Speech Volume Monitor for Hearing Impaired

R.DEEPA *(Mphil Research scholar)*
PSGR Krishnnaml college for women.
GRG School of Applied Technology
Coimbatore,India

R.AMSAVENI *(Lecturer)*
PSGR Krishnnaml college for women
GRG School of Applied Technology
Coimbatore,India

***Abstract*— Hearing impaired can be classified into people who were affected by birth and those who developed the problem at a later stage. The second category knows how to speak but cannot hear them. They encounter embarrassment by not adjusting their speech levels to their surroundings. In this paper the development of a system helps them to adjust their speech volume to the surrounding taking into consideration the ambient noise. It is achieved by using VAD (Voice Activity Detector) and spectral estimation for the speech spectrum. The system proposed will generate pulses which tell the user to raise or lower the volume.**

Keywords*— **Signal Processing, Speech Processing, Speech Enhancement, Hearing aids, Auditory System***

I. INTRODUCTION

The hearing impaired (considering only people who lost their hearing ability later) are not able to determine the volume at which they should be speaking. They get the feedback from listeners mainly by lip synching and facial expressions which hardly tell them at what volume they should be speaking. At times they are not even heard when the environment is too noisy.

The noise in the environment is measured and then add it to a level at which the speech of the person is clearly audible. This can be then listed in a table to adjust the speech level. Yet another method is to compare the voice of the user with other speakers in the environment and then give a feedback to the person to speak in a range of 10 dB in relation to the measured value. In this paper two methods are used to measure the noise. Using VAD (voice activity detector), and using spectral estimation for the speech spectrum. It is considered that the areas of spectrum having the minimum spectral characteristics are background noise. Then generate and send the vibrations to user using a vibrating embedded hardware. When the system sends very frequent pulses, it indicates that the user will have to raise the speech volume to be audible. [1]

II. SYSTEM DESIGN & ALGORITHM OVERFLOW

For both online and offline system, female voiced audio file is sampled on 22050 Hz. The 2.866 sec file says “don’t ask me to carry an oily rag like that”.

Methods to test robustness:

1) Hit: indicates how much percentage the algorithm succeeds in correctly identifying the presence of speech in the sentence given.

2) *Error:* indicates how much percentage, the algorithm fails to correctly identify the presence of speech in the sentence given. It is just the opposite of hit in the context.

Offline approach

There will be four parts in the offline implementation:

- Power spectrum calculation.
- Implementing the algorithm.
- Speech pause detection.
- Output sent to user.

1) An efficient algorithm to trace the noise:

To calculate the signal to noise ratio and noise power in a noisy environment, the algorithm was used which was initially developed by Rainer Martin [7]. The algorithm is based on the assumption that noise has minimum spectral values, so calculate the minimum values of the smoothed power estimate.

The spectral envelope contains peaks and valleys that correspond to the speech and noise. For better performance, smoothed power envelope is taken. The algorithm will be tracking the valleys in the smoothed power spectrum. It then creates a noise power estimate from it in a finite range. A matlab simulation can be performed.

The noisy signal is the sum of the speech signal s and noise signal n.

The noisy composed signal $x(i) = s(i) + n$

The representation is in the time domain. Also s and n are statistically independent. Hence

$E\{x2(i)\} = E\{s2(i)\} + E\{n2($.

By taking the minimum of a smoothed short time power estimate $P\tilde{x}$ within a window of samples, compute the noise power estimate Pn

This algorithm can be divided into

1. Calculating the smoothed power spectrum estimate over a short window of time.
2. Finding out the minimum in this power estimate and determining the noise power estimate $Pn(i)$

The power spectrum can be obtained by the FFT algorithm or by using an exponential sliding window of length N.

Let $P\tilde{x}(i)$ be the smoothed power estimate. The smoothing takes place in short time slot at a time index i and is performed by means of a recursive system. The smoothing factor α in the recursive system is chosen between 0.95 and 0.98.The recursion starts at $i \geq N$

$$Px(i) = Px(i-1) + |x(i)|^2 - [x(i-N) * x(i-N)]$$
$$P\tilde{x}(i) = \alpha * P\tilde{x}(i) + (1-\alpha) * Px(i))$$

N= 128 or 256, we assume that N=156.

