Speech Enhancement Preprocessing to J-RASTA-PLP

Michael Shire
EECS 225D
Prof. Morgan and Prof. Gold

1.0 Introduction

The concept of the experiment presented here is to investigate whether a process that can make a noisy observation of speech more intelligible to humans will also make it more intelligible to machine. With that in mind, a speech enhancement algorithm was added as a preprocess to a given robust front end to see if it's performance could be improved upon. J-RASTA-PLP as a front end feature extractor and the Hidden-Markov-Model Toolkit (HTK) as a recognizer serve as the test-bed for this experiment. J-RASTA-PLP attempts to make recognition more robust and invariant to acoustic environment variables. It in some sense performs speech enhancement but for the purposes of feature extraction. The speech enhancement process added was designed to improve intelligibility of noisy speech for human listeners. Addition of this enhancement may improve recognition scores further.

2.0 Speech Enhancement

A plethora of speech enhancement schemes exist. Practically all of them share the common goal of attempting to increase the signal-to-noise ration (SNR). They differ in complexity, ease of implementation, and suitability for real-time processing.

2.1 Selecting an Enhancement Schemes

Of the techniques that exist, those which operated on a single available channel (that is, only the observed signal with noise available) were considered. Some schemes benefit from having a second observation of related noise without the signal. In general, however, this second channel is not typically available.

The methods meeting the noted criteria fall into roughly four categories: Generalized spectral subtraction, speech modelling, fundamental frequency tracking, and other ear inspired methods. With generalized spectral subtraction, SNR is reduced by subtracting an estimated power of the noise from the power of the observed signal. The resulting signal is typically "stretched" in some fashion to reduce the warping done to the speech by the subtraction. Speech modelling methods use an iterative combination of Weiner filtering and linear prediction to obtain an ARMA representation of the speech without the noise. The Fundamental frequency tracking approach is based on the notion that for

voiced segments, the fundamental frequency and it's harmonics are of sole importance and all else is noise. A comb filter can be used to extract the harmonics. Lastly, there are techniques that attempt to use knowledge of the human ear and auditory perception. These procedures use forms of formant peak enhancement.

Of the available schemes, I initiated work on simulating lateral inhibition [5], but finally settled on a method described by Clarkson and Bahgat [1]. This method seemed particularly attractive because it's band processing structure resembles that of the vocoder and especially experiments by R. Drullman [6].

2.2 Envelope Expansion Method

The basic structure of the processing is shown in Figure 1. First band-pass filters separate the speech signal into adjacent bands. The Hilbert envelope of each band is extracted from its excitation. A nonlinear expansion operates on each envelope; a separate expansion is applied for each band. Because the expansion is nonlinear, the envelope passes through a low-pass filter to suppress higher frequency distortion. The expanded and filtered envelope then multiplies back with its corresponding excitation pattern. Finally, the separate bands recombine to form the enhanced speech signal.

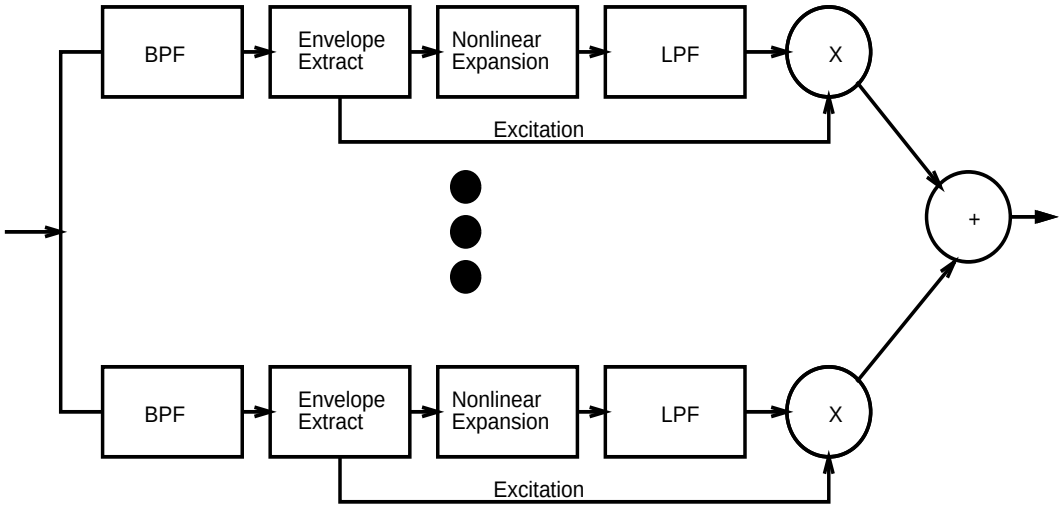

FIGURE 1. Envelope Expansion Method

2.2.1 Envelope Extraction

The envelope and excitation patterns are obtained by using the Hilbert transform and exploiting the fact that each band is approximately narrow band. Given the real signal *x(t)*, the hilbert transform can be computed as $\tilde{x}(t)$. From this we can construct an analytic signal *z(t)*.

