavelet reconstruction follows. Similarly, three calls of the K-SVD are required, in order to train the three dictionaries that correspond to the three scales. Each one of these calls receives as input only the part of the training coefficients matrix that corresponds to the desired scale.

The coefficients matrix of either the training data or the mixed signal can be separated into the three different scales by dividing the matrix in three groups of rows. The upper rows belong to A2, the middle rows to D2 and the bottom rows to D1. The same number of rows belong to scales of the same level, such as A2 and D2, while approximately double rows belong to D1. As the different scales are treated independently, no normalization of the coefficients is required.

The major benefit of coding the different scales separately, is that a different residual coherence threshold can be used in each coding procedure. This offers a broader room for optimization depending on the noise type. However, the coherence between the scales is not maintained in the dictionary learning part, leading thus to loss of information.

The frequency analysis of Figure 107, shows the influence of changing the residual coherence threshold (μ) of a scale during LARC coding, on a certain range of frequencies. In this example, a speech signal was degraded with white noise at 0dB SNR and was enhanced by the algorithm. Pink represents the reference enhanced signal, where μ equals 0.2 for all scales. In the yellow curve μ of A2 was 0.5, in the green curve μ of D2 was 0.5 and in the cyan curve μ of D1 was 0.5. When the μ of a scale changed, the μ of the remaining scales was set to the reference value of 0.2. The influence on the related frequency bands is obvious, as A2 corresponds to low frequencies, D2 to middle frequencies and D1 to high frequencies.

Regarding the benefit of the separate method in relation to the computational cost, it increases with the increase in the residual coherence threshold (μ). At the reference value μ=0.2, the total computational time of the separate version is comparable to the standard one. For lower values of μ, the standard method is clearly preferable. For example, for μ=0.1, the separate method is 6 times slower. However, for μ larger than 0.2, the computational time of the separate method decreases dramatically in relation to the one of the standard method. For example, for μ=0.3, the separate method is 5 times faster. In any case, the major benefit of the separate method in terms of the computational time, is that the three calls of the LARC function are independent. Therefore, they could be executed in parallel in a real-time system, gaining up to almost three times more speed, as measured.

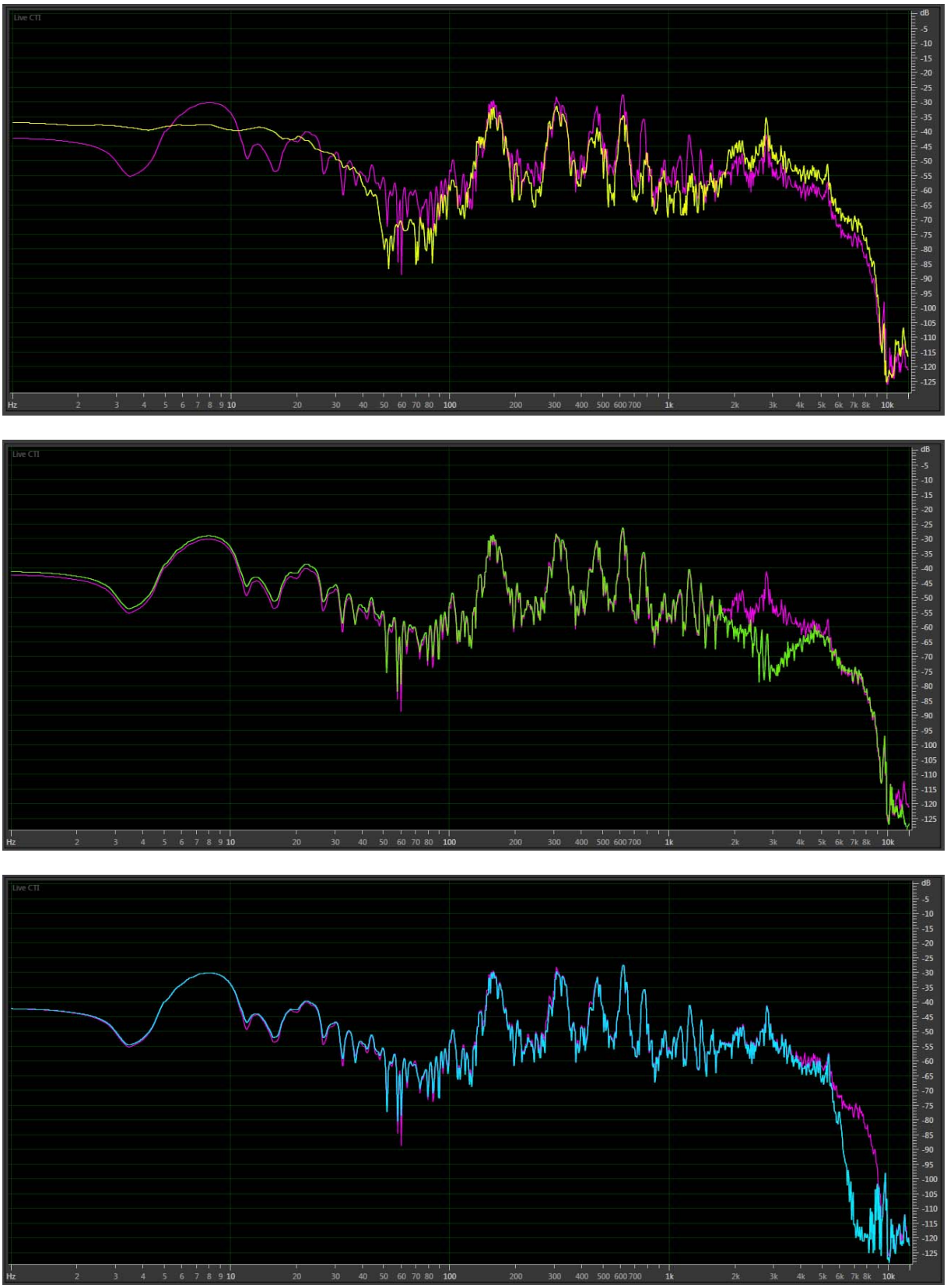


Figure 107: Frequency analysis of the enhanced signal in relation to the residual coherence threshold of the separately coded scales. For the reference (pink), μ=0.2 for all scales. (Up): μ of A2 equals 0.5. (Middle): μ of D2 equals 0.5. (Down): μ of D1 equals 0.5.

Moreover, the residual coherence threshold during training was investigated in relation to the algorithm's performance both for the speech and the interferer dictionaries. It was shown that it has a negligible influence.

Finally, the use of more than 2 levels of decomposition was examined. A circular optimization was followed for the 6 residual coherence thresholds of the different scales of a 5 level decomposition. The enhancement performance was comparable to the one with 2 levels. In addition, the total computational time was the same, both according to the serial and parallel approach of executing the LARC functions. The computational time for D1 is the same regardless of the number of levels, while it is accordingly distributed among the remaining levels depending on their number. The use of more levels of decomposition offers greater flexibility in optimizing the residual coherence threshold at the cost of coherence among the levels.

E. Performance Evaluation

Three versions of the dictionary learning SE method are compared in this paragraph:

1) The standard version, where the feature space is the FFT domain (DL).
2) The modification of the standard version, where the feature space is the wavelet domain, without separate scale sub-dictionaries (DLW).
3) The modification of the standard version, where the feature space is the wavelet domain, with separate scale sub-dictionaries (DLW_SEP).

