Delft University of Technology

Faculty Electrical Engineering, Mathematics and Computer Science

Speech Enhancement Report
Digital Audio and Speech Processing
2014

Sreejith Chandrasekaran	4324846
Abhishek Sen	4319850
P.S.N.Chandrasekaran	4301994

June 27, 2014

1 Introduction

This reports details the implementation a single-channel real time speech enhancement system comprising of a gain function, noise PSD estimator and a speech PSD estimator. The system is evaluated using the standards of Perceptual Evaluation of Speech Quality (PESQ).

In section 2, we discuss the different algorithms and details of the implementation; section 3 discusses how the system is incorporated in real time scenarios; we present the findings and results of evaluation in section 4; and conclude our findings in section 5.

2 Methodology

2.1 Data Interpretation

The sample audio files provided were used in the design of the system:

filename	**description**	**length (in ms)**
clean.wav	clean speech signal	36,000
noise1.wav	stationary noise signal	36,000
intersection_soundjay.wav	non-stationary noise signal	64,000

There are two variants of the noisy input speech signal (i) combination of the stationary noise and clean speech and (ii) combination of the truncated non-stationary noise and clean speech. Both the input noisy speech signals have a sampling rate of 16,000Hz. The system is evaluated against both noisy input signals and we present our findings in section 4.

The built-in MATLAB function `audioread` was used to read the individual noisy and speech .wav files which were later combined to form the input noisy speech signal.

2.2 Windowing and Transform

The input noisy signal is split into individual windows by means of a modified version of a Hanning window for compatibility with speech signals. Each window comprises of 512 samples to ensure that enhancement is accomplished in real time.

```
window =(.5 + .5*cos(2*pi*(-(length-1)/2:(length-1)/2)/length))';
```

Note: Referred from the sample code by Richard Heusdens

The windows then undergo a FFT transform to convert the signal from the time domain to the frequency domain comprising of individual frames of frequency bins of which we consider only 257 in order to take values from 0 to π. This is done by properties of DFT symmetry.

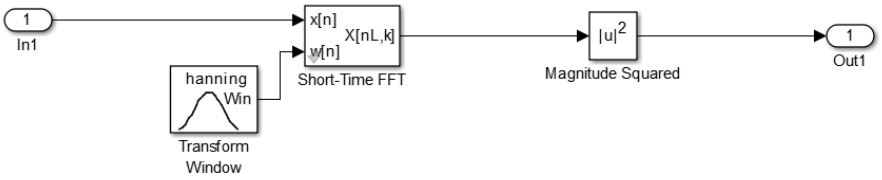

Figure 1: Simulink model for windowing and transform

2.3 Noise Estimation

The noise PSD estimate was done using two different algorithms which will be discussed in the following sections:

2.3.1 Minimum Statistics (MS)

The MS apporach assumes that the input noisy speech signal decays to a representative of the pure noise which can be used to make an estimate of the noise signal. This estimate is then employed in spectral subtraction of the noise signal and the input noisy speech signal to gain the final clean speech signal.[6][2]

The following is the algorithm employed for the MS approach:

1. Get frequency domain signal after FFT and Windowing:

$$Y(\lambda, k) = \sum_{\mu=0}^{L-1} y(\lambda R + \mu) h(\mu) e^{-j2\pi k\mu/L}$$

where
λ - time index
k - frequency bin
L - window size
R - L/2
$h(\mu)$ - window

2. Compute the recursive smoothed periodogram from Y:

$$P(\lambda, k) = \alpha P(\lambda - 1, k) + (1 - \alpha)|Y(\lambda, k)|^2$$

where α - smoothing parameter
A heuristic approach was adapted to compute the optimal value for α it was finally decided to use 0.75.

3. Compute bias correction factor by averaging over the periodogram

$$B_{min}(\lambda, k) = \frac{1}{E[P_{min}(\lambda, k)]}$$

P_{min} is calculated for successive λ values.

