Reduction of Additive Noise in the Digital Processing of Speech

Final Report

Avner Halevy

Department of Mathematics

University of Maryland, College Park

ahalevy at math.umd.edu

Dr. Radu Balan

Department of Mathematics

Center for Scientific Computation and Mathematical Modeling (CSCAMM)

University of Maryland, College Park

rvbalan at math.umd.edu

Abstract

This project implements two standard algorithms for reducing additive white noise in the processing of speech signals. These are known as “spectral subtraction” and “iterative Wiener filtering”. The performance of the algorithms is evaluated and compared using TIMIT, a database of phonetically rich sentences, widely used in the industry in the development of speech processing algorithms. Objective measures are used to evaluate the quality of processed speech.

Background

The need to enhance speech signals arises in many situations. Most commonly the signal originates from a noisy environment, or is degraded by noise over a communication channel. Speech enhancement algorithms can be used to enhance both quality and intelligibility of speech signals, thus making communication more effective and reducing listener fatigue. The precise goals of speech enhancement algorithms depend on the specific application, and the specific type of noise involved, as well as its statistical relation to the clean signal. The main challenge in designing effective speech enhancement algorithms is reducing noise without introducing perceptible distortion to the speech signal.

This project focuses on the reduction of additive white Gaussian noise which is uncorrelated with the clean speech signal. We are assuming that $y(n)$, the noisy signal, is composed of the clean speech signal $x(n)$, and the noise $d(n)$, i.e. $y(n) = x(n) + d(n)$.

Analysis and Synthesis

Since speech signals are highly non stationary, the enhancement algorithms are applied one frame at a time. A short time Fourier transform (STFT) is used to analyze the signal, and once modifications have been made, the inverse transform is used to synthesize the enhanced signal. The STFT is defined as follows:

$$X_m(\omega) = \sum_{n=-\infty}^{\infty} w(m-n)x(n)e^{-j\omega n} \qquad (1)$$

Where *x* is the signal and *w* is the analysis window. To achieve perfect reconstruction we must be sure to sample the STFT at high enough rates in both time and frequency. From the definition above we see that $X_m(\omega)$ can be thought of as the output of a linear filter with impulse response *w*. Typically, *w* will have the properties of a low pass filter with bandwidth *B*. Thus the sequence $X_m(\omega)$ will have bandwidth *B*, so according to the sampling theorem it must be sampled at a rate of 2*B* to avoid aliasing. We will be using the Hamming window, given by

$$w(n) = 0.54 - 0.46\cos\left(2\pi\frac{n}{L}\right)\ n = 0,1,\dots,L \quad (2)$$

which can be shown to have bandwidth $B = 2F/L$, where F is the sampling rate of the signal *x* and *L* is the length of the window. Thus $X_m(\omega)$ must be evaluated $2B = 4F/L$ times per second, or every $L/4$ samples. Thus we can rewrite (1) as

$$X_m(\omega) = \sum_{n=-\infty}^{\infty} w(mR-n)x(n)e^{-j\omega n} \qquad (3)$$

where $R = L/4$ is the hop size in samples. As for frequency sampling, we first note that since $X_m(\omega)$ is periodic with period 2π, it is only necessary to sample over an interval of 2π. Next we note that by (1) we can think of the STFT as the Discrete Time Fourier Transform (DTFT) of the sequence $w(m-n)x(n)$ for a fixed *m*. Since the window is time-limited with length *L*, according to the sampling theorem it suffices to sample $X_m(\omega)$ in frequency every $2\pi/L$. Thus we may replace (3) by

$$X_m(\omega_k) = \sum_{n=-\infty}^{\infty} w(mR-n)x(n)e^{-j\omega_k n} \quad \omega_k = \frac{2\pi k}{L},\ k = 0,1,\dots,L-1 \quad (4)$$

In practice we evaluate $X_m(\omega)$ at *N* uniformly spaced frequencies around the unit circle, where *N* is the first power of 2 greater than *L*.