Noise Power estimation

Assuming that a window of L samples and, it is needed to have the noise power estimate, track the minimum from the signal power within this window. Then decompose the window with L samples to smaller windows with size M. Now $M * W = L$ .The sampling frequency $fs = 8KHz$. Typical values suggested by Rainer Martin [6] are: M=1250, W=4 and L=5000. These parameters correspond to a time slot of 0.625 sec. For every M samples, a sample by sample comparison between the actual minimum $P_{Mmin}(i)$ and the smoothed signal power $P\tilde{x}(i)$ is conducted. The M samples have been read, i.e. $i = r * M$, and the minimum power P_{Mmin} of the last M samples will be reassigned to its maximum value. [4]

i.e $P_{Mmin}(i = r * M) = Pmax$

There are two kinds of noise power. Slow varying noise power and monotonically increasing noise power.

Slow varying noise power doesn't increase constantly (non monotonic) in this case the noise power is set to minimum of L samples,

i.e. $P_n(i) = P_{Lmin}(i)$ which is obtained by taking the minimum for last W minimum power estimates:

$$P_{Lmin}(i) = min(P_{Mmin}(i = r * M), P_{Mmin}(i = (r-1) * M), P_{Mmin}(i = (r-2) * M), \ldots, P_{Mmin}(i = (r-W+1) * M))$$

In monotonically increasing noise power, minimum power of the last W windows is always increasing. And hence the noise power is equal to the minimum of the last M samples

$P_n(i) = P_{Mmin}(i = r * M)$.

If the estimated noise power is bigger than smoothed power then $P_n(i)$ is updated to the minimum of the two,

i.e. $P_n(i) = min(P_x(i), P_n(i))$.

2) Noise estimation by speech pause detection

This algorithm is based on Marzinzik and Kollmeier [1]. The algorithm tracks the envelope dynamics and finds if speech is present or pause is present in the data. Whenever there is a pause at this instant, the envelope spectral characteristics represent the noise characteristics. To calculate the signal temporal power envelopes apply DFT transform to the input signal and then summation of the squared frequency components over the whole band is done.

$$E(p) = \sum_k |X(p, wk)|^2$$

At time$p$, $X(p, wk)$ is a spectral component of the signal. Then divide the whole spectrum into high pass and low pass characteristics and process it.

$$E_{LP}(p) = \sum_l |X(p, wl)|^2$$
$$E_{HP}(p) = \sum_m |X(p, wm)|^2$$

l is from zero to cut-off frequency, and m is in the remaining spectrum (high-pass).Smoothing to the spectral components is applied by averaging. Low pass recursive filter is used for low pass averaging over short time intervals with release time τE. If the signal is increasing, stop the smoothing so that no smearing happens over the onsets.

1. Initializing the process.

At first leave 200 ms for initial phase of noise. Then assign the maximum and minimum values as follows:

$$E_{min}(p) = E(p) \quad E_{max}(p) = E(p)$$
$$E_{LP,min}(p) = E_{LP}(p) \quad E_{LP,max}(p) = E_{LP}(p)$$
$$E_{HP,max}(p) = E_{HP}(p) \quad E_{HP,max}(p) = E_{HP}(p)$$

matches the minimum values of $E_{LP}(p)$, $E_{HP}(p)$ to the noise energy at the beginning.

2. Updating values of the envelops

a) The new maximum is set to the current value, if the current value of envelope exceeds the maximum. if it doesn't exceed, the maximum value is decreased with a time constant(τ_{decay}) where the input to the recursive first order low pass filter are current envelope values.

b) If current value of the spectral power is below the minimum assumed, then a new minimum is set to the current value. Else the envelope is slowly raised by recursive filtering with a time attack (τ_{raise}) where the input is the current value of the envelope.