4. Final unbiased noise estimate

$$\widehat{\sigma_N^2}(\lambda, k) = B_{min}(\lambda, k) P_{min}(\lambda, k)$$

The following figure is the implementation of the algorithm in Simulink:

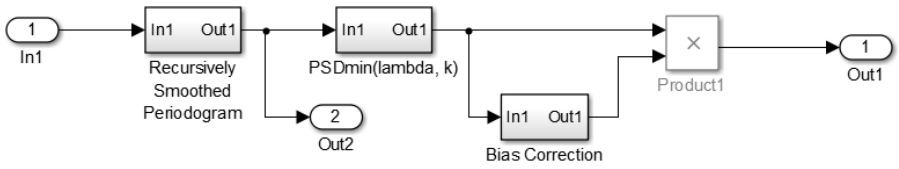

Figure 2: Simulink model for Minimum Statistics

2.3.2 Speech Presence Probability(SPP)

In SPP, there are two hypotheses - (i)$H_{1,k}(l)$ indicates that speech is present in frequency bin k at time segment l while $H_{0,k}(l)$ indicates speech absence. After computing the time frequency point where speech is absent, we use this information to update the noise power spectral density estimate.

For our purposes, we modified the original algorithm to make use of a soft SPP rather than a hard SPP that uses VAD so that the noise estimator is unbiased. This also ensures a low tracking delay without a bias trade-off.[3][4]

The following is the algorithm employed for the SPP approach:

1. Through recursive smoothing of the noise periodogram, the noise PSD is computed as:

$$\widehat{\sigma_N^2}(l) = \alpha_{pow} \widehat{\sigma_N^2}(l-1) + (1 - \alpha_{pow}) E(|N|^2 | y(l))$$

where $\alpha_{pow} = 0.8$ is computed through heuristics pre-defined

2. Using the noise PSD estimate, we compute the a-posteriori SPP under the assumption that $P(H_1) = P(H_0)$.

$$P(H_1|y) = \left(1 + \frac{P(H_0)}{P(H_1)}(1 + \xi_{H_1}) e^{-\frac{|y|^2}{\widehat{\sigma_N^2}} \frac{\xi_{H_1}}{1+\xi_{H_1}}}\right)$$

the noise PSD is considered for the previous frame $\widehat{\sigma_N^2} = \widehat{\sigma_N^2}(l-1)$

3. To avoid stagnation of a-posterioir SPP to one, we recursively smooth $P(H_1|y)$

$$\bar{P}(l) = 0.9\bar{P}(l-1) + 0.1P(H_1|y(l))$$

4. Based upon the magnitude of $\bar{P}(l)$, we compute the a-posteriori estimate

$$P(H_1|y(l) \leftarrow \begin{cases} min(0.99, P(H_1|y(l)) & \bar{P}(l) > 0.99 \\ P(H_1|y(l)) & else \end{cases}$$

5. The final unbiased estimator is then computed

$$E(|N|^2|y) = P(H_0|y)|y|^2 + P(H_1|y)\widehat{\sigma_N^2}$$

2.4 Speech Estimation

It is observed that speech is more non-stationary than noise and a simple estimation through recursive smoothing is not representative of the speech signal. Therefore, we make use of adaptive smoothing techniques to retain the original speech properties.

The algorithm we adopt is the modified Decision Directed approach that considers the time-correlation between successive speech spectral components. The estimator is computed using a time-varying frequency dependent weighing factor.[1]

The following is the algorithm employed for the DD approach:

1. Initialize parameters lower bound for a-priori SNR $\xi_{min} = -25dB$ and smoothing parameter $\alpha = 0.9$. Compute the "propagation step" for successive frames

$$\hat{\xi}_{l|l-1} = max\left\{(1-\alpha)\hat{\xi}_{l-1|l-1} + \alpha\frac{\hat{A}_{l-1}^2}{\lambda_{D_{l-1}}}, \xi_{min}\right\}$$

2. Compute the estimate of the noise spectral variance

$$\hat{\lambda}_{x_l|l-1} = max\left\{(1-\alpha)\hat{\lambda}_{x_{l-1}} + \alpha\hat{A}_{l-1}^2, \lambda_{min}\right\}$$

where λ_{min} is the minimum variance observed when there is only noise in the input speech signal i.e. input speech signal decays to noise.