The evaluation was based on a speech file degraded with babble and white noise at 0 dB SNR. The parameters used for K-SVD training were common for all versions. More specifically, 1000 atoms comprised each dictionary, 20 iterations were performed in the K-SVD algorithm and the residual coherence threshold of training was 0.2. Furthermore, the wavelet decomposition was performed with 'db8' using 2 levels. In addition, the buffering window of DLW and DLW_SEP was set to 50 msec with 40 msec overlap.

The enhancement parameters are even more crucial for the performance. Regarding the DL version, the parameterization was based on the clinical tests conducted with NH people in Chapter IV. The parameter set 1 was chosen for white noise, while the parameter set 2 for babble noise. The aforementioned sets appeared as optimal by conducting tests with NH people (Figure 83 of paragraph IV.D). The parameters comprising sets 1 and 2 are listed in Table 10 of paragraph IV.C. The residual coherence thresholds during enhancement are listed in Tables 16 and 17 for DLW and DLW_SEP, respectively. It was aimed to make the comparison having chosen the optimal parameters.

NOISE TYPE	RESIDUAL COHERENCE THRESHOLD
BABBLE	0.07
WHITE	0.12

Table 16: Enhancement parameters of DLW

NOISE TYPE	RES. COH. THR. A2	RES. COH. THR. D2	RES. COH. THR. D1
BABBLE	0.2	0.2	0.2
WHITE	0.1	0.5	0.5

Table 17: Enhancement parameters of DLW_SEP

The fwSegSNR gains measured for each version are listed in Table 18 both for babble and white noise. The corresponding LARC computational times in seconds are presented in Table 19.

NOISE TYPE	DL	DLW	DLW_SEP
BABBLE	0.796	0.704	0.277
WHITE	3.819	2.912	0.910

Table 18: FwSegSNR gains of the three versions for babble and white noise.

NOISE TYPE	DL	DLW	DLW_SEP
BABBLE	22.42	10.42	4.46
WHITE	2.45	1.97	8.63

Table 19: LARC computational times (seconds) of the three versions for babble and white noise.

It can be observed that DL offers greater enhancement in terms of the objective measure, both for babble and white noise. DLW is comparable to DL, while DLW_SEP has a worse performance, especially for white noise. However, the benefit of DLW is the shorter LARC computational time, which is even half of DL's for babble noise. DLW_SEP is very slow for white noise, while very fast for babble noise.

The LARC computational time of the wavelet versions, DLW and DLW_SEP, can be reduced by using a smaller overlap in the buffering windows. It has been shown that an overlap of half duration doubles LARC's speed without significantly deteriorating the enhancement performance. Furthermore, the computational time of DLW_SEP can be reduced by almost 1/3, by executing the three LARC coding functions in parallel.

The computational time of an algorithm also involves the delay of a real-time system which would implement it. The delay can be expressed as the time duration of one processing frame. In the wavelet versions, the frame time has been set to 50 msec. In the DL version, the frame time depends on the length of the FFT transform. In the examples under investigation, an FFT of 1024 points for babble noise leads to a frame time of 64 msec, while an FFT of 256 points for white noise leads to a frame time of 16 msec.

The enhanced audio files under comparison, were subjectively evaluated by 6 listeners. The degraded speech files, for babble and white noise, were used as reference. The aforementioned audio files are provided as

- Audio_17: Speech degraded with babble noise at 0 dB SNR.
- Audio_18: Audio_17 enhanced with DL.
- Audio_19: Audio_17 enhanced with DLW.
- Audio_20: Audio_17 enhanced with DLW_SEP.
- Audio_21: Speech degraded with white noise at 0 dB SNR.
- Audio_22: Audio_21 enhanced with DL.
- Audio_23: Audio_21 enhanced with DLW.
- Audio_24: Audio_21 enhanced with DLW_SEP.

The same files after the CI Simulator are provided as Audio_25-32.

A frequency analysis of the enhanced files of all three versions, in comparison to the clean speech file, is provided in Figure 108, both for babble (up) and white (down) noise.

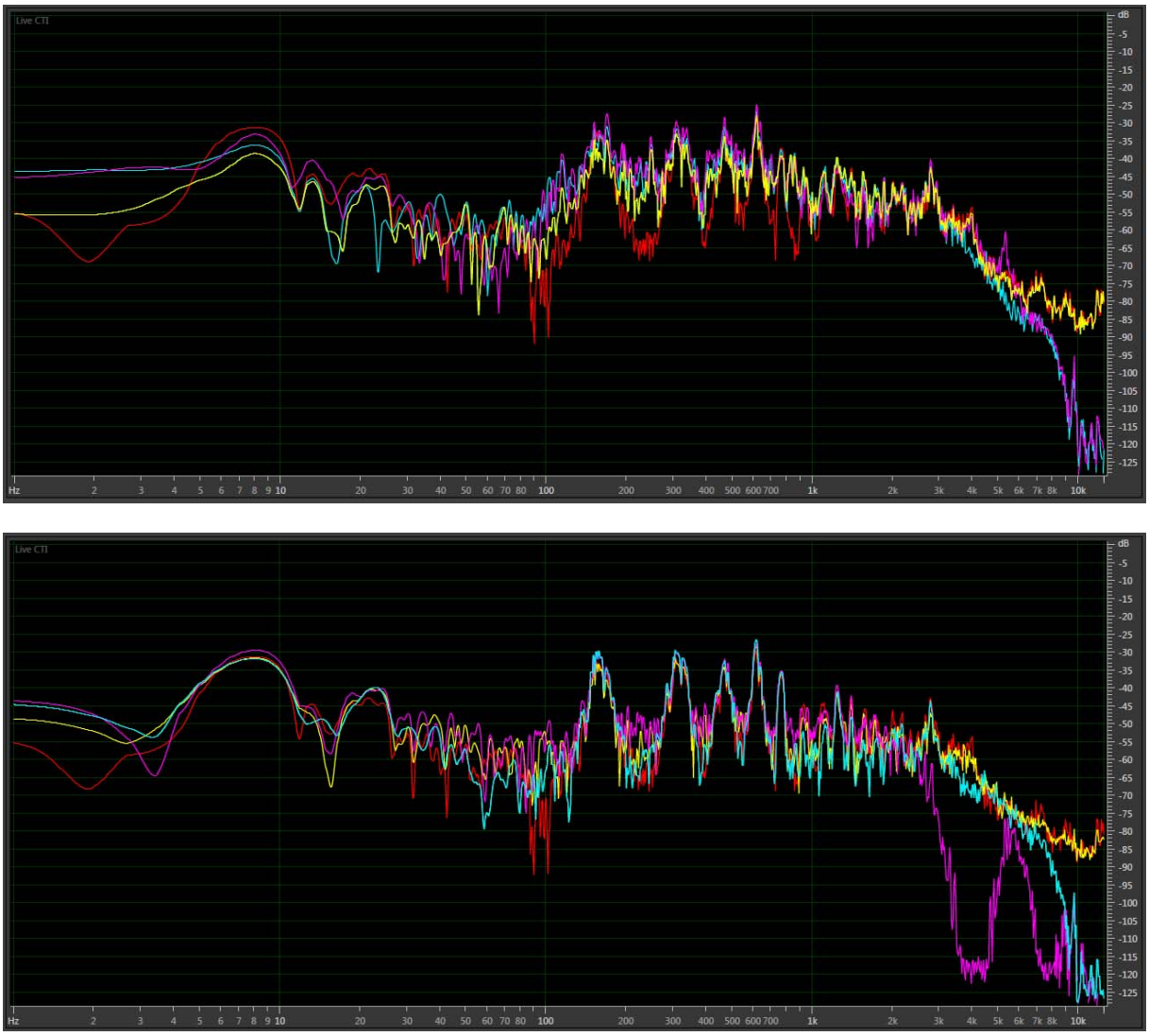


Figure 108: Frequency analysis of the enhanced signals in comparison to the clean speech signal. Red: clean speech signal. Yellow: enhanced signal with DL. Cyan: enhanced signal with DLW. Pink: enhanced signal with DLW_SEP. (Up): degradation with babble noise. (Down): degradation with white noise.