3. Calculating differences between maximum and minimum values.

$$\Delta(p) = E_{max}(p) \quad E_{min}(p)$$
$$\Delta_{LP}(p) = E_{LP,max}(p) \quad E_{LP,min}(p)$$
$$\Delta_{HP}(p) = E_{HP,max}(p) \quad E_{HP,min}(p)$$

4. Deciding about pause or speech present

a) If both values of envelope are below certain threshold η then, no speech is present. Only noise is found in this frame

b) Pause is decided based on information from low pass band.

c) Decision made upon information from high pass band information.

i. $\Delta_{LP} < \eta$ and $\Delta_{HP} < \eta$

This case is represented of low range dynamics. Only noise is assumed to be present in this frame.

ii. If $\Delta_{LP}(p) < \eta$ and previous condition fails, then there are very small dynamic changes and no LP pause is found in

this frame. If $\Delta_{LP}(p) > \eta$, and difference between $E_{LP}(p)$ and $E_{LP,min}(p)$ is smaller than ρc of the Δ_{LP}, then the envelope values are close to its minimum. Finally concentrate on high pass-band characteristics to make the right decision. [8]

• Case (i): The difference Δ_{HP} is smaller than η.In this case, speech pause is detected because of the low dynamics in high pass band range. Else it is not determined if speech pause is detected.

• Case(ii): The difference in the high pass band $\Delta_{HP}(p)$ is bigger than twice .In this case, examine the difference between the current envelope $E_{HP}(p)$ and $E_{HP,min}(p)$ .If it is less than twice the fraction ρc of Δ_{HP}, then assure that with the small envelope in low pass band, it is a speech pause. Else, speech might be found in this frame.

• Case (iii): The difference is smaller than 2η but bigger than η

This is an ambiguous case. It is required that $E_{HP}(p)$ lies in its lower half of its dynamic range. In such case, it is sure that a speech pause is present, else speech might be present.

5. The b) section assumes that the mitigated noise is of high pass nature. The decision of presence of speech pause is made based on information from the low-pass band. In the case of disturbing noise of high pass nature, apply the same study with same conditions examined. Difference is to exchange every LP with HP and vice versa.

To test robustness, the error-rate was introduced. To have an optimal error rate, adapt the threshold η and it fragment ρC. For a low error rate, it will reduce the speech distortion in the subsequent noise reduction process. Generate different kinds of noise, and use different levels of noise in dB's and different SNR's to test the performance.

The best sampling frequency for the DFT of the signal is 22050 Hz. This is then passed by partitions of 8 ms windows and padded by zeros to avoid delays. The cut-off frequency that separates low-pass and high passes frequency is chosen between 1.9 KHz and 2 KHz so that it does not affect the intelligibility of the speech. The time constant for envelope smoothing was set to 32 ms, τ_{raise} and τ_{decay} were set to 3s. In real life, under normal condition, these values simulate the actual signal. Threshold η is 5 dB and fragment of it is ρc. Estimate the speech power along with noise power for a proper feedback signal to the user.[12]

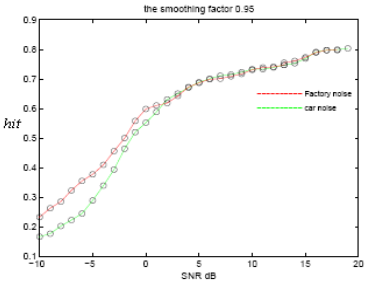
IV.ALGORITHM IMPLEMENTATION

Offline Approach

For implementing algorithm, the values are used which is suggested by Marzinzik and Kollmeier [1]. The parameters which were found more suitable are also introduced. When the noise rises to higher levels, it masks the voice of speakers in the environment.

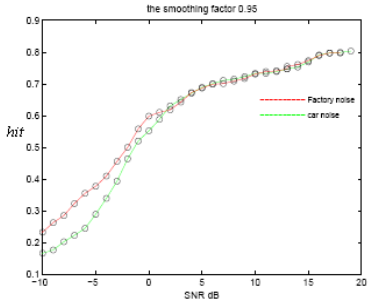
To test the effect the smoothing factor on the hit and the error of the speech pause detection algorithm, standard audio files are used and add a fixed amount of noise and then change the amount of the noise till 30 results ranging from -10 dB to +20 dB. The results are compared after running the modified algorithm with three different values of α (smoothing factor) 0.95, 0.90 and 0.85.