$$z(t) = x(t) + j\tilde{\tilde{x}}(t)$$

The analytic signal can also be written in polar form.

$$z(t) = A(t)\, e^{j\phi(t)}$$

A(t) and $\phi(t)$ are both real functions. *A(t)* represents the instantaneous amplitude, or envelope, and can be extracted by:

$$A(t) = mag(z(t)) = \sqrt{x^2(t) + \tilde{x}^2(t)}$$

Similarly $\phi(t)$, the argument of *z(t)*, represents the instantaneous phase. We can recover the original signal by noting Euler's identity:

$$x(t) = Re(z(t)) = A(t)\cos(\phi(t))$$

The signal *A(t)* is used as the envelope to be expanded. The cosine term represents the excitation.

2.2.2 Nonlinear Expansion

The envelopes obtained above are non-linearly expanded by EQ 1, below. $A_k(n)$ represents the envelope of the *k*th band with *n* as the time index. $S_k(n)$ is the new expanded envelope.

$$S_k(n) = \frac{\left[\dfrac{A_k(n)}{\alpha(k,n)}\right]^{\upsilon}}{\left[1 + \left[\dfrac{A_k(n)}{\alpha(k,n)}\right]^{\upsilon}\right]} \times A_k(n) \qquad \text{(EQ 1)}$$

The expansion performs a point-by-point and band-by-band scaling on the original envelope. The ratio in EQ 1, is a function of EQ 2, which in turn is a function of EQ 3,. $\alpha(k,n)$ acts as a type of "importance" function for the envelope and the parameter ν stretches it.

$$\alpha(k,n) = \alpha\left[\frac{\bar{A}_k}{\bar{A}}\right]\left[\frac{C}{\gamma(n)}\right] \qquad \text{(EQ 2)}$$

The scaling parameter $\alpha(k,n)$, instead of being a constant, is made adaptive by being dependant on the mean of the envelope of each band and introducing $\gamma(n)$ below. *C* is a normalizing constant.

$$\gamma^2(n) = \frac{1}{N}\sum_{j=1}^{N} [A_j(n) - \bar{A}(n)]^2 \qquad \text{(EQ 3)}$$

Since the Hilbert envelope of narrow-band signals is identical to the modulus of the short-time Fourier transform (viewed as a function of time), $\gamma(n)$ is in some sense an estimate of the spectral variance. $\gamma(n)$ tends to become large when speech is present.

In EQ 1, EQ 2, and EQ 3, $\bar{A}_k$ is the arithmetic mean of band *k*, $\bar{A}(n)$ is the arithmetic mean across all bands at each point *n*, and $\bar{A}$ is the average across both bands and time.

$$\bar{A}_k = \frac{1}{M}\sum_{n=1}^{M} A_k(n)\ ,\ \bar{A}(n) = \frac{1}{N}\sum_{k=1}^{N} A_k(n)\ ,\ \bar{A} = \frac{1}{MN}\sum_{k=1}^{N}\sum_{n=1}^{M} A_k(n)$$

M is the number of samples of an envelope. N is the total number of bands.

The envelope expansion algorithm does not optimally maximize the SNR. Its intent is to improve intelligibility to listeners as well as reduce listener fatigue caused by noise. It seeks to improve the SNR with minimal warping of the speech component. The enhancing effects and the results on intelligibility are chronicled in [1] and are not re-explored here for the sake of brevity.

3.0 Implementation

The algorithm described in the previous section was implemented as scripts in Matlab. Appendix B contains the principal script listings. The following sections describe the choices of parameters and implementation issues. The scripts were designed to operate on speech sampled at 8 kHz.

3.1 Filterbank

Clarkson and Bahgat used a linearly spaced bank of 20 band-pass filters. I chose to use a logarithmically spaced bank of filters. This is to make the bands correspond closer to the critical bands of the human ear. Experiments by Drullman [2] indicate that manipulation of the band envelopes are ineffective until the band-widths of the filterbanks are at most a quarter-octave in width.