The listeners were asked to rate the 4 audio files under comparison in order of preference. Three ratings were made by each listener: one for babble noise, one for white noise and one for their overall impression about the algorithms regardless of noise type. A final score was assigned to each file for all three cases (babble noise, white noise and overall). The score was calculated by summing the individual scores of every listener. The first file in preference received 3 points, the second 2 points, the third 1 point and the last 0 points, from each listener. The scores are illustrated in Figure 109.

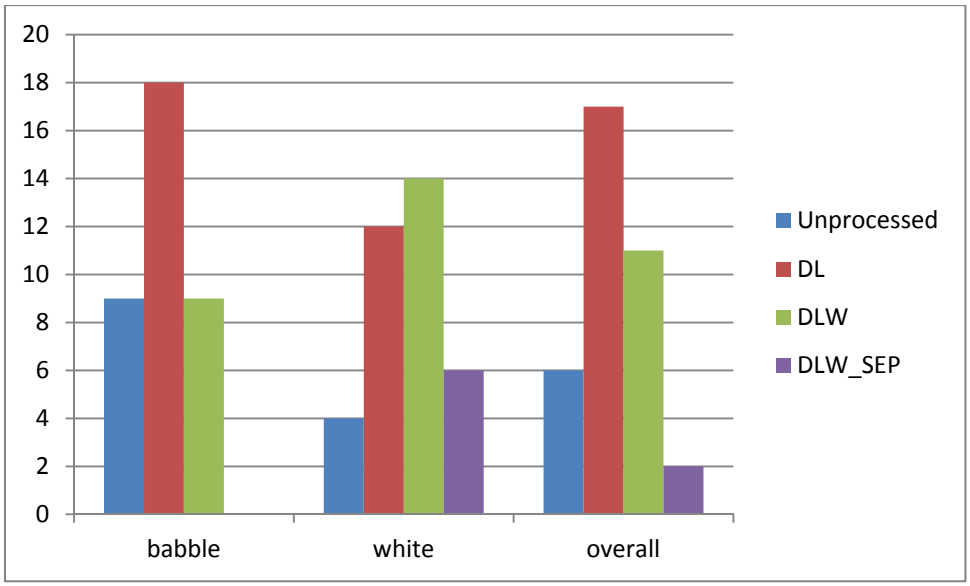


Figure 109: Evaluation of the 3 SE versions, together with the Unprocessed degraded speech file, by 6 listeners. Summed scores for babble noise, white noise and overall impression.

For babble noise, the most preferred algorithm is the standard DL. The DLW is equally preferable to the Unprocessed, while the DLW_SEP is for all listeners the least preferred. For white noise, DLW exhibits the highest score, while DL follows. DLW_SEP is the least preferred SE algorithm, however, it provides a certain degree of enhancement in relation to the Unprocessed. Regarding the overall impression about the algorithms, DL is the best with DLW following in preference. DLW_SEP is even less preferable than the Unprocessed.

A few comments on behalf of the listeners include the following. The Unprocessed files have a better quality than the ones after SE, but the intelligibility is low due to the presence of noise. Especially the high frequency components of white noise are very disturbing. DL increases the intelligibility of speech. However, a very pleasant artifact is generated for white noise. The speech, although intelligible, is accompanied with a rough unnatural “shadow”. DL for babble noise, leads to a noticeable suppression of the babble noise, which appears to have a lower energy than before. The speech quality is clear. In general, DL offers a good compromise between suppression and artifacts. DLW for white noise, increases the intelligibility of speech, which appears very clear. However, musical noise exists in the enhanced sound. This, besides being disturbing, makes white noise lose its character. For babble noise, DLW results in a more dull enhanced file than DL and it does not offer too much benefit in relation to the Unprocessed file. Regarding DLW_SEP, it leads to a dull enhanced file. However, it also reduces the intensity of white noise, making it much less disturbing.

The musical noise that is generated by DLW for white noise appears to be its biggest drawback. Musical noise can be decreased by increasing the residual coherence threshold during enhancement, at the cost of speech clarity. An enhanced file with very little musical noise and at the same time without significant speech deterioration is provided as Audio_33 to be compared with Audio_23. In Audio_33 the residual coherence threshold is 0.18 instead of 0.12. In addition, a larger threshold leads to faster coding.

Finally, it would be interesting to compare the output of the three SE versions, when clean speech or pure noise is provided as input. The outputs of the three SE versions are compared to the clean speech input in Figure 110, through a frequency analysis. A babble interferer dictionary was used in this example.

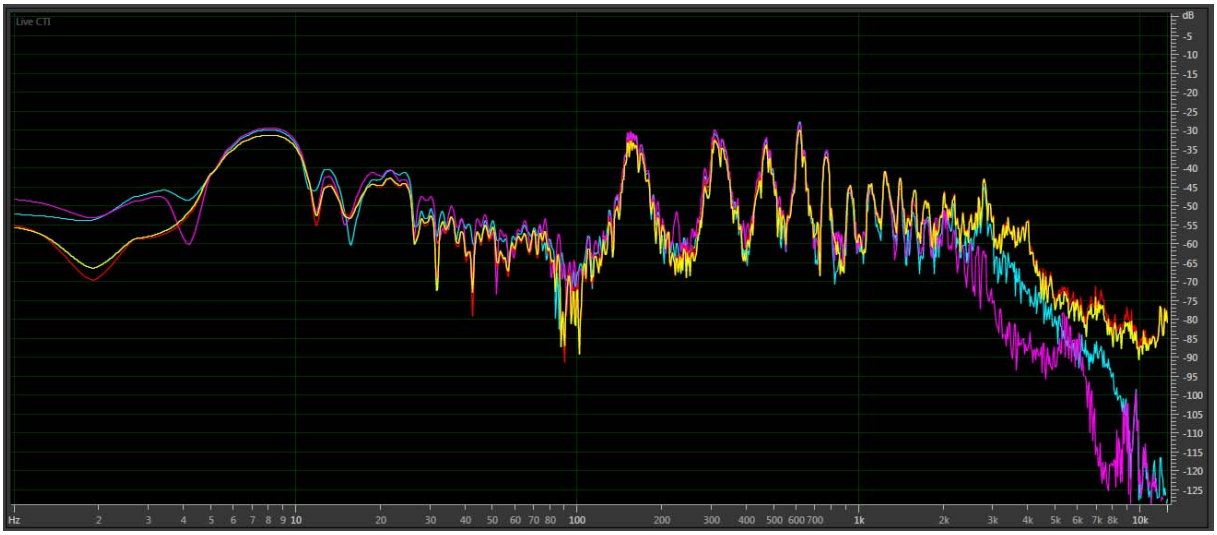


Figure 110: Frequency analysis of the enhanced signals in comparison to the clean speech signal, when the input is clean speech. Red: clean speech signal. Yellow: enhanced signal with DL. Cyan: enhanced signal with DLW. Pink: enhanced signal with DLW_SEP.