To actually reconstruct the enhanced signal, we first note that for a band-limited window, and in particular for the Hamming window, we have

$$\sum_{m=-\infty}^{\infty} w(mR-n) = C, \forall n \quad (5)$$

where $C = W(0)/R$, and $W(\omega)$ is the DTFT of w. Without loss of generality we may assume $C = 1$. This is known as the constant Overlap Add (OLA) constraint, and it allows us to obtain a simple scheme for perfect reconstruction, as follows. By (3) we have

$$\sum_{m=-\infty}^{\infty} X_m(\omega) = \sum_{m=-\infty}^{\infty}\sum_{n=-\infty}^{\infty} x(n)w(mR-n)e^{-j\omega n}$$

$$= \sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n}\sum_{m=-\infty}^{\infty} w(mR-n) = C\sum_{n=-\infty}^{\infty} x(n)e^{-j\omega n} = X(\omega) \quad (6)$$

where *X* is the DTFT of *x*. Thus, we may recover *x* by computing the inverse DTFT of this sum. However, as the discussion above shows, it suffices to use the sampled values in (4) and replace the inverse DTFT with the inverse DFT:

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1}\sum_{m=-\infty}^{\infty} X_m(\omega_k)\, e^{j\omega_k n} = \sum_{m=-\infty}^{\infty}\frac{1}{N}\sum_{k=0}^{N-1} X_m(\omega_k)e^{j\omega_k n} = \sum_{m=-\infty}^{\infty} x_m(n) \quad (7)$$

where $x_m(n) = w(m-n)x(n)$. This result serves as the basis of the OLA method, which is used in the implementation . If we denote by H_m the frequency response of the filter applied to frame *m*, and use a fast Fourier transform (FFT) long enough to accommodate spectrum modifications, we may describe the input (*x*)-output(*y*) chain explicitly as follows:

$$y = \sum_{m=-\infty}^{\infty} FFT^{-1}\{H_m FFT(x_m)\} \quad (8)$$

In particular, if we apply no processing ($H_m \equiv 1$) we get $y = x$.