3. Update the a-priori SNR $\xi_{l|l'}$ and a-posteriori SNR γ_l

$$\xi_{l|l'} \triangleq \frac{\lambda_{x_l|l'}}{\lambda_{D_l}}; \gamma_l \triangleq \frac{|Y_l|^2}{\lambda_{D_l}}$$

4. Compute the gain with the a-posteriori SNR γ_l represented as SNR_k

$$H_k = \frac{P_{SS,k}/P_{NN,k}}{P_{SS,k}/P_{NN,k}} = \frac{SNR_k}{SNR_k + 1}$$

5. Final estimator is computed using the gain

$$\hat{S}_k = H_k \cdot Y_k$$

2.5 Gain Calculation

The gain is computed through the spectral subtraction of the input noisy signal and the computed estimator

$$g = \left(1 - \frac{\sigma_N^2}{|y|^2}\right)^{0.1}$$

A heuristic approach was adopted to compute the power 0.1 for the gain. Finally the enhanced speech signal is computed in the following manner

$$\hat{s} = gy$$

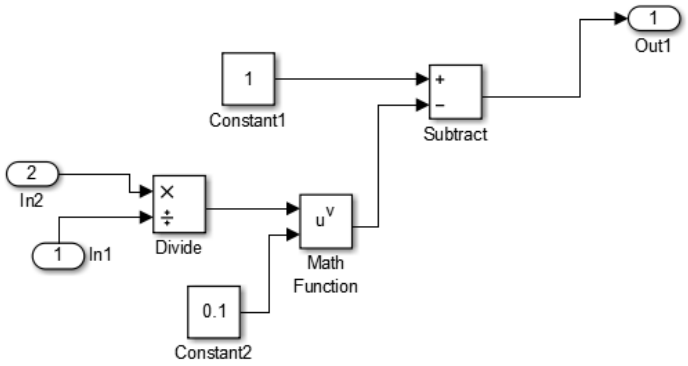

Figure 3: Simulink model for gain calculation

A further optimization to the gain function would be to fine-tune the parameters across all the various algorithms that we tried for noise and speech power estimation. What we found is that gain value performed the best for non-stationary noisy signal for SPP and DD algorithms.

2.6 Inverse FFT and Overlap-Add

The enhanced input signal is first overlapped and added to form a 512 window from the original 257 frequency bins through a hanning window finally this windowed signal is then converted from frequency domain to the time domain through Inverse FFT to form the final audible enhanced output.

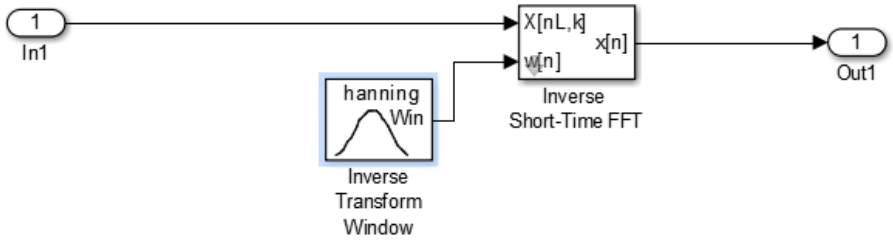

Figure 4: Simulink model for Inverse FFT and Overlap Add

3 Real Time

Our real-time tests were quite rudimentary and not working correctly. Our initial approach was to do the entire project in Simulink but we found this quite difficult. Our initial models for both offline and real-time processing were developed in Simulink for the MS algorithm. Our real-time model for the MS algorithm does not have echo cancellation and this causes any input noise from the laptop microphone to be amplified indefinitely. A future extension to this part of the project would be to perform an accurate real-time estimation of the noise and speech power parameters in order to play out the audio with echo cancellation and low tracking delay.