The fwSegSNR measured between the output and the input is 12.12, 10.64 and 9.51, for DL, DLW and DLW_SEP, respectively. Given that the maximum fwSegSNR -reported when the output is exactly the same as the input- is 35, the closest output to the input is provided by the standard DL. By listening to the enhanced files, the one generated with DL is without doubt the one with the highest resemblance to the clean speech file. Negligible differences can be detected. The enhanced files generated by DLW and DLW_SEP are more dull. Analogous results arise with the use of a white interferer dictionary.

When pure babble noise is given as input, the fwSegSNR gains measured are -0.19, -0.007 and 0.04, for DL, DLW and DLW_SEP, respectively. The remaining noise of the output can still be characterized as babble noise. However, it is clearly suppressed. For pure white noise as an input, the fwSegSNR gains measured are -0.59, -0.73 and -1.19, for DL, DLW and DLW_SEP, respectively. The output with DL sounds like babble noise, because of source confusion. The output of DLW contains a lot of musical noise. Finally, DLW_SEP results in a smooth output where the high frequency component has been suppressed. It could be said that it resembles car noise. In general, the fwSegSNR is not an appropriate measure in the case where pure noise is given as input. For all SE versions, there is noise suppression but not noise elimination. Babble noise maintains its character on the contrary to white noise. The effect of the SE algorithms on pure noise, is similar to the effect on the noise component that exists in the estimated speech signal when the input is speech degraded with noise.

F. Conclusions and Future Suggestions

Two modifications of the standard SE algorithm (DL) have been presented in this Chapter. In these modifications, the feature space where sparse coding takes place is the wavelet domain instead of the Fourier domain. Therefore, the matrices that are sparsely coded consist of wavelet coefficients. In the first modification (DLW), a single dictionary is trained for all decomposition scales. In the second version (DLW_SEP), the scales of wavelet decomposition are treated independently, by training separate dictionaries for each one of them.

The three SE versions were evaluated for speech degraded with babble and white noise, both objectively and subjectively. Parameters which optimize the SE performance, were chosen for each version. For white noise, it was shown that DLW has a comparable performance to DL. The decision about the most efficient version is formulated by the preference between a rough-shadow speech artifact for DL or a musical noise artifact for DLW. Regarding babble noise, DL is the most preferable version, as it provides clarity in speech, accompanied with an adequate degree of noise suppression. DLW offers babble noise suppression, however, the speech sounds more dull than in DL. DLW_SEP is the least preferred version for both noise types.

In general, the wavelet versions are expected to have a lower computation cost with respect to the standard DL, as they lead to data matrices of smaller width. However, this depends highly on the residual coherence threshold used during enhancement, on the noise type and on the geometry of the overlapping blocks in DL. For the parameterization of the evaluation in paragraph V.E, DLW is always faster than DL. DLW_SEP is faster than the other two versions for babble noise and slower than them for white noise. The computational time of DLW_SEP can be reduced by executing in parallel the LARC functions for sparse coding of the coefficients of each scale.

A few suggestions for future investigation of the wavelet based versions are the following. First of all, a patch based approach could be examined. The patches might lead to more wide matrices for sparse coding, but, on the other hand, they group neighboring coefficients, which share common characteristics. Furthermore, the information about the phase of the input signal is lost through the wavelet decomposition. If complex wavelets were used, the output signal could be reconstructed using the phase of the mixed input signal. This is the case in the DL version, where the phase of the STFT of the input signal is used for the enhanced signal. Moreover, dictionary training could be investigated further, in order to design speech and interferer dictionaries with low mutual coherence, leading to less source confusion. Alternatively, the speech dictionary could be trained in such a way, that it would provide better speech quality after enhancement, if possible, at the same residual coherence threshold. Finally, post processing could be applied, especially for the elimination of musical noise, which is the major drawback of enhancing speech in white noise.

VI. REFERENCES

[1] S. F. Boll, “Suppression of acoustic noise in speech using spectral subtraction”, in IEEE Trans. on Acoust., Speech and Signal Process., 1979, vol. 27, pp. 113–120.

[2] D. P. W. Ellis and R. J. Weiss, “Model-based monaural source separation using a vector-qunatized phase-vocoder representation”, in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., 2006, vol. 5.

[3] J. Sang, H. Hu, Ch. Zheng, G. Li, M. Lutman and S. Bleeck, “Evaluation of a sparse coding shrinkage algorithm in normal hearing and hearing impaired listeners”, in 20th Europ. Signal Process. Conf., 2012, pp. 1074-1078.

[4] C. D. Sigg, T. Dikk and J. M. Buhmann, “Speech enhancement using generative dictionary learning”, ETH Zurich, Computer Science Department, 2011.

[5] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, “Least angle regression”, in Annals of Statistics, 2004, vol. 32, pp. 407–499.

[6] R. Rubinstein, M. Zibulevsky, and M. Elad, “Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit”, Technical report, Technion, Computer Science Department, 2008.

[7] Y. Lu and P. C. Loizou, “A geometric approach to spectral subtraction” , in Speech Commun., 2008, vol. 50, pp. 453-466.

[8] L. M. Litvak, A. J. Spahr, A. A. Saoji and G. Y. Fridman, “Relationship between perception of spectral ripple and speech recognition in cochlear implant and vocoder listeners”, in J. Acoust. Soc. Am., 2007, vol. 122, pp. 982–991.

[9] J. Ma, Y. Hu, and P. C. Loizou, “Objective measures for predicting speech intelligibility in noisy conditions based on new band-importance functions”, in J. Acoust. Soc. Am., 2009, 125(5).

[10] Y. Hu and P. C. Loizou, “Evaluation of Objective Quality Measures for Speech Enhancement”, in IEEE Trans. on Audio, Speech and Lang. Process., 2008, vol. 16.

[11] S. Quackenbush, T. Barnwell, and M. Clements, “Objective measures of speech quality”, Englewood Cliffs, NJ: Prentice-Hall, 1988.

[12] D. Klatt, “Prediction of perceived phonetic distance from critical band spectra,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., 1982, vol. 7, pp. 1278–1281.

[13] J. Hansen and B. Pellom, “An effective quality evaluation protocol for speech enhancement algorithms,” in Proc. Int. Conf. Spoken Lang. Process., 1998, vol. 7, pp. 2819–2822.

[14] “Perceptual evaluation of speech quality (PESQ), and objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs,” ITU, ITU-T Rec. P. 862, 2000.

[15] K. Wagener, V. Kühnel and B. Kollmeier, “Entwicklung und Evaluation eines Satztests in deutscher Sprache I-III: Design, Optimierung und Evaluation des Oldenburger Satztests”, in Zeitschrift für Audiologie, 1999, vol. 38, pp. 86-95.