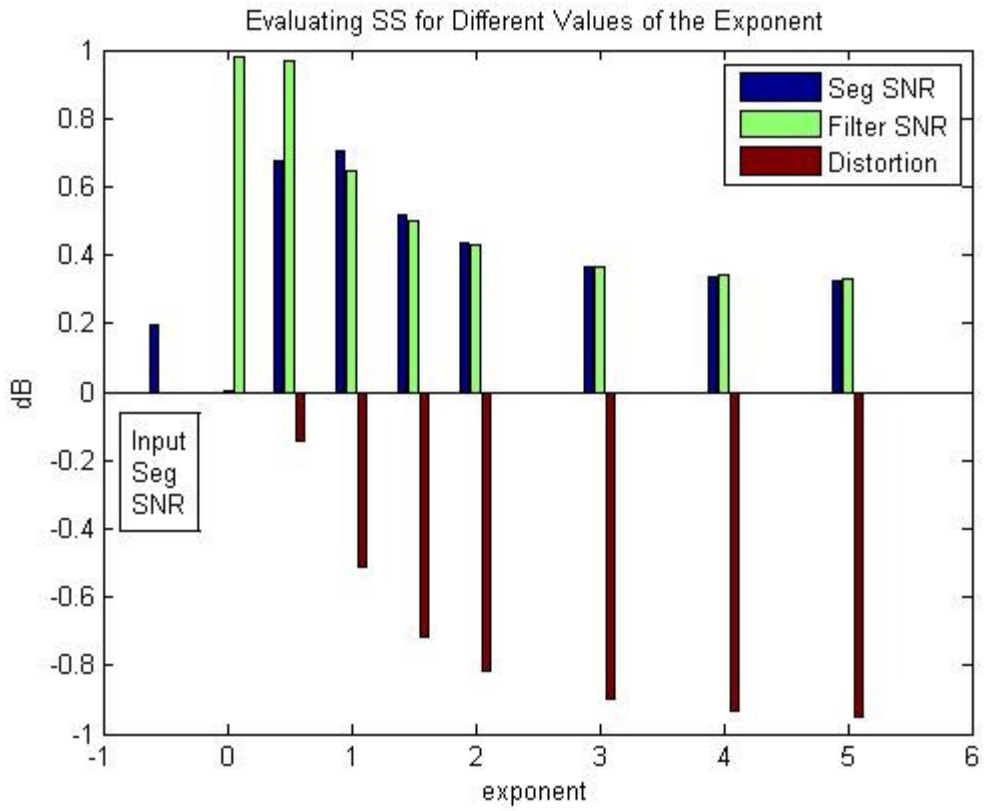
Spectral Subtraction

We now describe the first algorithm implemented, spectral subtraction. We begin by estimating the noise spectrum. The first five frames of the noisy signal are assumed to be noise only. The noise power spectrum is averaged over these frames to obtain an estimate which is used for the rest of the algorithm. The heart of the algorithm consists of subtracting the estimate of the noise magnitude from the magnitude of the noisy signal spectrum, to recover (an estimate) of the magnitude of the clean signal spectrum. Due to fluctuations in the noise spectrum, this subtraction may result in negative values, in which case the most basic approach is to set these values to zero. The magnitude estimate is then combined with the noisy phase, an approximation (of the clean phase) which has been shown to be good enough for practical purposes. Symbolically, the algorithm is described as follows:

$$y(n) = x(n) + d(n) \quad (1)$$
$$Y(\omega) = X(\omega) + D(\omega) \quad (2)$$
$$Y(\omega) = |Y(\omega)|e^{i\varphi(\omega)} \quad (3)$$
$$\left|\hat{X}(\omega)\right| = \left(|Y(\omega)|^p - \left|\widehat{D}(\omega)\right|^p\right)^{\frac{1}{p}} \quad (4)$$
$$\hat{X}(\omega) = \left|\hat{X}(\omega)\right|e^{i\varphi(\omega)} \quad (5)$$
$$\hat{x}(n) = inverse\ Fourier\left\{\hat{X}(\omega)\right\} \quad (6)$$

Here $\widehat{D}(\omega)$ is the estimate of the magnitude of the noise spectrum at frequency ω, computed in practice by averaging the magnitude of the fast Fourier transform of the noise signal at frequency ω over the first five frames.

There is freedom in choosing the exponent p above, and we experimented with various values in the range .1 – 5. The results are summarized below:



These results suggest that as p increases, the filter becomes less aggressive, and thus less distortion is introduced. Conversely, as p decreases to zero, the clean signal is progressively annihilated. Subjective evaluation (listening) suggests that values in the range .5 - 2 offer the best compromise between a desirable increase in SNR and an undesirable increase in distortion.

To gain further insight into the algorithm, as well as access to objective quality measures described below, we may view the subtraction as a "filtering" process:

$$\left|\hat{X}(\omega)\right| = \left(|Y(\omega)|^p - \left|\widehat{D}(\omega)\right|^p\right)^{1/p} \quad (7)$$

$$\left|\hat{X}(\omega)\right| = \left[1 - \left(\frac{\left|\widehat{D}(\omega)\right|}{|Y(\omega)|}\right)^p\right]^{1/p} |Y(\omega)| \quad (8)$$

$$\left|\hat{X}(\omega)\right| = H(\omega)|Y(\omega)| \quad (9)$$

$$H(\omega) = \left[1 - \left(\left(\frac{|Y(\omega)|}{\left|\widehat{D}(\omega)\right|}\right)^{-1}\right)^p\right]^{1/p} \quad (10)$$

Using this view we see that the lower the SNR, the more the noisy magnitude is attenuated.

The main challenge in spectral subtraction is dealing with the so called "musical noise" artifacts which result from the flooring of negative components. These artifacts were blatantly evident in our experiments. One approach is to use an over subtraction coefficient for the noise spectrum. We experimented with this approach (though only in a nonsystematic way), and the results were not convincing.

Iterative Wiener Filtering

We now describe the second algorithm implemented, iterative Wiener filtering. We begin once again with $y(n) = x(n) + d(n)$ and we first look for a filter with impulse response *h* to obtain an estimate of the clean signal $\hat{x}(n) = \sum_{k=-\infty}^{\infty} h(k)\, y(n-k), -\infty < n < \infty$. If we define the error by $e(n) = x(n) - \hat{x}(n)$ and seek to minimize $E[e^2(n)]$ we obtain, in frequency domain, the well-known Wiener filter: $H(\omega) = \frac{P_{xx}(\omega)}{P_{xx}(\omega) + P_{dd}(\omega)}$. $P_{dd}(\omega)$ is the power spectrum of the noise, estimated in practice by using the square of the magnitude spectrum (i.e. $P_{dd}(\omega) = \left|\widehat{D}(\omega)\right|^2$, where $\widehat{D}(\omega)$ was given under "Spectral Subtraction"). If we define the *a priori* SNR at frequency ω by $\xi(\omega) = \frac{P_{xx}(\omega)}{P_{dd}(\omega)}$ we see that $H(\omega) = \frac{\xi(\omega)}{1+\xi(\omega)}$. From this it is clear that the filter is approximately 0 at low SNR and approximately 1 at high SNR.