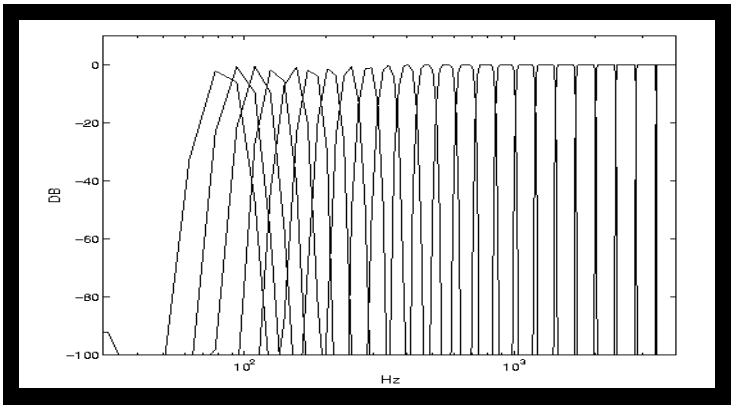

FIGURE 2. Filter bank of quarter-octave filters.

Figure 2 shows the frequency response of the filters. Each filter is a quarter-octave wide starting at 4 kHz and going downward six octaves. Hence there are a total of 24 quarter-octave filterbanks spanning the range of 62.5 Hz to 4000 Hz.

The filters were created using a Hamming window algorithm. The filters are FIR zero-phase having 511 taps.

3.2 Envelope Extraction

The envelope was extracted using the principles described in Section 2.2.1. However, instead of explicitly computing the Hilbert transform to achieve the analytic signal, Matlab's Hilbert() function computed it directly from the incoming signal band. This routine zero-pads the signal so that the length is an appropriate power of 2 and computes its Discrete Fourier Transform using the ubiquitous FFT routine. It then it sets the samples corresponding to the negative frequencies to zero while appropriately scaling the positive and DC frequencies. Lastly, the complex Inverse FFT achieves an approximation of the desired analytic signal. The envelope is then the magnitude of this complex signal and the excitation is the cosine of the argument portion.

3.3 Nonlinear Expansion

There are a few free parameters in EQ 1 and EQ 2. These parameters are α, ν, and C. Clarkson and Bahgat, through empirical testing, determined that $\alpha = 2.5\sigma$, $\nu = 2$, and $C = \sigma/12$ were reasonable parameters with good performance. Here, σ is the standard deviation of the noise. In testing situations where the noise power is not known, it must be estimated from segments of silence. In the experiments performed here, the testing database had known SNR and hence the power of the noise was readily computable using the following:

$$\sigma = \frac{\sigma_{SN}}{\sqrt{\left(1 + 10^{2\frac{SNR}{20}}\right)}}$$

where σ_{SN} is the standard deviation of the incoming signal with noise.

3.4 Low-Pass Filter

The implemented low-pass filter has a cut-off frequency of 25 Hz. This frequency effectively preserves the speech intelligibility information while suppressing the high frequency noise caused by the nonlinear expansion. Experiments have revealed that the important and relevant information with respect to intelligibility in the band envelope is concentrated in frequencies below 16 Hz. [2]

The filter was based on the Hamming-window design method. It is 127-tap FIR with zero-phase response.

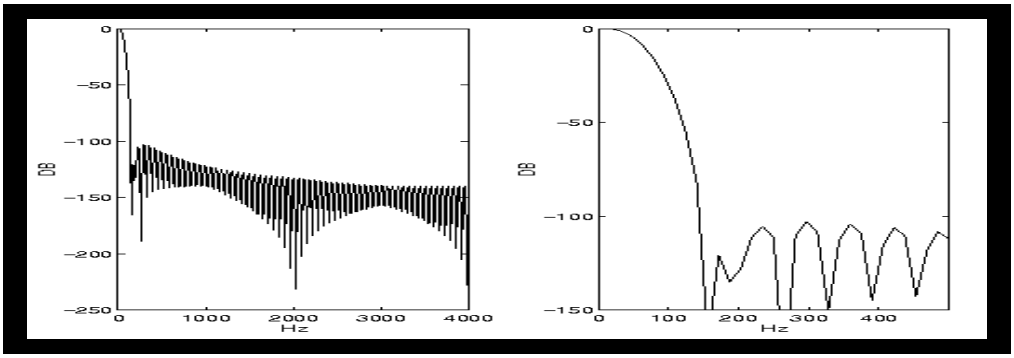

FIGURE 3. Low-pass filter. Cutoff = 25Hz.

4.0 Recognition Tests

4.1 Training and Testing

J-RASTA-PLP served as a front end feature extractor. The analysis window was 25ms with a window step-size of 12.5ms. The order of the PLP model was 9 and the number of output parameters was 10. RASTA was configured to incorporate high-pass filtering and slight spectral subtraction. A constant J of 1e-6 was used for training. Multiple regression J mapping was used during testing.

The Digits database served as the corpus for the training and testing. This database consisted of samples of 200 speakers uttering digits. The digits consisted of 1 through 9 plus ‘oh’, zero, yes and no. For the purposes of expediency in producing preliminary results for this report, HTK trained and tested only on the numbers 1 through 5[1].