[16] A. Varga, H. J. M. Steeneken, M. Tomlinson, and D. Jones, “The noisex-92 study on the effect of additive noise on automatic speech recognition”, Technical report, DRA Speech Research Unit, Malvern, England, 1992.

[17] R. Rubinstein, A. Bruckstein and M. Elad, “Dictionaries for sparse representation modeling”, in Proc. IEEE, 2010, vol. 98, pp. 1045–1057.

[18] M. Yaghoobi, L. Daudet and M. Davies, “Structured and incoherent parametric dictionary design”, in IEEE Int. Conf. Acoust., Speech, Signal Process., 2010, pp. 5486–5489.

[19] R. Rubinstein, M. Zibulevsky and M. Elad, “Double sparsity: learning sparse dictionaries for sparse signal representation”, in IEEE Trans. Signal Process., 2010, vol. 58, no. 3, pp. 1553-1564.

[20] B. Ophir, M. Lustig and M. Elad, “Multi-scale dictionary learning using wavelets”, in IEEE J. of Sel. Topics in Signal Process., 2011, vol. 5, pp. 1014-1024.

[21] R. Liang, Z. Zhao and S. Li, “Image denoising using learned dictionary based on double sparsity model”, in IEEE 4th Int. Cong. on Image and Signal Process., 2011, pp. 691-695.

[22] G. Kemper and Y. Iano, “An audio compression method based on wavelets subband coding”, in IEEE Latin America Trans., 2011, vol. 9, no. 5, pp. 610-621.

[23] C. Lin, S. Chen, T. Truong and Y. Chang, “Audio classification and categorization based on wavelets and support vector machine”, in IEEE Trans. on Speech and Audio Process., 2005, vol. 13, no. 5, pp. 644-651.

[24] R. Hegner, H-D. Lang and G. M. Schuster, “LOCO: a high performance low complexity noise suppression algorithm using spatial information and Wiener-filtering”, University of Applied Sciences of Eastern Switzerland, Rapperswil, 2008.

[25] G. Yu, E. Bacry and S. Mallat, “Audio signal denoising with complex wavelets and adaptive block attenuation”, in IEEE Int. Conf. Acoust., Speech, Signal Process., 2007, pp. 869–872.

VII. APPENDIX

A. Study Plan

Title:

Speech Enhancement in Cochlear Implants

Introduction:

The enhancement of speech degraded by noise is a highly relevant task to enhance speech intelligibility for cochlear implant users. Today's cochlear implants incorporate speech enhancement algorithms. The goal of this thesis is to improve the speech intelligibility for cochlear implant users by a better algorithm.
Comparing performances and improvement evaluation of speech enhancers is a crucial step, since the output of an algorithm gets not reconstructed to an audio signal, but converted to generate stimulation patterns for the electrodes implanted in the cochlea. For this thesis a cochlea implant simulator is available (Leonid Litvak et al., 2007, Relationship between perception of spectral ripple and speech recognition in cochlear implant and vocoder listeners, J. Acoust. Soc. Am., 982–991) which allows to listen to an audio signal that is close to the one perceived by cochlear implant patients. It was shown that the intelligibility of a normally hearing person through this cochlear implant simulator is sufficiently similar to the intelligibility of a cochlear implant patient.

This master thesis project attempts to evaluate potential advancements in speech enhancement for CIs with an algorithm based on Generative Dictionary Learning (Christian Sigg et al., Speech Enhancement using Generative Dictionary Learning, submitted) and compare it to the state of the art speech enhancers in CIs.
Depending on the evaluation results of the already existing implementation of this algorithm with the CI simulator two options are foreseen. Either methods for potential improvements for speech enhancement have to be evaluated or necessary algorithmic changes that allow an implementation of the enhancement algorithm to a CI processor have to be identified. One of these two options have to be implemented and evaluated with the CI simulator.

Task List:

- Write a time-table of the work to be performed
- Review the relevant literature
- Performance evaluation of Speech enhancers with the CI simulator
- Evaluate potential methods to enhance the performance of the dictionary based speech enhancement algorithm
- Implement a method using Matlab or Simulink
- Evaluate the performance of the new algorithm with the CI simulator
- Write a report of the project

Supervisors:

Prof. N. Dillier, Laboratory for Experimental Audiology, ORL-USZ/D-ITET

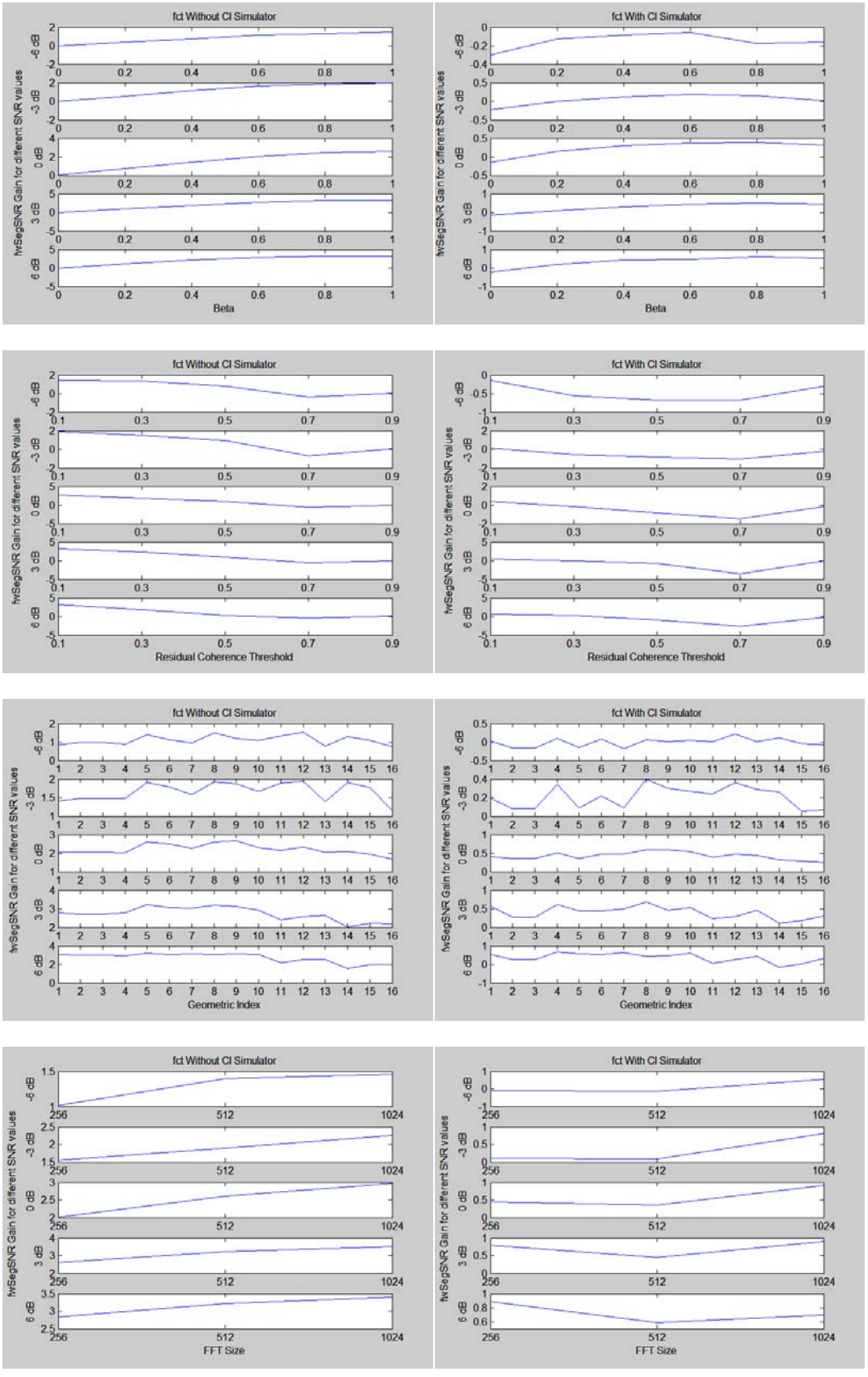
Dr. M. Feilner, Advanced Concepts and Technology, Phonak AG