We can see above that the Wiener filter requires access to the power spectrum of the clean signal, which we do not have. This brings us to the iterative scheme, in which at each iteration we arrive at a new estimate of the filter and a corresponding estimate of the clean signal spectrum. The algorithm was first proposed by Lim and Oppenheim in [5]. The present approach relies on the assumption that the clean speech was generated by an all-pole model given in the *z* domain by $V(z) = \frac{g}{1 - \sum_{k=1}^{p} a_k z^{-k}}$, where *g* is the gain of the system, $\{a_k\}$ are the all-pole coefficients, and *p* is their number. In time domain the clean signal is now given by $x(n) = \sum_{k=1}^{p} a_k x(n-k) + g w(n)$, where $w(n)$ is the input excitation, assumed to be white Gaussian with zero mean and unit variance. Thus, rather than try to estimate the clean signal, we may try to estimate the all-pole coefficients. This approach gives rise to the following linear iterative algorithm, based on a maximum *a posteriori* (MAP) estimate of the clean signal given the noisy signal and the all-pole coefficients:

Step 0: initialize the signal estimate with the noisy signal $x_0 = y$.

For i = 0,1,2,…

Step 1: Given x_i estimate the all-pole coefficients $\{\mathbf{a}_i(k)\}$ using linear prediction.

Step 2: Using $\{\mathbf{a}_i(k)\}$ estimate the gain term g^2.

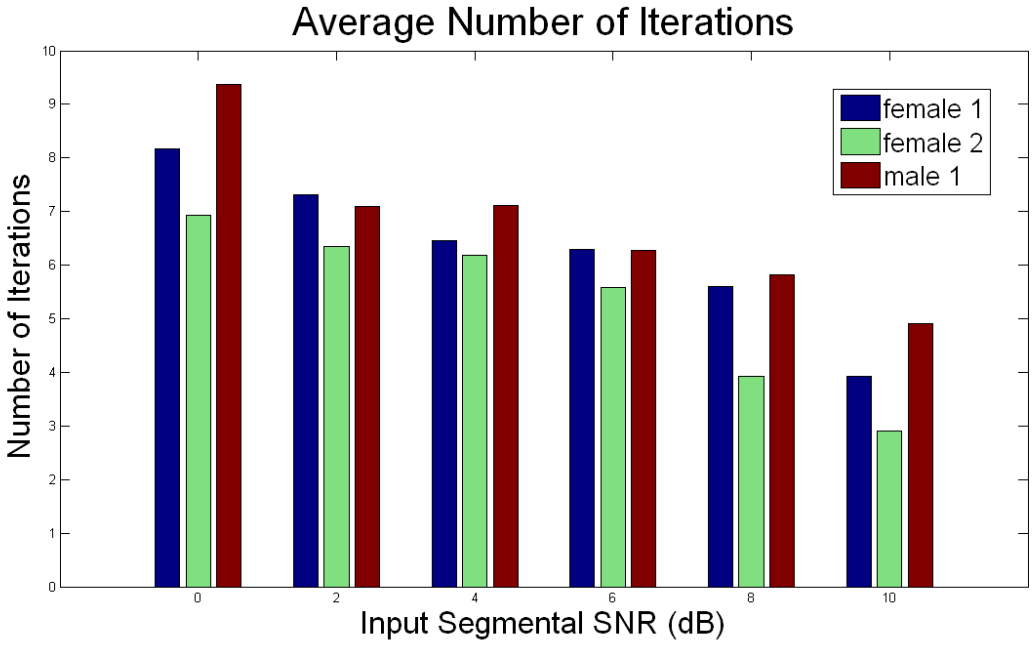
Step 3: Compute the short term power spectrum $P_{x_i x_i}(\omega) = \frac{g^2}{\left|1-\sum_{k=1}^{p} a_i(k)e^{-jk\omega}\right|^2}$

Step 4: Compute the Wiener filter $H_i(\omega) = \frac{P_{x_i x_i}(\omega)}{P_{x_i x_i}(\omega)+P_{dd}(\omega)}$