An HTK recognizer computed the training and testing recognition results. The recognizer was trained using clean speech and tested using speech with added noise. The SNR levels for the tests were 10dB and 5dB. Samples of “Hynek’s Volvo” served as the noise source. Only additive noise was considered in this experiment.

In summation, the experiment proceeded as follows:

- The recognizer was trained on clean speech.
- The speech with 5dB and 10dB SNR were preprocessed with the speech enhancement algorithm.
- The recognizer tested on processed noise conditions.

1. The speech enhancement preprocessing takes about 2 minutes per utterance using the current Matlab script on a SPARC 20. The Digits database contains over 2000 utterances. Hence, preprocessing the entire database would take on the order of three days per noise condition.

- The recognizer iterated four times using a jackknife procedure; The 200 speakers were separated into four groups of fifty. One set of speakers was used for training and the other three were used for testing. Then the groups were rotated.
- As control and comparison, the recognizer repeated the above steps but without the speech enhancement.

4.2 Results

Table 1 shows the resulting recognition scores in terms of percent correct for each iteration of the jackknife training and testing. The final column is an average over the four iterations. The detailed listings of each result is attached in Appendix A[1].

TABLE 1. Recognition Results

Condition	Iter 1	Iter 2	Iter 3	Iter 4	Avg
10dB SNR **with** speech enhancement	92.00%	94.40%	93.20%	90.40%	92.50%
10dB SNR **no** speech enhancement	91.60%	89.60%	94.80%	89.20%	91.30%
5dB SNR **with** speech enhancement	85.60%	90.80%	88.80%	86.40%	87.90%
5dB SNR **no** speech enhancement	86.00%	85.20%	90.00%	82.40%	85.90%

5.0 Conclusions

The envelope expansion method produces speech that, in the author's opinion, is noticeably cleaner and more intelligible especially in low SNR conditions. Furthermore, performing the expansion on clean speech does not degrade the intelligibility of the speech noticeably. While the process does reduce the noise level, particularly in regions of silence, it does appear to also change the color of the noise. This could have potentially hurt the recognition scores. Fortunately, though, the preliminary recognition scores do not indicate this.

Table 1 shows a very slight improvement in overall performance by up to 2%. This may not be significant since there are some iterations where the addition of the enhancing algorithm fares slightly poorer. But at first glance, it would seem that its addition as a preprocess merits further investigation. There are several variables in the enhancing and also in the front end that if adjusted properly could yield more consistent improvement in recog-

1. The results for the 5 dB SNR with speech enhancement do not match those presented in class due to a subsequent adjustment of a parameter in the enhancing algorithm.

nition scores. The confusion matrices in appendix A show that the enhancing scheme does not generally change the types of errors being made. Hence, the addition of the enhancing scheme does not appear to create any new problems with recognition that were not already apparent in the front end. One can expect that the scores for the other noise conditions including no noise would show similar results.

One of the goals of this experiment was to see if the current performance of J-RASTA-PLP could be improved upon with this additional preprocessing. J-RASTA-PLP with high-pass filtering, slight spectral subtraction, and multiple-regression was used as a test front end because past experiments have shown this to be the configuration yielding the best scores. It may also be instructive to attempt similar test runs on different configurations including log-RASTA and plain PLP to see if performance can be equal to that presented here. Since the envelope expansion algorithm reduces noise using the amplitude contour in the time domain, there is conceivably a duplication of effort with RASTA which filters temporal trajectories of the power spectrum.

The time-constraining nature of this project together with a deadline have limited the range of experiments performed. In addition to those topics of further study mentioned previously, other possibilities abound. For example, employment of a neural network or hybrid based recognizer might produce decidedly different results from those mentioned here. The difference between the linearly spaced filterbank and the logarithmically spaced filterbank used here should be explored. Alternate larger corpus could prove insightful. At the very least, the use of the entire Digits database and not merely the first five numbers would have been desirable for performance measurement. Whether or not intelligibility enhancement geared to human listeners improves machine recognition is not specifically answered in with this project. However, the author's intuition that it is so remains intact with these results.

6.0 References

1. Clarkson, P.M., and Bahgat, S.F. (**1991**). “Envelope Expansion Methods for Speech Enhancement,” J. Acoust. Soc. Am. **89**. 1378-1382.
2. Deller, J.R., Proakis, J.G., and Hansan, J.H.L. (**1993**). *Discrete-Time Processing of Speech Signals*. (Macmillan Publishing Company, New York, NY).
3. Hermansky, H. (**1990**). “Perceptual Linear Predictive (PLP) Analysis for Speech,” J. Acoust. Soc. Am., **87**. 1738-1752.
4. Koehler, J., Morgan, N., Hermansky, H., Hirsch, H.G., and Tong, G. (**1994**). “Integrating RASTA-PLP into Speech Recongition,” ICASSP, Adelaide, Australia.
5. Cheng, Y.M., and O’Shaughnessy, D. (**1991**). “Speech Enhancement Based Conceptually on Auditory Evidence,” IEEE Trans. on Signal Processing, **39**. 1943-1954.
6. Drullman, R., Feston, J.M., and Plomp, R. (**1993**). “Effect of Temporal Envelope Smearing on Speech Reception,” J. Acoust. Soc. Am. **95**. 1053-1064.