(A)Smoothing factor 0.95



(a) *hit* to SNR

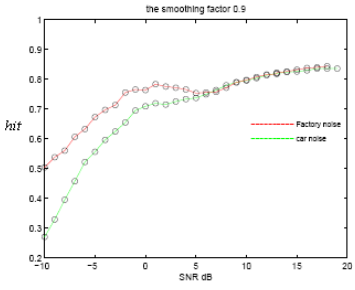
Fig.1a:Hit decreases near the value 0 dB till it reaches almost 18% of its value at -10 dB.



(a) *hit* to SNR

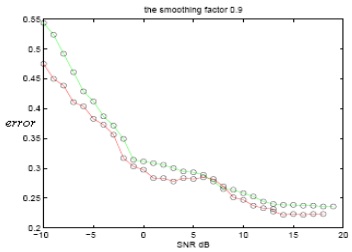
Fig.1b:Error rises to 60% for -10 dB. The algorithm is not able to identify the speech in the audio file at this point due to the high level of noise. On a high dB's (over 7 dB), error drops to almost 20%.

Smoothing factor 0.90



(a) *hit* to SNR

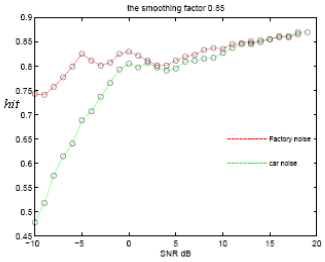
Fig.2a: Improved Hit rate to 11.5% for car noise and 16.13% for factory noise. Reduces the efficiency of the algorithm. The SNR is around -5 dB.



(b) *error* to SNR

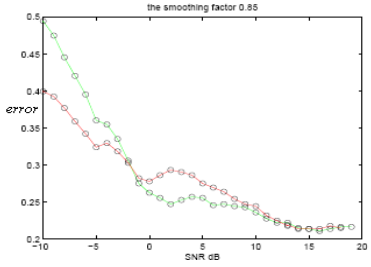
Fig.2b:Improved the error rate to 5.69% for car noise and 8.14% for factory noise.

(A) Smoothing factor 0.85



(a) *hit* to SNR

Fig 3a:Obvious improvement in hit. 7.83% improvement with smoothing of 0.9 and 0.23% with smoothing of 0.95 for car noise, and 7.38%, 23.51% respectively for factory noise.



(b) *error* to SNR

Fig 3b: Improvement for error by reducing the smoothing factor. Reduced by 3.48% and 9.17% for smoothing factor of 0.9 and 0.95 respectively for car noise. Reduced by 3.08% and 11.3% for smoothing factor of 0.9 and 0.95 respectively for factory noise.

The power spectrum of the noise will have sharp edges and rapid fluctuations because of the lowering of smoothing factor. Hence the algorithm will make wrong decisions taking noise for a speech. The feed-back signal is not affected if some parts of the noise are considered as speech because there will be enough pause intervals to update noise power. High smoothing factor (0.95) is used in environments where speech is not presented and also in situations where noise level is high that it masks the speech. It is preferable to use low smoothing factor (0.85) when conversation occurs to determine speech and pause parts accurately, and construct feed-back signal accordingly. *By* using a filter it removes all peaks that were diagnosed wrongfully, which are shorter than 15 samples (60 ms). Hit and error of algorithm's output will be studied for all smoothing factors to see the improvement[13]

Smoothing factor 0.95 : There was a degradation of 0.66% and 1.18% on the hit for car noise and factory noise respectively after introducing the filter, and no change in error.

*Smoothing factor 0.9:*There was degradation in hit up to 4% and 5.8% for car noise and factory noise respectively. The error remains the same.