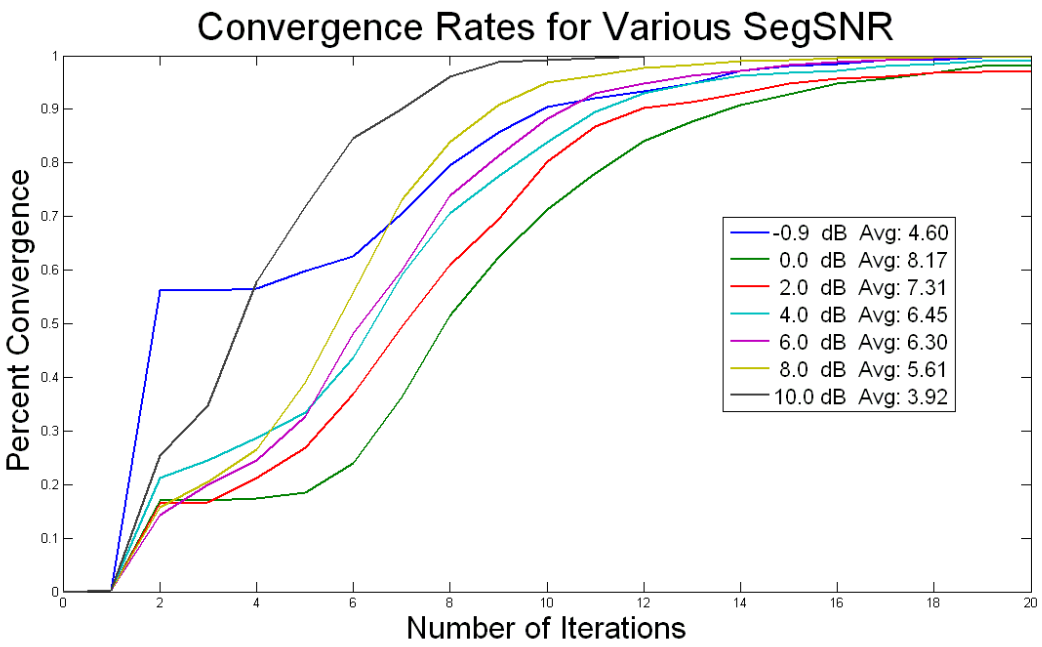
Step 5: Estimate the spectrum of the enhanced signal $X_{i+1}(\omega) = H_i(\omega)Y(\omega)$

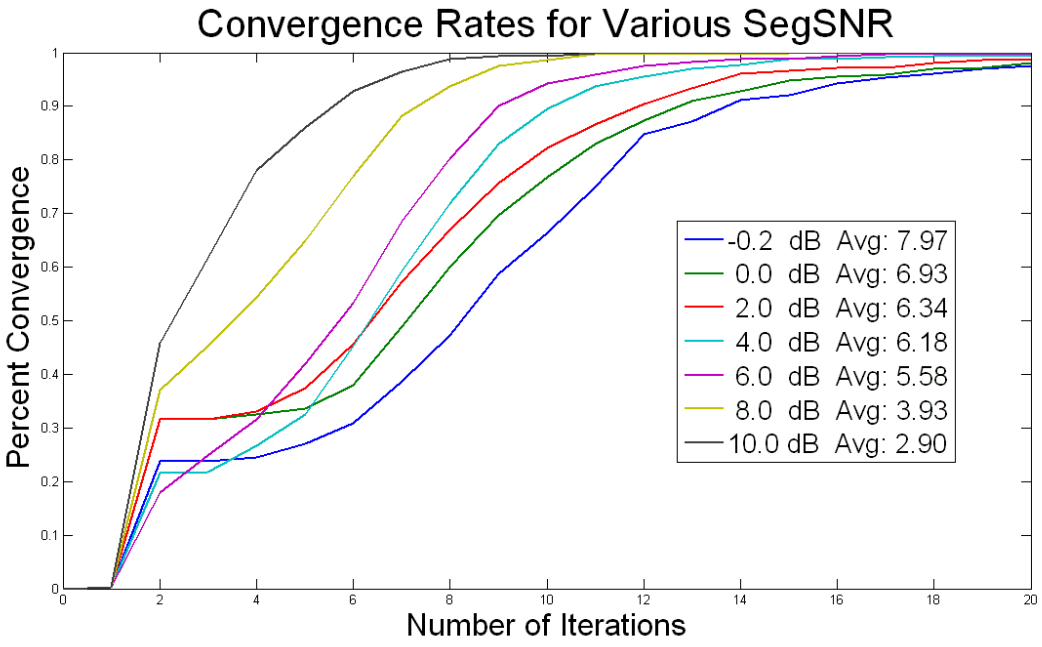
Step 6: Compute the inverse Fourier transform of $X_{i+1}(\omega)$ to obtain $x_{i+1}(n)$.