Appendix A

Recognition Results

10 DB SNR With Speech Enhancement

```
RASTA parameter: -S 8000 -w 25 -s 12.5 -n 10 -m 9 -J -f /n/icsib76/da/ce/
icsi/mapping/map_weights_8kHz_withHP_10jah_digits.dat -F -M

Wave-files from subdirectory: /n/icsib14/da/shire/ee225d/proj/
dig_enhance/10_noise

Results of iteration 4
**************************************

------------------------ Overall Results -------------------------
PHRASE: %Correct=90.40 [H=226, S=24, N=250]
PHONE:  %Corr=90.40, Acc=90.40 [H=226, D=0, S=24, I=0, N=250]
-----------------------------------------------------------------
Confusion Matrix
        w  w  w  w  w
        _  _  _  _  _
        1  2  3  4  5  Del [ %c / %e ]
  w_1  46  0  0  4  0    0  [92.0/ 1.6]
  w_2   0 49  1  0  0    0  [98.0/ 0.4]
  w_3   0  5 45  0  0    0  [90.0/ 2.0]
  w_4   4  0  2 44  0    0  [88.0/ 2.4]
  w_5   8  0  0  0 42    0  [84.0/ 3.2]
  Ins   0  0  0  0  0

Results of iteration 3
**************************************

------------------------ Overall Results -------------------------
PHRASE: %Correct=93.20 [H=233, S=17, N=250]
PHONE:  %Corr=93.20, Acc=93.20 [H=233, D=0, S=17, I=0, N=250]
-----------------------------------------------------------------
Confusion Matrix
        w  w  w  w  w
        _  _  _  _  _
        1  2  3  4  5  Del [ %c / %e ]
  w_1  48  0  0  1  1    0  [96.0/ 0.8]
  w_2   0 50  0  0  0    0
  w_3   0  1 49  0  0    0  [98.0/ 0.4]
  w_4   8  0  0 42  0    0  [84.0/ 3.2]
  w_5   5  0  1  0 44    0  [88.0/ 2.4]
  Ins   0  0  0  0  0

Results of iteration 2
**************************************

------------------------ Overall Results -------------------------
```

```
PHRASE: %Correct=94.40 [H=236, S=14, N=250]
PHONE:  %Corr=94.40, Acc=94.40 [H=236, D=0, S=14, I=0, N=250]
---------------------------------------------------------------
Confusion Matrix
        w  w  w  w  w
        _  _  _  _  _
        1  2  3  4  5  Del [ %c / %e ]
  w_1  46  0  0  0  4    0  [92.0/ 1.6]
  w_2   0 49  1  0  0    0  [98.0/ 0.4]
  w_3   0  0 50  0  0    0
  w_4   3  1  1 45  0    0  [90.0/ 2.0]
  w_5   4  0  0  0 46    0  [92.0/ 1.6]
  Ins   0  0  0  0  0

Results of iteration 1
**************************************

----------------------- Overall Results ------------------------
PHRASE: %Correct=92.00 [H=230, S=20, N=250]
PHONE:  %Corr=92.00, Acc=92.00 [H=230, D=0, S=20, I=0, N=250]
---------------------------------------------------------------
Confusion Matrix
        w  w  w  w  w
        _  _  _  _  _
        1  2  3  4  5  Del [ %c / %e ]
  w_1  45  0  1  1  3    0  [90.0/ 2.0]
  w_2   0 50  0  0  0    0
  w_3   0  5 45  0  0    0  [90.0/ 2.0]
  w_4   5  0  0 45  0    0  [90.0/ 2.0]
  w_5   3  0  2  0 45    0  [90.0/ 2.0]
  Ins   0  0  0  0  0
```

10 DB SNR Without Speech Enhancement

```
RASTA parameter: -S 8000 -w 25 -s 12.5 -n 10 -m 9 -J -f /n/icsib76/da/ce/
icsi/mapping/map_weights_8kHz_withHP_10jah_digits.dat -F -M
```

Wave-files from subdirectory: /n/icsib76/db/gracet/rmclickData/10_noise

Results of iteration 4
****

---------- Overall Results ----------
PHRASE: %Correct=89.20 [H=223, S=27, N=250]
PHONE: %Corr=89.20, Acc=89.20 [H=223, D=0, S=27, I=0, N=250]