*Smoothing factor 0.85:*There was higher degradation in performance up to 7% in Hit for both car noise and factory noise and a 1% improvement in error.

Combine the usage of the filter with low smoothing factor in only noise environment to have better results.

To calculate the noise power, the system will save the power of no speech segment sample and check the one after and save it if it's noise also. If the system finds a speech sample, then it calculates the average of previous noise power samples. In a situation, the long periods of just noise in the frame without speech, the system will calculate the power each 1.5 sec, and will adapt to the changes of noise level in the ambient environment.

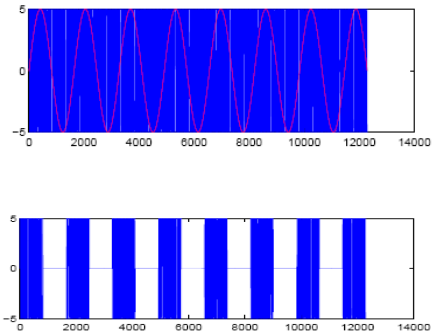
When speech is present in the mixed signal, it is also averaged in the same way and saved to be used in the output part where feedback signal is shaped. After finishing power calculation for noise and speech, the values of averaged powers are received along with the audio sample, where the power of speech and noise are known.

User Output

The user output signal is created from the power calculated in previous stages and scaled. It then forms a vibration which changes in frequency or intensity depending on output from the algorithm. The vibration output signal is shaped in following two ways.

Output depending on noise

This method depends on changing the frequency or the amplitude of a sinusoidal. Two overlapped sinusoidal waves are used to create one wave. The first sinusoid has a very high frequency and the overlapping sinusoid frequency depends on the power of noise. The higher the power, the higher is the frequency. The two waves are overlapped and the output is created only if, the second signal is over zero. This is shown in figure 4.



Output depending on noise and speech power

The second method relies on both noise and speech powers to give better output. Assuming that the speech power got from the algorithm is the power of speech of the user himself and also that other speakers will be speaking in a reasonable right level, and user will speak in higher or lower than this level. The power of speech of user is used to send him the information to raise or lower the voice.

. The output signal created from the noise power is multiplied depending on its frequency. If speech level is higher than the optimum, the Linespace will change from 1 to 0. This tells the speaker to lower his voice as shown in the figure 5

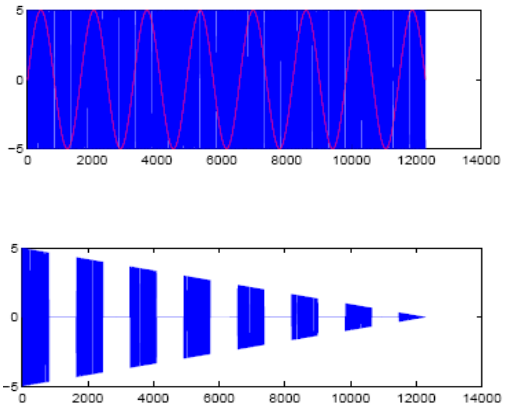


Fig.5

If the speech power is lower than the optimum it will be multiplied by Linespace going from 0 to 1 along the 1.5 sec output. The user will know that he is advised to raise his voice.

Once the user comes close to the optimum area, Linespace will not change further. The component left in the output is the frequency of the noise.[14]

Online Approach

To test online algorithm the Matlab platform is used to provide a multi microphones live feeding to a PC. After processing, the output is fed back directly from the device. The online method gives us extremely fast updating for output when the input changes. This approach is based on the same principles of offline approach.

To calculate hit, and error of output of the algorithm for different noise levels starts from -9 dB and up to 15 dB with 3 dB increase using the same audio sample used in offline approach. In this paper factory noise is used as the noise. Four different levels of noise are taken to find the SNR. The microphone is kept 20 cm away from a speaker, and audio sample is played. The average power of it is calculated. Another speaker 20 cm away from the microphone is used to find the average power of noise. The SNR is calculated using the equation given below.

$$SNR = 10*log10(P_{speech}/P_{noise})$$

The procedure is repeated for calculating all SNR's with varying speech and noise power and composite signals. Then the system is tested.