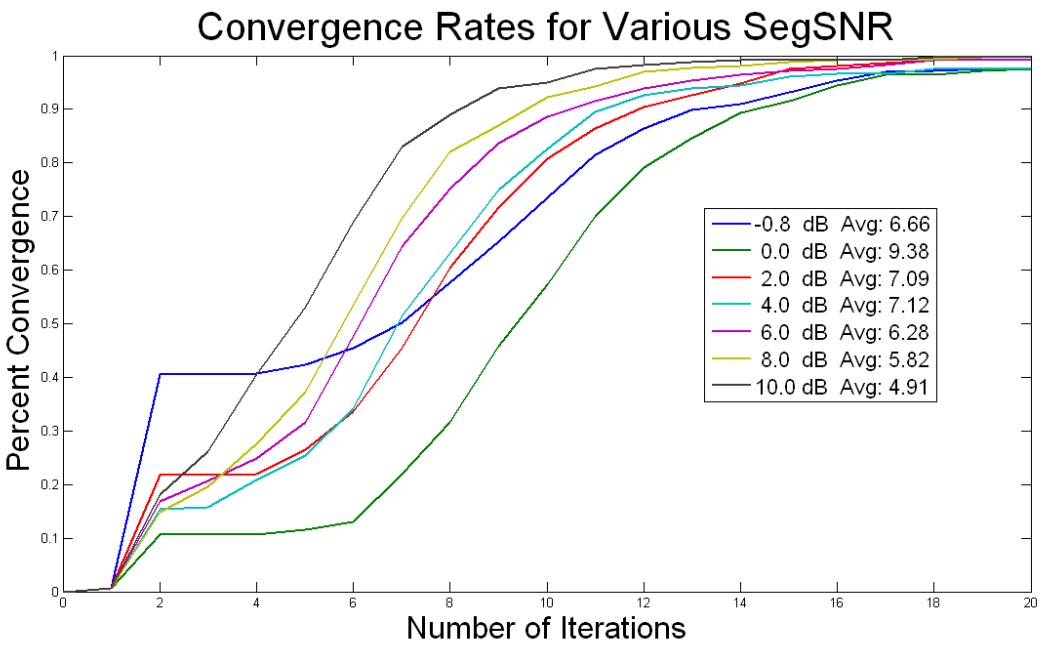
No optimal convergence criterion has been found. Spurious spectral peaks may be generated if the algorithm is run for too long, and the optimal number of iterations may be different for different signals. However, convergence is usually quick in practice, with an average of 4-9 iterations for a one percent tolerance. More precisely, below we see the average number of iterations for three different test signals, for six different input Segmental SNRs (for a discussion of this measure see "Validation and Testing" below). The figure shows that convergence is faster with increasing Segmental SNR.



A closer view of the behavior of the algorithm for these signals is given in the following three figures (in the order: female sentence 1, female sentence 2, and male sentence 1).


Convergence Rates for Various SegSNR
Percent Convergence
Number of Iterations
-0.9 dB Avg: 4.60
0.0 dB Avg: 8.17
2.0 dB Avg: 7.31
4.0 dB Avg: 6.45
6.0 dB Avg: 6.30
8.0 dB Avg: 5.61
10.0 dB Avg: 3.92


Convergence Rates for Various SegSNR
Percent Convergence
Number of Iterations
-0.2 dB Avg: 7.97
0.0 dB Avg: 6.93
2.0 dB Avg: 6.34
4.0 dB Avg: 6.18
6.0 dB Avg: 5.58
8.0 dB Avg: 3.93
10.0 dB Avg: 2.90



Validation and Testing

Validation of both implementations was done by setting the estimate of the noise spectrum magnitude equal to zero, in which case the output (enhanced) signal was identical to the input (noisy) signal, as desired. Stability of the implementations was verified using small non-zero noise.