Confusion Matrix
w w w w w
_ _ _ _ _
1 2 3 4 5 Del [%c / %e]
w_1 47 1 0 2 0 0 [94.0/ 1.2]
w_2 0 49 0 1 0 0 [98.0/ 0.4]
w_3 1 6 43 0 0 0 [86.0/ 2.8]
w_4 7 0 0 42 1 0 [84.0/ 3.2]

```
 w_5   7  0  0  1 42    0  [84.0/ 3.2]
 Ins   0  0  0  0  0
```

Results of iteration 3
****

```
---------- Overall Results ----------
PHRASE: %Correct=94.80 [H=237, S=13, N=250]
PHONE:  %Corr=94.80, Acc=94.80 [H=237, D=0, S=13, I=0, N=250]
-------------------------------------------------------------
Confusion Matrix
      w  w  w  w  w
      _  _  _  _  _
      1  2  3  4  5  Del [ %c / %e ]
 w_1  49  0  0  1  0    0  [98.0/ 0.4]
 w_2   0 50  0  0  0    0
 w_3   0  1 49  0  0    0  [98.0/ 0.4]
 w_4   6  0  0 44  0    0  [88.0/ 2.4]
 w_5   5  0  0  0 45    0  [90.0/ 2.0]
 Ins   0  0  0  0  0
```

Results of iteration 2
****

```
---------- Overall Results ----------
PHRASE: %Correct=89.60 [H=224, S=26, N=250]
PHONE:  %Corr=89.60, Acc=89.60 [H=224, D=0, S=26, I=0, N=250]
-------------------------------------------------------------
Confusion Matrix
      w  w  w  w  w
      _  _  _  _  _
      1  2  3  4  5  Del [ %c / %e ]
 w_1  48  1  0  0  1    0  [96.0/ 0.8]
 w_2   0 48  1  0  1    0  [96.0/ 0.8]
 w_3   0  1 49  0  0    0  [98.0/ 0.4]
 w_4  14  0  0 36  0    0  [72.0/ 5.6]
 w_5   7  0  0  0 43    0  [86.0/ 2.8]
 Ins   0  0  0  0  0
```

Results of iteration 1
****

```
---------- Overall Results ----------
PHRASE: %Correct=91.60 [H=229, S=21, N=250]
PHONE:  %Corr=91.60, Acc=91.60 [H=229, D=0, S=21, I=0, N=250]
-------------------------------------------------------------
Confusion Matrix
      w  w  w  w  w
      _  _  _  _  _
      1  2  3  4  5  Del [ %c / %e ]
 w_1  49  0  0  0  1    0  [98.0/ 0.4]
 w_2   0 48  2  0  0    0  [96.0/ 0.8]
 w_3   1  5 44  0  0    0  [88.0/ 2.4]
 w_4   7  0  0 43  0    0  [86.0/ 2.8]
```

```
 w_5  4  1  0  0 45   0  [90.0/ 2.0]
 Ins    0   0   0   0   0
```

5 DB SNR With Speech Enhancement

RASTA parameter: -S 8000 -w 25 -s 12.5 -n 10 -m 9 -J -f /n/icsib76/da/ce/icsi/mapping/map_weights_8kHz_withHP_10jah_digits.dat -F -M

Wave-files from subdirectory: /n/icsib14/da/shire/ee225d/proj/dig_enhance/5_noise

Results of iteration 4
****

---------- Overall Results ----------
PHRASE: %Correct=86.40 [H=216, S=34, N=250]
PHONE: %Corr=86.40, Acc=86.40 [H=216, D=0, S=34, I=0, N=250]

Confusion Matrix

```
      w  w  w  w  w
      _  _  _  _  _
      1  2  3  4  5  Del [ %c / %e ]
 w_1  46  0  0  4  0    0  [92.0/ 1.6]
 w_2   0 48  2  0  0    0  [96.0/ 0.8]
 w_3   1  5 44  0  0    0  [88.0/ 2.4]
 w_4  10  0  0 40  0    0  [80.0/ 4.0]
 w_5  12  0  0  0 38    0  [76.0/ 4.8]
 Ins   0  0  0  0  0
```

Results of iteration 3
****

---------- Overall Results ----------
PHRASE: %Correct=88.80 [H=222, S=28, N=250]
PHONE: %Corr=88.80, Acc=88.80 [H=222, D=0, S=28, I=0, N=250]

Confusion Matrix

```
      w  w  w  w  w
      _  _  _  _  _
      1  2  3  4  5  Del [ %c / %e ]
 w_1  47  1  0  1  1    0  [94.0/ 1.2]
 w_2   0 47  3  0  0    0  [94.0/ 1.2]
 w_3   0  1 49  0  0    0  [98.0/ 0.4]
 w_4  10  0  0 40  0    0  [80.0/ 4.0]
 w_5  11  0  0  0 39    0  [78.0/ 4.4]
 Ins   0  0  0  0  0
```