For testing the smoothing factors 0.9 and 0.95are used. The testing method is same as offline approach. Finally take 9 trials are taken, for SNR from 15 dB to -9 dB.With low noise environment; the algorithm gave us up to 81.46% of hit and only 10% error. Increasing the noise up to 9 dB and 6 dB gives a decrease in the hit to 76.97% and 76.97% respectively and an increase in the error upto 14.4% and 15.6% respectively. By increasing noise level up to 3 dB then it will reach 68.54% Hit and error up to 18%. The algorithm output compared to real pauses in the speech from audio sample is shown in figure. When the noise is increased to reach levels of -6 dB and -9 dB, the performance of our speech pause detection algorithm is degraded severely. At -6 dB the hit fall down to 52.25% and at -9 dB it reached 42.13%, on the error side, it was up to 35.2% and 38.4% for -6 dB and -9 dB respectively.

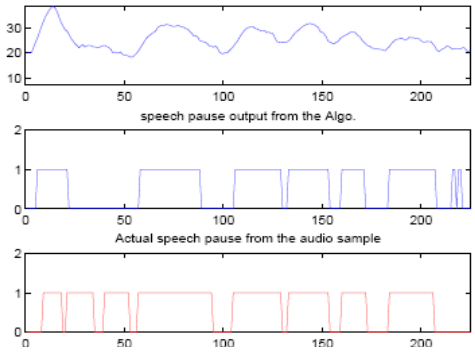


Fig.6

Figure 6 shows the output from the algorithm at SNR equal to -9 dB compared to real pause in speech from our audio sample.

The same test is done to find differences of using different smoothing factor. The smoothing factor is changed from 0.90 to 0.95 and results are compared. In low level noise environment (6dB and higher), raising the smoothing factor to 0.95 didn't change significantly. The degradation on hit is from 1% and up to max 4%. On error the degradation was only up to 4%. When the noise is raised to higher levels (-3 dB and lower), the effect of lowering the smoothing factor is obvious. The hit was degraded from 52.25% to 19.1% at -6 dB and error is higher by 15% for same noise level.

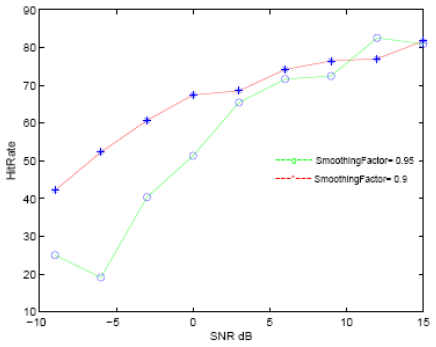


Fig.7

The results for hit for both smoothing factors (0.9 and 0.95), and for noise levels ranging from 15 dB to -9 dB is shown in the figure7. Lower smoothing factor gave us a lower error and more robust results. The main draw back of using a lower smoothing factor is the degradation of error for noisy environment when no speech is present. For the input of pure factory noise without speech, if smoothing factor of 0.95 or 0.9 is compared, the error is improved from 8% to 31,87% when 0.9 is used. In pure noise environments, the error occurs when the algorithm takes certain parts of the noise and considers it as a speech. This gives a wrong estimation about the speech power. If the speech power is not used in creating the output signal sent to the user, this will not have a significant impact on the robustness of algorithm.

Based on the output from our speech pause detection algorithm the noise and the speech power are calculated. This calculation is done sample by sample to maintain online concept. The noise power and the speech power can be used for creating the output for the user.

Output to user

To create the output signal for user, only noise power is used in online approach. Power of speech is not used because no post processing could be done.

For creating output for user, the first method used is based on the online approach. A fixed DC value is added for each single noise power coming from the algorithm, and decision is made by observing to what range of noise levels it belongs to. If there are speech samples in the overall signal, the latest value of noise power is assigned to the speech power. Till the next noise the same output will be given. Depending on the range of values the power of the noise plus the DC value belongs to, a scaled value will be given, and that value will be representing the power in creating the output. If the noise power is higher, the scaled value also is higher.