Testing of the algorithms was performed using clean signals from TIMIT, a database of phonetically rich sentences spoken in 8 different dialects, widely used in the industry in the development of speech processing algorithms. White Gaussian noise was added using MATLAB's **randn** function.

Evaluation of the algorithm was mostly limited to the use of objective measures. The first measure, called *segmental SNR*, considers the energy of the clean signal as compared with the energy of the error in estimation, and is defined as

$$SNR_{seg} = \frac{10}{M}\sum_{m=0}^{M-1}\log_{10}\frac{\sum_{n=Nm}^{Nm+N-1}x^2(n)}{\sum_{n=Nm}^{Nm+N-1}(x(n)-\hat{x}(n))^2}$$

where $x(n)$ is the clean signal, $\hat{x}(n)$ is the enhanced signal, N is the frame length, and M is the number of frames. In this, as well as in the SNR measure described next, lower and upper thresholds are used to prevent the final measure from being distorted by atypical frames: if the frame value computed is negative, it is set to 0, and if it is higher than 35, it is set to 35.

Two other measures use the "filter" H discussed above. Once again an average over all frames is computed, but the quantity computed in each frame depends on H. If we denote by $\tilde{x}$ the time domain result of applying H to x (the clean signal) and by $\tilde{d}$ the analogous quantity for the noise signal d, then the first measure, which we call *Filter SNR*, considers the energy of $\tilde{x}$ compared with the energy of $\tilde{d}$ and is defined as

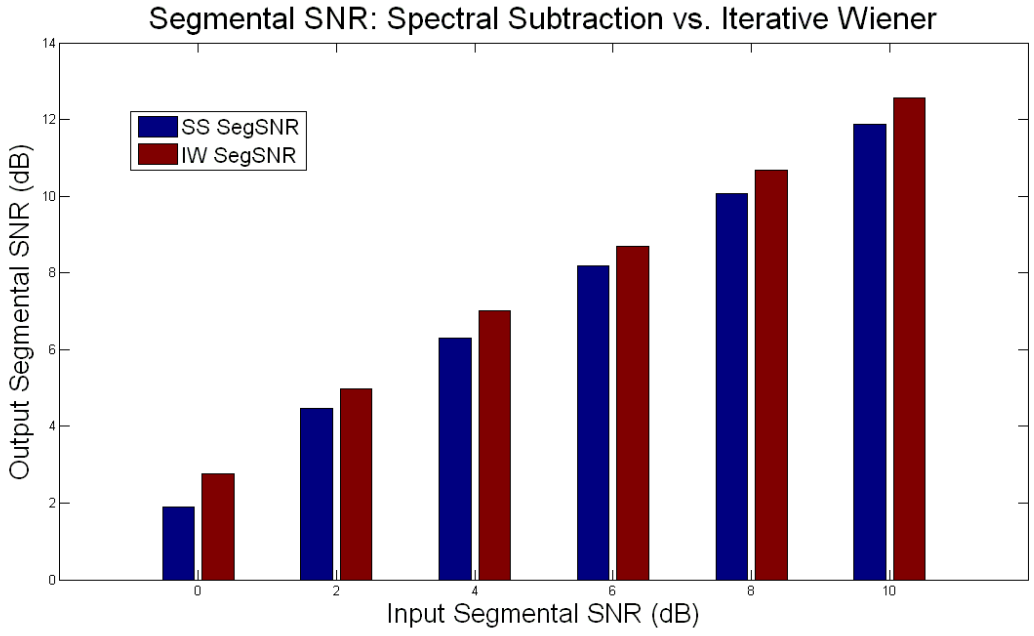
$$Filter\ SNR = \frac{10}{M}\sum_{m=0}^{M-1}\log_{10}\frac{\sum_{n=Nm}^{Nm+N-1}\tilde{x}^2(n)}{\sum_{n=Nm}^{Nm+N-1}\tilde{d}^2(n)}$$

The second measure, which we call *Distortion*, considers the energy of $x-\tilde{x}$ compared with the energy of x, and is defined as