Results of iteration 2
****

---------- Overall Results ----------
PHRASE: %Correct=90.80 [H=227, S=23, N=250]
PHONE: %Corr=90.80, Acc=90.80 [H=227, D=0, S=23, I=0, N=250]

Confusion Matrix

```
      w  w  w  w  w
      _  _  _  _  _
      1  2  3  4  5  Del [ %c / %e ]
 w_1  45  0  0  1  4    0  [90.0/ 2.0]
 w_2   0 49  1  0  0    0  [98.0/ 0.4]
 w_3   0  0 50  0  0    0
 w_4   5  1  1 42  1    0  [84.0/ 3.2]
 w_5   9  0  0  0 41    0  [82.0/ 3.6]
 Ins   0  0  0  0  0
```

Results of iteration 1
****

---------- Overall Results ----------
PHRASE: %Correct=85.60 [H=214, S=36, N=250]
PHONE: %Corr=85.60, Acc=85.60 [H=214, D=0, S=36, I=0, N=250]

Confusion Matrix

```
      w  w  w  w  w
      _  _  _  _  _
      1  2  3  4  5  Del [ %c / %e ]
 w_1  43  0  1  1  5    0  [86.0/ 2.8]
 w_2   0 48  2  0  0    0  [96.0/ 0.8]
 w_3   0  5 45  0  0    0  [90.0/ 2.0]
 w_4   9  0  0 40  1    0  [80.0/ 4.0]
 w_5  11  0  1  0 38    0  [76.0/ 4.8]
 Ins   0  0  0  0  0
```

5 DB SNR Without Speech Enhancement

RASTA parameter: -S 8000 -w 25 -s 12.5 -n 10 -m 9 -J -f /n/icsib76/da/ce/icsi/mapping/map_weights_8kHz_withHP_10jah_digits.dat -F -M

Wave-files from subdirectory: /n/icsib76/db/gracet/rmclickData/5_noise

Results of iteration 4
****

---------- Overall Results ----------
PHRASE: %Correct=82.40 [H=206, S=44, N=250]
PHONE: %Corr=82.40, Acc=82.40 [H=206, D=0, S=44, I=0, N=250]

Confusion Matrix

```
      w  w  w  w  w
      _  _  _  _  _
      1  2  3  4  5  Del [ %c / %e ]
 w_1  46  1  0  2  1    0  [92.0/ 1.6]
 w_2   1 49  0  0  0    0  [98.0/ 0.4]
 w_3   1 10 39  0  0    0  [78.0/ 4.4]
 w_4  18  0  0 31  1    0  [62.0/ 7.6]
 w_5   8  0  0  1 41    0  [82.0/ 3.6]
 Ins   0  0  0  0  0
```

Results of iteration 3
****

---------- Overall Results ----------
PHRASE: %Correct=90.00 [H=225, S=25, N=250]
PHONE: %Corr=90.00, Acc=90.00 [H=225, D=0, S=25, I=0, N=250]

Confusion Matrix

	w_1	w_2	w_3	w_4	w_5	Del	[%c / %e]
w_1	45	0	0	2	3	0	[90.0/ 2.0]
w_2	0	50	0	0	0	0	
w_3	1	5	44	0	0	0	[88.0/ 2.4]
w_4	10	0	0	40	0	0	[80.0/ 4.0]
w_5	4	0	0	0	46	0	[92.0/ 1.6]
Ins	0	0	0	0	0		

Results of iteration 2
****

---------- Overall Results ----------
PHRASE: %Correct=85.20 [H=213, S=37, N=250]
PHONE: %Corr=85.20, Acc=85.20 [H=213, D=0, S=37, I=0, N=250]

Confusion Matrix

	w_1	w_2	w_3	w_4	w_5	Del	[%c / %e]
w_1	46	0	0	1	3	0	[92.0/ 1.6]
w_2	0	47	2	0	1	0	[94.0/ 1.2]
w_3	0	1	49	0	0	0	[98.0/ 0.4]
w_4	19	0	0	31	0	0	[62.0/ 7.6]
w_5	9	0	0	1	40	0	[80.0/ 4.0]
Ins	0	0	0	0	0		

Results of iteration 1
****

---------- Overall Results ----------
PHRASE: %Correct=86.00 [H=215, S=35, N=250]
PHONE: %Corr=86.00, Acc=86.00 [H=215, D=0, S=35, I=0, N=250]