The scaled values are following the increasing of noise. The drawback is the fluctuation that happens when the power changes its value by small amount around the threshold value, which decides to which level of noise it belongs. So the solution is given to this problem by making an update for the noise for each few samples (5 to 15 samples). This corresponds to updating the output each 100 to 150 ms. Depending on the average of those samples power, the scale is updated. The system is a combination of offline and online approach for the adequate solution.[15]

III. CONCLUSIONS

In our system, VAD gives an inspiring hit and tell the user, at what point of time the volume of speech should be increased or decreased based on the noise level of environment. It is achieved by checking out the spectrum & power density of signal. This algorithm can be implemented in a small hardware device, which won't be so embarrassing to the user, which can be incorporated in hearing aid device. These hearing aid device it self have a noise estimator built-in already, which means only signal based output is required, for our VAD.

More work can be done to the system by introduction of some type of self analyzing for pervious hit & error made by system and can be saved for future analyzing. This would improve the device reliability. Most common usage is Artificial Intelligent system, which will make the reliability of device much higher than the present and proposed device. This methodology will look for patterns that are registered as noise and voice. This information will help the device to find the right type of information & learn from the mistakes (error), to intimate the user for right occasion. This type of system will study all the way, day by day improving the efficiency of the system dramatically.

The interface to user can be made using a vibrator like instrument that could be placed at abdomen or wrist, with band type forming device. These devices will vibrate according to the output made by the system after calculating amount of change that should be made by the user (either rise or lower the volume). The mentioned device will not be incorporated with any other device because it would be clearer for the user not to be confused with other output signals.

REFERENCES

[1] M. Marzinzik and B. Kollmeier. Speech pause detection for noise spectrum estimation by tracking power envelope

dynamics. Speech and Audio Processing, IEEE Transactions on, 10(2):109–118, Feb 2002.

[2] C. Breithaupt, T. Gerkmann, and R. Martin. A novel a priori snr estimation approach based on selective cepstro-temporal smoothing. 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, pages 4897–4900, 2008.

[3] L. Buera, J. Droppo, and A. Acero. Speech enhancement using a pitch predictive model. 2008 IEEE International Conference on Acoustics, Speech and Signal Processing , pages 4885–4888, 2008.

[4] J.S. Erkelens and R. Heusdens. Fast noise tracking based on recursive smoothing of mmse noise power estimates. 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, pages 4873–4876, 2008.

[5] Fischer and V.Stahl. On improvement measures for spectral sub-traction applied to robust automatic speech recognition in car environments. Workshop Robust Methods Speech Recognition Adverse, 1999.

[6] H. Knutsson, M. Andersson, and J. Wiklund. Advanced filter design. In Proceedings of the 11th Scandinavian Conference on Image Analysis, pages 185–193, Greenland, June 1999. SCIA. Also as report LiTH-ISY-R-2142.

[7] Rainer Martin. An efficient algorithm to estimate instantanous snr of speech signals.Eurospeech, 1999.

[8] Yao Ren and M.T. Johnson. An improved snr estimator for speech enhancement. 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, pages 4901–4904, 2008.

[9] D. Rudoy, P. Basu, T.F. Quatieri, B. Dunn, and P.J. Wolfe. Adaptive short-time analysis-synthesis for speech enhancement. 2008 IEEE International Conference on Acoustics, Speech and Signal Processing, pages 4905–4908, 2008.

[10] Daniel Gureasko Bobrow. Artificial intelligence in perspective.

[11] Minker, Jack (Ed.). Logic-Based Artificial Intelligence.

[12] Dov M Gabbay, Christopher John Hogger. Handbook of Logic in Artificial Intelligence.

[13] Svetlana N Yanushkevich. Artificial Intelligence in Logic Design.

[14] Toshinori Munakata. Fundamentals of the New Artificial.

[15] Alex Waibel. Prosody and Speech Recognition.