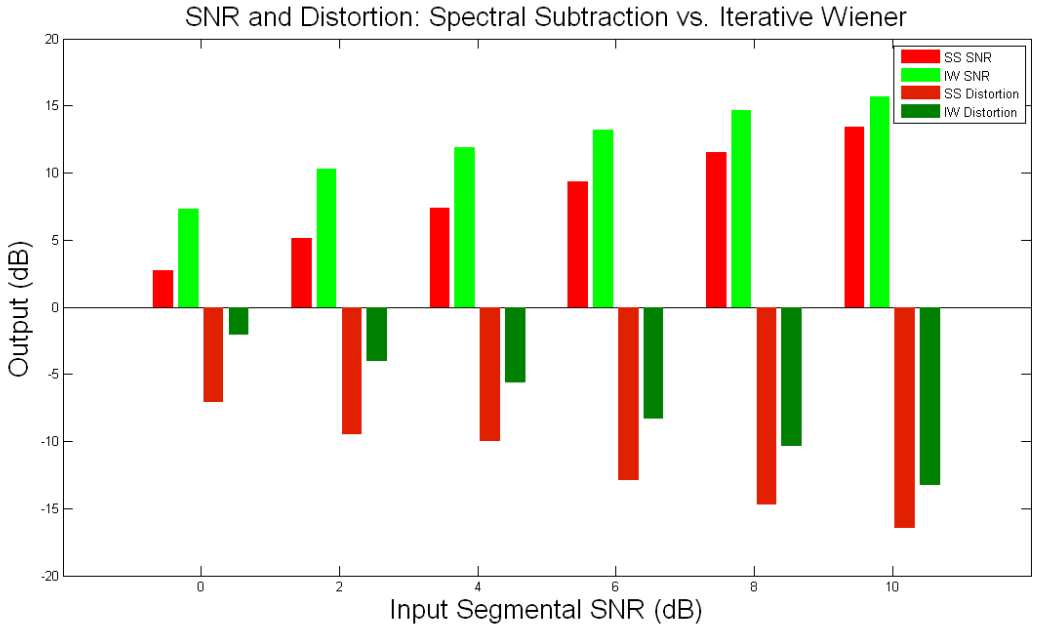
$$Distortion = \frac{10}{M}\sum_{m=0}^{M-1}\log_{10}\frac{\sum_{n=Nm}^{Nm+N-1}(x(n)-\tilde{x}(n))^2}{\sum_{n=Nm}^{Nm+N-1}x^2(n)}$$

Comparison of Results

We now compare the performance of the two algorithms in terms of the measures described above. First we consider Segmental SNR of the output for six different input Segmental SNRs:



The figure above shows that iterative Wiener filtering consistently performs better. When we consider the "filter" SNR below, we notice the same trend. However, the figure also shows that this comes at a price of higher distortion introduced by Wiener filtering.



Conclusions

In this project we implemented, validated and tested two standard algorithms for speech enhancement: spectral subtraction, and iterative Wiener filtering. Our results have shown that the latter performs better in terms of the objective measures described above. However, subjective evaluation (listening) revealed that both algorithms suffer from severe artifacts in the enhanced signals. It may be that more sophisticated methods for noise estimation would improve performance.

Acknowledgement

Thank you to Dr. Radu Balan for his guidance and insights throughout the project.

Bibliography

[1] Deller, J., Hansen, J., and Proakis, J. (2000) *Discrete Time Processing of Speech Signals*, New York, NY: Institute of Electrical and Electronics Engineers

[2] Quatieri, T. (2002) *Discrete Time Speech Signal Processing*, Upper Saddle River, NJ: Prentice Hall

[3] Loizou, P. (2007) *Speech Enhancement: Theory and Practice*, Boca Raton, FL: Taylor & Francis Group

[4] Rabiner, L., Schafer, R. (1978) *Digital Processing of Speech Signals*, Englewood Cliffs, NJ: Prentice Hall

[5] Lim, J. and Oppenheim, A.V. (1978) All-pole modeling of degraded speech, *IEEE Trans. Acoust. Speech Signal Process.,* 26(3), 197-200

[6] Vaseghi, S. (1996) *Advanced Signal Processing and Digital Noise Reduction*, Stuttgart, Germany: B.G. Teubner

[7] Smith, J. (2007) *Spectral Audio Signal Processing*, Stanford, CA.

[8] Leon-Garcia, A. (2008) *Probability, Statistics, and Random Processes for Electrical Engineering*, Upper Saddle River, NJ: Pearson Prentice Hall