Confusion Matrix

	w_1	w_2	w_3	w_4	w_5	Del	[%c / %e]
w_1	49	0	0	1	0	0	[98.0/ 0.4]
w_2	0	49	1	0	0	0	[98.0/ 0.4]
w_3	1	10	39	0	0	0	[78.0/ 4.4]
w_4	11	1	0	38	0	0	[76.0/ 4.8]
w_5	9	1	0	0	40	0	[80.0/ 4.0]
Ins	0	0	0	0	0		

Appendix B
Matlab Script Listings

Listing 1: fbankmake.m
Script for making filterbank and low-pass filter

```
% filterbank creation
% quarter octave filters
% spanning 6 octaves = 24 bands

disp 'fbankmake: creating filterbank'
nbands = 24;
fbanklen = 511;

fbank = zeros(fbanklen,nbands);
for i = nbands:-1:2
low = 2^(-i/4);
high = 2^(-(i-1)/4);
f = [low high];
disp([i,f])
fbank(:,(nbands-i+1)) = (fir1(fbanklen-1,f))';
end
fbank(:,24) = (fir1(fbanklen-1,high,'high'))';

lpflen = 127;
lpf = fir1(lpflen-1,(25/4000))';

clear m i low high f

save filterbank fbank nbands fbanklen lpf lpflen

% how to plot frequency response
% mm = fft(fbank,512);
% plot(20*log(abs(mm(1:256,:))))
% set(gca, 'xscale','log')
```

Listing 2: envexpand.m
Script that implements the speech enhancement process

```
% envelope expansion routine
% - use filterbank to split signal
% - use hilbert to extract envelope and excitation
% - expand envelope
% - recombine envelope with excitation
% - sum and scale

%signame = 'blah/blah'
%sigsource =
%sigdest =
%sigoutmat =
%sigoutsd =

if ~exist('snr')
```

```
snr = 5
end

% make sure the filterbank is loaded
if ~exist('fbanklen')
disp 'loading filters'
load filterbank.mat
end

% convert the file to a matlab file
cmd = ['!fea2mat -f samples ',sigsource,' ',sigdest];
disp(cmd)
eval(cmd)

% load the file
cmd = ['load ',sigdest];
disp(cmd)
eval(cmd)

% compute alpha
if exist('doalpha')
pwr = std(samples);
alpha = pwr/((10^(2*snr/20)+1)^0.5)
end

% filterbank the samples
disp 'separating out channels'
channels = zeros(length(samples)+fbanklen-1,nbands);
for i = 1:nbands
disp(['channel ',int2str(i)])
channels(:,i) = conv(samples,fbank(:,i));
end



% compute envelopes and excitations
disp 'computing envelopes and excitations'
chanhilb = hilbert(channels);
A = abs(chanhilb);
excit = cos(angle(chanhilb));
clear chanhilb

% expand the envelopes
doexpand

% low pass filter the expanded envelopes
disp 'low pass filtering the expanded envelopes'
B = zeros(length(A)+lpflen-1,nbands);
for i = 1:nbands
disp(['envelope ',int2str(i)])
B(:,i) = conv(A(:,i),lpf);
end
cut = (lpflen-1)/2;
[chm,chn] = size(B);
```

```
A = B((cut+1):(chm-cut),:);
clear chm chn B i


% resynthesize
disp ‘resynthesizing signal’
channels = A.*excit;
clear A excit
resynth = (sum(channels’))’;

% adjust for delay and added samples due to convolution filter
disp ‘adjusting for delay and resizing’
cut = (fbanklen-1)/2;
[chm,chn] = size(resynth);
samples = resynth((cut+1):(chm-cut));

% save to binary file
fid = fopen(sigoutmat,’wb’);
fwrite(fid,samples,’short’);
fclose(fid);

% convert to esps file
cmd = [‘!btosps -t short -f 8000 -n 1 -c “matlab” ‘,sigoutmat,’
‘,sigoutsd];
disp(cmd);
eval(cmd);

disp ([sigoutsd, ‘ written’])


% save to matlab file
%cmd = [‘save ‘,sigoutmat, ‘ samples’];
%disp (cmd)
%eval(cmd)
```

Listing 3: doexpand.m
Script subroutine that perform envelope expansion

```
% doexpand
% script to do the envelope expansion
% assume the envelopes are columns of A

nu = 2;
%alpha =


disp ‘computing expansion parameters’

AN = (sum(A’))’ ./nbands;
AK = (sum(A))’ ./length(A);
AA = sum(AK) ./nbands;

gamma = zeros(size(AN));
for i = 1:nbands
```

```
gamma = gamma + ((A(:,i)-AN).^2) ./nbands;
end
gamma = gamma .^ (0.5);

Akn = ((1 ./ gamma)* AK') * alpha * nbands / AA .* alpha ./ (2.5*12);

Skn1 = (Akn ./ A) .^ nu;
A = A ./ ( Skn1 + 1 );

clear AN AK AA gamma i Akn Skn1
```