Track Advisor:

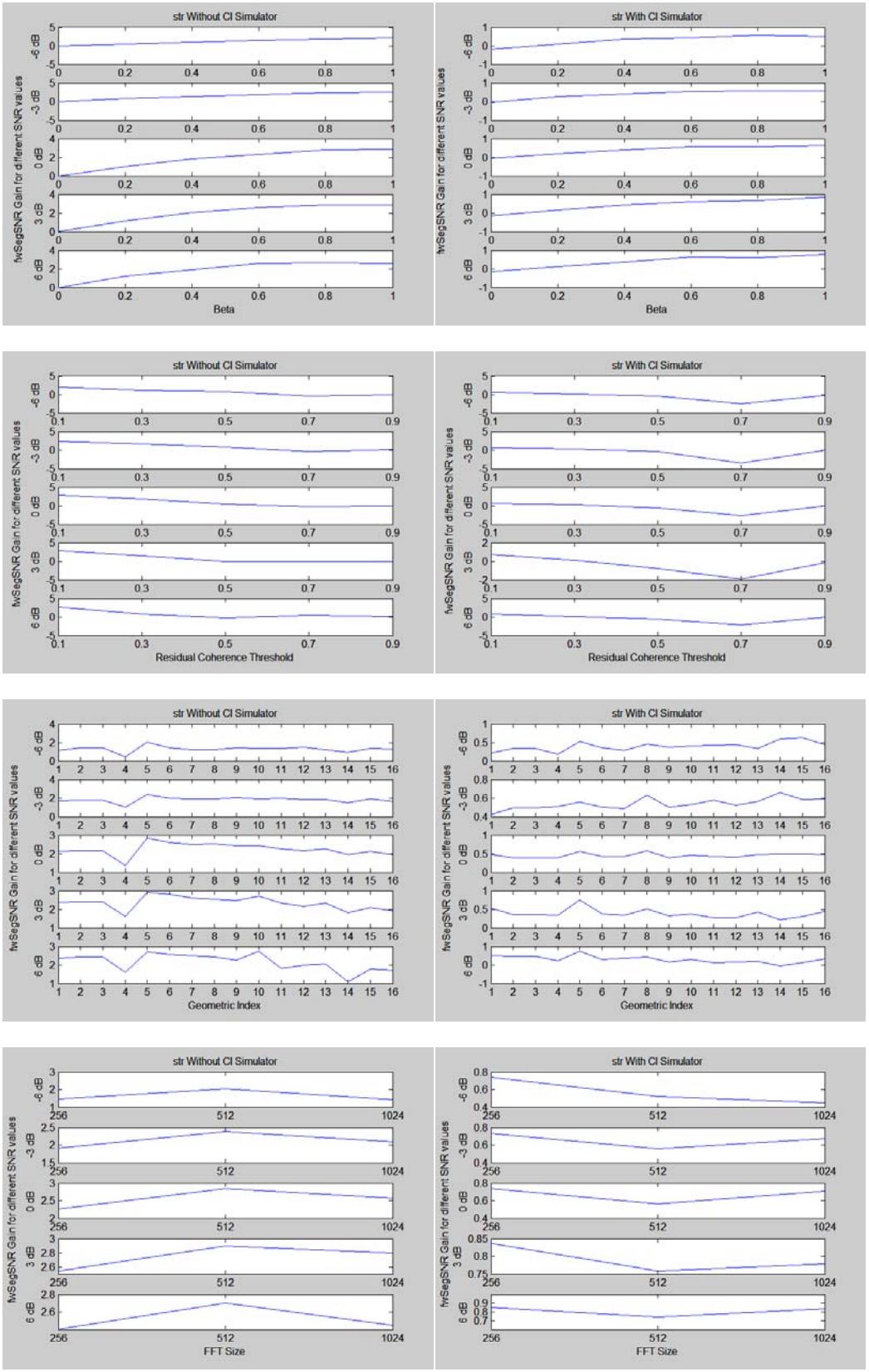
Prof. J. Vörös

B. Figures from III.C

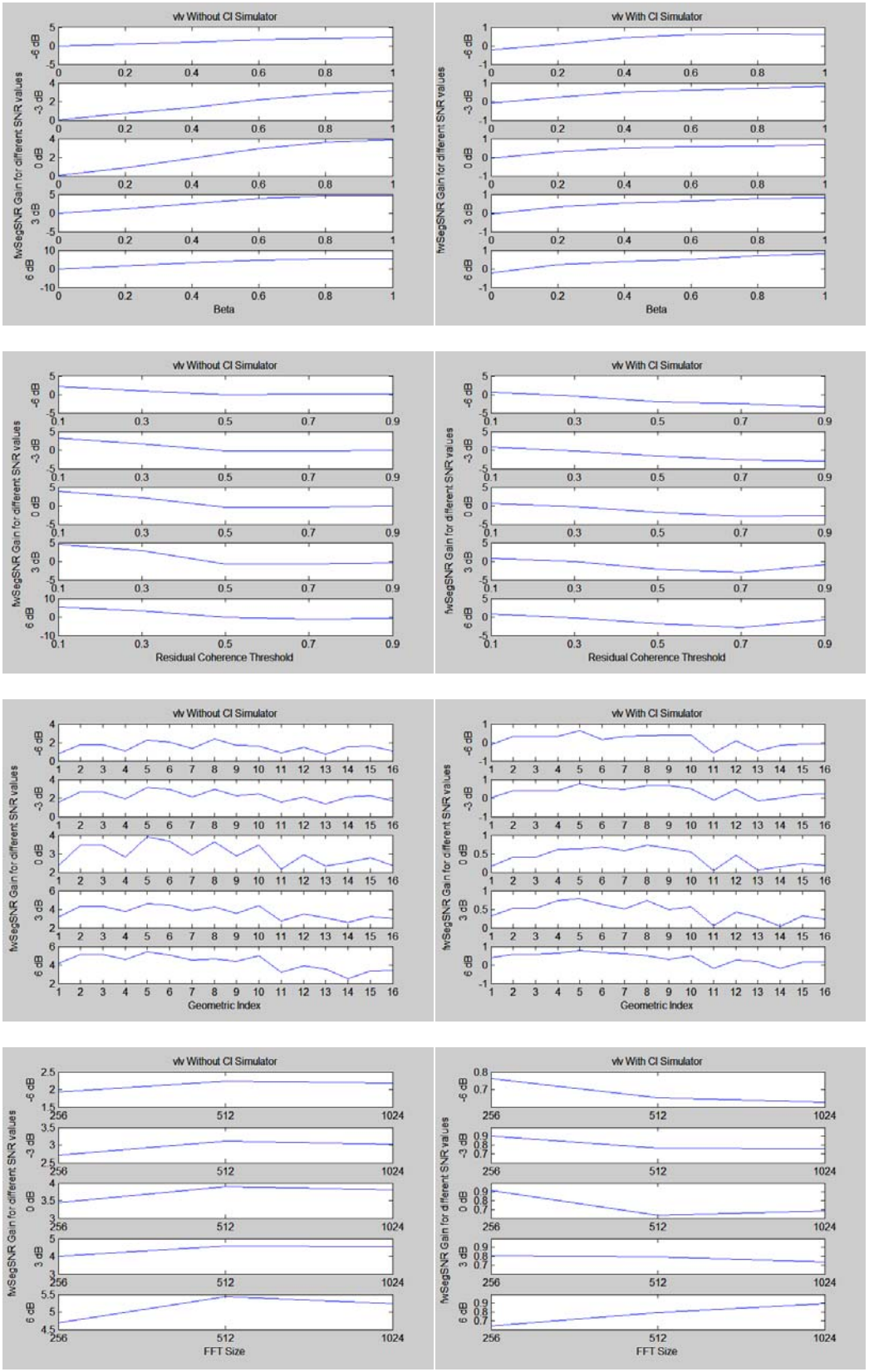
Factory Noise



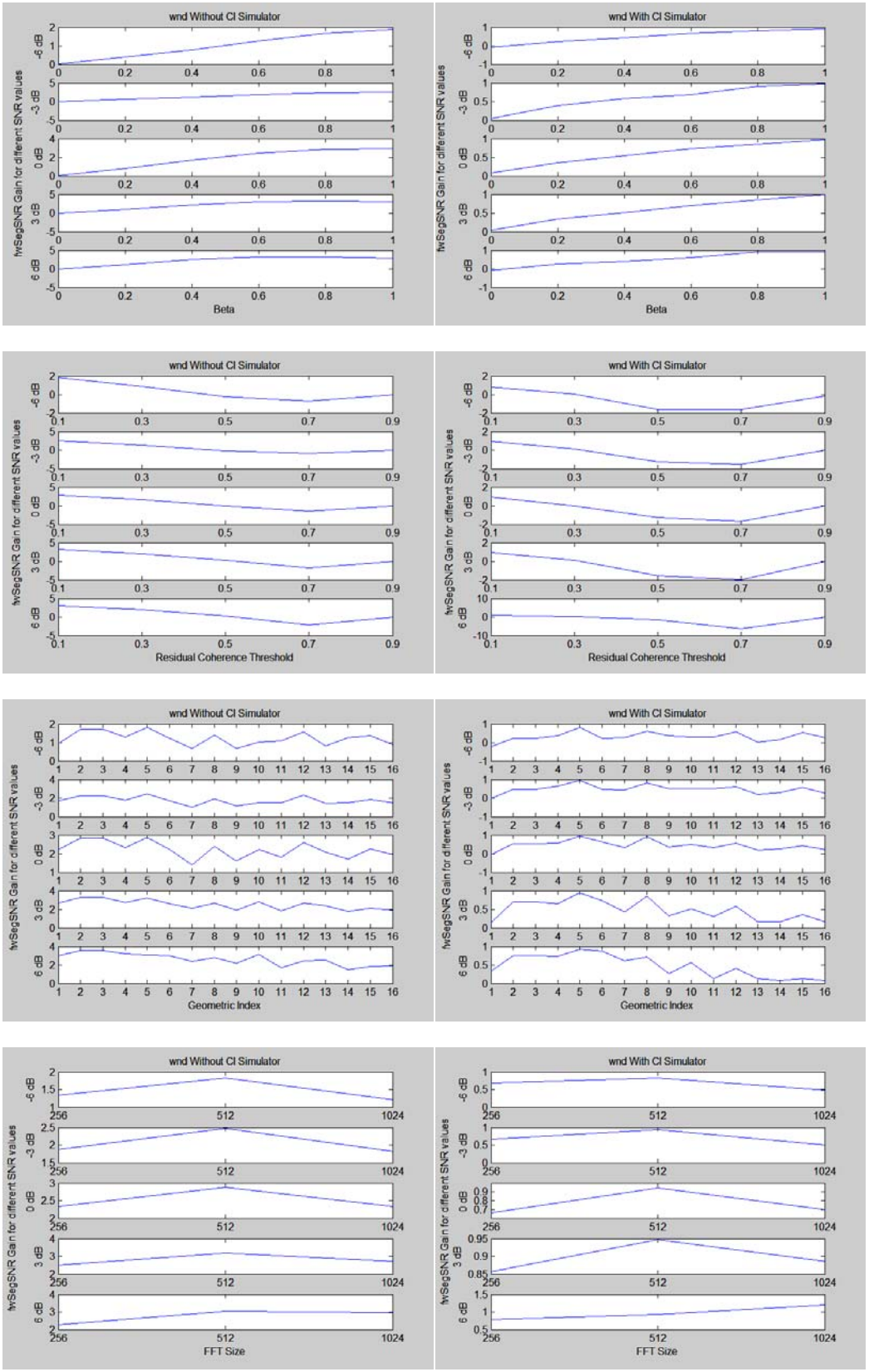
Street Noise



Volvo Car Noise



Wind Noise


wnd Without CI Simulator
wnd With CI Simulator
fwSegSNR Gain for different SNR values
-6 dB
-3 dB
0 dB
3 dB
6 dB
Beta
Residual Coherence Threshold
Geometric Index
FFT Size

C. Patient Information Document

UniversitätsSpital Zürich

Klinik für Ohren-, Nasen-, Hals- und Gesichtschirurgie

Probandinnen- / Probandeninformation für Erwachsene

TITEL DER STUDIE
Vergleich unterschiedlicher Parametersätze eines Verfahrens zur verbesserten Sprachverständlichkeit von CI-Patienten
Comparison of different parametrization sets of a dictionary-learning-based speech enhancement algorithm for CI recipients

Sehr geehrte Versuchsteilnehmerin,
Sehr geehrter Versuchsteilnehmer

1 Auswahl der Studienteilnehmer

Sie wurden für die Studie angefragt, weil die Unterdrückung von Störgeräuschen mit Cochlea-Implantaten heute noch sehr eingeschränkt ist und verbessert werden soll. Hierfür werden Sprachverständlichkeitsmessungen im Störgeräusch mit erwachsenen Cochlea-Implantat-Trägern durchgeführt. Im Falle einer beidseitigen Versorgung wird lediglich das bessere Ohr verwendet.

2 Ziel der Studie

Das Ziel dieser Studie ist es, verschiedene Arten von Störgeräuschunterdrückung miteinander zu vergleichen, indem die Sprachverständlichkeit von Cochlea-Implantat-Trägern in Lärm getestet wird. Zusätzlich nehmen normalhörende Probanden an denselben Messbedingungen, unter zusätzlicher Verwendung eines Cochlea-Implantat-Simulators (Simulation von Hörverlust und Cochlea Implantat), teil. Die Resultate werden dann miteinander verglichen. Die daraus gewonnenen Informationen sollen der Verbesserung der Sprachverständlichkeit mit Cochlea Implantaten in geräuschvollen Umgebungen dienen.