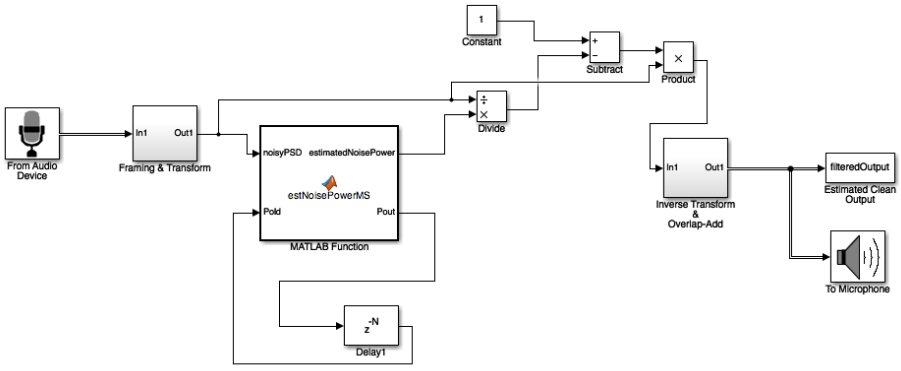

Figure 5: Simulink model for real time implementation

4 Evaluation

Evaluation is accomplished using the standards established by PESQ. The criteria for evaluation is the Mean Opinion Score (MOS). The scores ranges from 1 to 5 where 1 is very bad and 5 very good quality. A visual representation would not be reliable since perception differs for individuals, therefore we adopted the standard PESQ scores for evaluation.[5]

The following tabulation indicates the MOS values for the algorithms discussed:

Algorithm			**Filename**	**MOS**
Noisy Input	-	Stationary	noisy1.wav	1.159
		Non-Stationary	noisy2.wav	1.606
Noise Estimator	MS	Stationary	output11.wav	1.378
		Non-Stationary	output12.wav	1.458
	SPP	Stationary	output21.wav	1.451
		Non-Stationary	output22.wav	1.981
Speech Estimator	DD	Stationary	output31.wav	1.748
		Non-Stationary	output32.wav	1.928

From the above tabulation, we observe that SPP noise estimation approach performs best in the non-stationary noise scenario whereas the DD speech estimation approach performs best in the stationary noise scenario.

5 Conclusions and Future Work

In this paper, we presented the design and implementation of hearing aid functionality using MATLAB and SIMULINK. It contains window and DFT computation, noise/speech PSD estimators and gain functions. Three popular methods for speech enhancement - Minimum Statistics (MS), Speech Presence Probability (SPP) and Decision Direct(DD) methods are presented here. As can be seen from the evaluation section, SPP performs better compared to the other two methods considering the overall performance. But there is an increased computational complexity with the SPP and DD methods compared to MS. This supports the choice of MS method for real-time processing.

For future enhancements, it is required to have a better gain function. Heuristically we came up with one final gain function but the experiments showed that our gain function could be improved a lot. Also it is good to use additional FFT bins to ensure that the gain is optimised for each small frequency band - thereby getting better frequency resolution. We have already done a few experiments with the alpha and a priori values but it is good to experiment more with these values. With real-time speech processing, echo cancellation is mandatory which is not present in the current implementation. This will make sure that the hearing aid algorithm can be used with a user product as a future enhancement. Using two microphones instead of one (with beam-forming)

in our case with optimised algorithms will improve the functionality to a large extent.

References

[1] I. Cohen. On The Decision-Directed Estimation Approach of Ephraim and Malah. *IEEE*, 2004.

[2] A. R. Fukane and S. L. Sahare. Noise estimation Algorithms for Speech Enhancement in highly non-stationary Environments.

[3] T. Gerkmann and R. C. Hendriks. Unbiased MMSE-Based Noise Power Estimation with Low Complexity and Low Tracking Delay. *IEEE Transactions on Audio, Speech Language Processing*, 20(4):1383–1393, 2012.

[4] R. C. Hendriks, R. Heusdens, and J. Jensen. MMSE based noise PSD tracking with low complexity. In *ICASSP*, pages 4266–4269. IEEE, 2010.

[5] ITU-T. Perceptual evaluation of speech quality (PESQ). *ITU-T Recommendation P.862*, 2001.

[6] R. Martin. Noise power spectral density estimation based on optimal smoothing and minimum statistics. *IEEE Transactions on Speech and Audio Processing*, 9(5):504–512, 2001.