3 Allgemeine Informationen zur klinischen Studie

Die Messungen werden ausschliesslich am UniversitätsSpital Zürich im Labor für Experimentelle Audiologie durchgeführt. Um eine Vielzahl von Daten erfassen zu können, werden pro Proband innerhalb einer Sitzung neun Hörbedingungen getestet.

Diese Studie wird nach geltenden schweizerischen Gesetzen und nach international anerkannten Grundsätzen durchgeführt.

4 Freiwilligkeit der Teilnahme

Ihre Teilnahme an dieser Studie ist freiwillig. Wenn Sie auf die Teilnahme an dieser Studie verzichten, haben Sie keine Nachteile für Ihre weitere medizinische Betreuung zu erwarten. Das gleiche gilt, wenn Sie Ihre dazu gegebene Einwilligung zu einem späteren Zeitpunkt widerrufen. Diese Möglichkeit haben Sie jederzeit. Einen allfälligen Widerruf Ihrer Einwilligung bzw. den Rücktritt von der Studie müssen Sie nicht begründen. Im Falle eines Widerrufes werden die bis zu diesem Zeitpunkt erhobenen Daten weiter verwendet

5 Studienablauf

- Die Studie beinhaltet eine Testsitzung mit insgesamt neun Sprachverständlichkeitsmessungen unter verschiedenen Messbedingungen. Variiert werden die Art des Störgeräuschs sowie die Methode zur Störgeräuschreduktion im Cochlea-Implantat
- Nach jeweils drei Sprachverständlichkeitsmessungen ist eine kurze Pause von ca. 5 min eingeplant. In dieser Zeit ist eine subjektive Rückmeldung der Probanden über die Hörsituationen erwünscht
- Die Experimente finden in einem schallisolierten Raum statt
- Die akustischen Signale werden dem Probanden aus einem Lautsprecher von vorne mit einem Abstand von 1.5 m und einem maximalen Pegel für das Sprachsignal von 75 dB und für das Störgeräusch von 65 dB SPL präsentiert, was einer Lautstärke von lauter Sprache in einer Cafeteria-Situation entspricht

UniversitätsSpital Zürich

Klinik für Ohren-, Nasen-, Hals- und Gesichtschirurgie

- Der Proband wiederholt mündlich, was von ihm verstanden wurde
- Dauer pro Messbedingung: ca. 5 Minuten. Dauer der Sitzung: ca. 1 Stunde

6 Pflichten des Studienteilnehmers und des Prüfers

Als Studienteilnehmer sind Sie verpflichtet, den Anweisungen Ihres Prüfers / Ihrer Prüferin zu folgen und sich an den Studienplan zu halten.

7 Nutzen für die Teilnehmer

Die in dieser Studie gewonnenen Informationen können der Verbesserung der Störgeräuschreduktion mit Cochlea-Implantaten, und infolgedessen einer verbesserten Sprachverständlichkeit in lärmigen Situationen, dienen.

8 Risiken und Unannehmlichkeiten

Es bestehen keine bekannten Risiken bei der Durchführung der Hörversuche.

Die Präsentationen des Sprachsignals bei maximal 75 dB SPL und des Störgeräuschs bei 65 dB SPL liegen weit unterhalb den Schallpegeln, welche unangenehm oder schädigend sein könnten.

9 Vertraulichkeit der Daten

In dieser Studie werden persönliche Daten von Ihnen erfasst. Diese Daten werden anonymisiert. Sie sind nur Fachleuten zur wissenschaftlichen Auswertung zugänglich. Die zuständigen und in die Studie involvierte Fachleute können im Rahmen eines sogenannten Monitorings oder Audits die Durchführung der Studie überprüfen. Diese, sowie im Rahmen von Inspektionen auch die Mitglieder der zuständigen Behörden können Einsicht in Ihre Originaldaten nehmen. Ebenso kann die zuständige Ethikkommission Einsicht in die Originaldaten nehmen. Während der ganzen Studie und bei den erwähnten Kontrollen wird die Vertraulichkeit strikt gewahrt. Ihr Name wird in keiner Weise in Rapporten oder Publikationen, die aus der Studie hervorgehen, veröffentlicht.

10 Kosten

Ihnen entstehen durch die Teilnahme an der Studie keine zusätzlichen Kosten.

11 Entschädigung für die Studienteilnehmenden

Für die Teilnahme an dieser klinischen Studie werden Ihnen die Reisekosten unter Vorlage der Originaltickets erstattet.

12 Deckung von Schäden

Das UniversitätsSpital Zürich ersetzt Ihnen Schäden, die Sie gegebenenfalls im Rahmen des klinischen Versuchs erleiden. Stellen Sie während oder nach dem klinischen Versuch gesundheitliche Probleme oder andere Schäden fest, so wenden Sie sich bitte an die untenstehende Kontaktperson. Sie wird für Sie die notwendigen Schritte einleiten.

13 Kontaktperson(en)

Bei Unklarheiten, Notfällen, unerwarteten oder unerwünschten Ereignissen, die während der Studie oder nach deren Abschluss auftreten, können Sie sich jederzeit an die untenstehende Kontaktpersonen wenden:

Prof. Dr. sc. techn. Norbert Dillier
Leiter Experimentelle Audiologie
UniversitätsSpital Zürich
ORL-Klinik
Frauenklinikstrasse 24
8091 Zürich
Tel. +41 44 255 5801
Email: Norbert.Dillier@usz.ch

D. Components Separation after Non-Linear Normalization

Let S be the speech signal, I the interferer signal and X their additive mixture, in form of coefficient matrices. The following relations hold (separately for each scale):

$$X_{norm} = \frac{X - \min}{\max - \min} = \frac{S + I - \min}{\max - \min} = \frac{S - \min + I - \min + \min}{\max - \min} = S_{norm} + I_{norm} + \frac{\min}{\max - \min} = S_{norm} + I_{norm} + A. \quad \text{(D.1)}$$

It should be noted that the constant A is added to every element of the matrices involved, therefore it would be proper to indicate by A, a matrix of size equal to X and value of min/(max-min) in all its elements.

After LARC coding on the composite dictionary, X_{norm} is factorized into

$$X_{norm} = D \times C1 = Ds \times C1_s + Di \times C1_i \quad \text{(D.2)}$$

However, from D.1, X_{norm} should also be expressed as a linear combination of atoms from the Ds comprising S_{norm}, plus a linear combination of atoms from Di comprising I_{norm}, plus A. This can be formulated as

$$X_{norm} = Ds \times C2_s + Di \times C2_i + A \quad \text{(D.3)}$$

The estimation of S_{norm}, which needs to be back normalized according to the normalization rule in order to acquire the estimated coefficients of clean speech equals

$$S_{norm} = Ds \times C2_s \quad \text{(D.4)}$$

Therefore $C2$ needs to be expressed in relation to $C1$, which is known after LARC coding. $C2_s$ is the upper half of $C2$. From D.2 and D.4 it can be derived that

$$C2 = pseudoInverseOf(D) \times (D \times C1 - A). \quad \text{(D.5)}$$